

КОНСПЕКТ ЛЕКЦІЙ

ЗА КУРСОМ «ІНФОРМАТИКА та КОМП'ЮТЕРНА ТЕХНІКА »

ЗМІСТ

1. АПАРАТНІ ЗАСОБИ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ	4
1.1 Інформатика та інформаційні технології	4
1.2 Персональні комп'ютери. Особливості їх складу, конструкції і конфігурації.	7
1.3 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: системний блок, блок живлення, мікропроцесор, материнська плата, оперативна пам'ять ...	11
1.4 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: накопичувачі, адаптери	14
1.5 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: звукові карти, монітор, клавіатура	17
1.6 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: принтер, модеми, сканери, джерела безперебійного живлення.....	19
2. Операційна система windows xp. основні поняття на принципи роботи у системі.	25
2.1 Історія Windows	25
2.2 Файлова система та структура.	27
2.3 Основні принципи роботи з системою	29
2.4 Робота з файлами та папками.	35
3. ВВЕДЕНИЕ В MICROSOFT OFFICE.	41
3.1 MICROSOFT WORD.	41
3.1.1 ЗАПУСК І ОЗНАЙОМЛЕННЯ З ПАКЕТОМ.....	41
3.1.2. НАЛАШТУВАННЯ ПАРАМЕТРІВ СТОРІНКИ	44
3.2 Текстовий редактор Open Office org Writer	65
3.2.1 Робота з документами	65
3.2.2 Форматування тексту	68
3.2.3 Створення таблиць та діаграм	71
3.2.4 Вставка об'єктів у документ	74
3.2.5 Призначення стилів і їх використання в документах	75
4. ЕЛЕКТРОННІ ТАБЛИЦІ (MS Excel та OO Calc).....	82
4.1. Загальні положення	82
4.2. Типи даних, введення та форматування даних у ЕТ	84
4.3. Редагування даних в ЕТ	89
4.4. Функції та обчислення у ЕТ	91
4.5. Організація графічної інформації	98
4.6. Робота з базами даних (списками)	101
4.7. Аналіз даних.....	108
5. ЗАГАЛЬНІ ПОНЯТТЯ БАЗ ДАНИХ.	116
5.1.3 Мережева модель даних	119
5.1.4 Реляційна модель даних	120

5.1.5 Рівні моделі даних.	120
5.2. Інфологічеська модель даних "суть-зв'язок"	121
5.2.1 Поняття, використовувані в інфологічному моделюванні.	121
5.2.2 Основні класи суті.	122
5.2.3 ER- діаграми і мова інфологічеського моделювання (ЯІМ).....	123
5.2.4 Види зв'язків.....	124
5.3. Реляційна база даних.....	125
5.3.1 Основні поняття, використовувані в реляційних базах даних	125
5.3.2 Правила побудови реляційних баз даних	127
5.3.3 Поняття універсального відношення	128
5.3.4 Нормалізація.....	131
5.3.5 Процедура проектування	131
5.4. Microsoft Access. Создание таблиц	132
5.4.1 Запуск программы Microsoft Access	132
5.4.2 Створення таблиць.....	134
5.4.3. Види представлення таблиці.....	140
5.4.4. Редагування проекту таблиці.....	141
5.5. Визначення зв'язків між таблицями.....	143
5.6. Створення запитів.....	145
5.6.1 Створення запитів в режимі конструктора.....	146
5.6.2 Приклади запитів	147
5.6.3 Створення запитів в режимі SQL	149
5.6.4 Інструкції SQL.....	149
5.7. ACCESS. СТВОРЕННЯ ФОРМ	152
5.7.1 Майстер форм.....	153
5.7.2 Режим Конструктора	155
5.7.3 Заповнення Форми в режимі конструктора.....	156
5.7.4 Виклик однієї форми з іншої	159
5.7.5 Додавання в Примітку форми поточної дати і час	160
5.8. ACCESS. СТВОРЕННЯ ЗВІТІВ	162
5.8.1 Кроки створення звіту в режимі Майстра звітів.....	163
5.8.2 Створення звітів в режимі Конструктора	163
6. Комп'ютерні мережі.....	165
6.1. Введене	165
6.2. Класифікації мереж	166
6.2.1. Класифікація за розміром охопленої території :	166
6.2.2. Класифікація за типом функціональної взаємодії	168
6.2.3. Класифікація за типом мережевої топології.....	172
6.3. Огляд мережевого устаткування.....	175
6.3.1 Крайове устаткування лінії зв'язку.	175
6.3.2 Проміжне мережеве устаткування.....	177
6.3.3 Комутатор	180
6.3.4 Маршрутизатор	180

6.3.5 Шлюзи.....	181
6.3.6 Кабельні системи	182
6.3.7 Безпроводні технології.....	184
6.4.Вимоги, що пред'являються до мереж.....	185
6.5. Глобальна мережа Internet (інтернет). WWW.....	189
7. ОСНОВИ АЛГОРИТМІЗАЦІЇ І ПРОГРАМУВАННЯ	222
7.1. Стадії рішення завдань на комп'ютері.....	222
7.2. Алгоритми.	224
Література.....	242

1. АПАРАТНІ ЗАСОБИ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ

1.1 Інформатика та інформаційні технології

Інформатика – комплексна наукова та інженерна дисципліна, що вивчає всі аспекти проектування, створення, оцінювання, функціонування комп'ютерних систем обробки інформації, її застосування та вплив на різноманітні галузі соціальної практики.

Поява інформатики зумовлена появою та розповсюдженням нової технології збирання, обробки та передачі інформації, пов'язаною із фіксацією даних на машинних носіях.

Інформація – сукупність відомостей (даних), які сприймають з оточуючого середовища (вхідна інформація), повертають в оточуюче середовище (вихідна інформація) або зберігають всередині певної системи.

Інформація існує у вигляді документів, креслень, малюнків, текстів, звукових та світлових сигналів, енергетичних та нервових імпульсів і так далі.

Економічна інформація (ЕІ) – це сукупність відомостей економічного характеру, що можна фіксувати, передавати, обробляти, зберігати та використовувати для здійснення функцій керування.

ЕІ містить відомості про склад трудових, матеріальних, грошових ресурсів, про стан роботи об'єктів в певний момент часу, відображує діяльність підприємств та організацій за допомогою натуральних, вартісних та інших показників.

Поняття інформаційних технологій (ІТ) з'явилося з появою інформаційного суспільства, основою соціальної динаміки в якому є не традиційні матеріальні, а інформаційні ресурси: знання, наука, інтелект, здібності, ініціатива, творчість людей. Вперше поняття і перспективу розвитку ІТ детально розглянув академік Глушков В.М.

Інформаційна технологія – це людино – машинна технологія збору, обробки та передачі інформації. Це технологія, яка базується на обчислювальній техніці, швидко розвивається, охоплюючи усі види сучасної діяльності: виробництво, управління, науку, освіту, проектні розробки, торгівлю, фінансово-банківські операції, медицину, криміналістику, охорону оточуючого середовища, побут та інше.

В інформатиці виокремлюють три основні частини:

- Обчислювальну техніку (hardware);
- Алгоритми обробки інформації (algorithm);
- Комп'ютерні програми (software)

Електронні обчислювальні машини (ЕОМ)

Номенклатура видів ЕОМ на сьогодні дуже велика. Машини відрізняються потужністю, розміром, елементною базою, за призначенням і так далі. Під час вибору комп'ютерної техніки для вирішення економічних та ділових задач найважливішим є продуктивність та габаритні характеристики (розмір, маса). Ця класифікація, звичайно, умовна, адже розвиток комп'ютерної науки і техніки настільки стрімкий, що, наприклад, сьогоднішня мікро – ЕОМ не поступається по потужності міні – ЕОМ п'ятирічної давнини. Класифікацію ЕОМ за масогабаритними даними можливо надати у наступній таблиці:

Табл. 1.1 Класифікація ЕОМ за масогабаритними даними

Клас ЕОМ	Основне призначення	Основні технічні дані	Ціна (тис.у.о.)	Моделі і/або виробники
Супер ЕОМ	Складні наукові розрахунки	Швидкодія до десятків мільярдів операцій на секунду; кількість паралельно працюючих процесорів до 100	До 10 000	Cray3; Cray4; VAX – 1000; MULTICON; CYBER 205; Fujitsu VP2000
Великі ЕОМ (мейн – фрейми)	Обробка великих об'ємів інформації в банках, на великих підприємствах	Мультипроцесорна архітектура підключення до 200 робочих місць	До 250	IBM 370; IBM ES/9000; ES 1068; ES 1170
Супер міні-ЕОМ	Системи управління підприємствами, багато пультові обчислювальні системи	Мультипроцесорна архітектура підключення до 200 терміналів, дискові запам'ятовуючі пристрої, що нараховуються до десятків гігабайтів	До 180	Сімейство VAX (Digital Equipment); SPARC (SUN Microsystems); AS/400 (IBM)
Міні-ЕОМ	Системи управління підприємствами середнього розміру; багато пультові обчислювальні системи	Однопроцесорна архітектура, розгалужена периферія	До 100	ES/9370 (IBM); Серії A та 2200 (Unisys)
Робочі станції	САПР, системи автоматизації експериментів	Одно – двох процесорна архітектура, розгалужена периферія	До 50	MEPVA – 2 (IBM RS – 6000); SUN ULTRASPARC; NEXT
Мікро - ЕОМ	Індивідуальне обслуговування користувача, робота в у локальних автоматизованих системах управління	Однопроцесорна архітектура, гнучкість конфігурації, – можливість підключення різноманітних зовнішніх пристроїв	До 10	Широкий перелік моделей та виробників

Клас персональних ЕОМ складається із різноманітних видів машин і тому заслуговує окремої класифікації

Табл. 1.2 Класифікація персональних ЕОМ

Клас персональних ЕОМ	Маса, КГ	Джерело живлення	Примітка
Настільні Desktop	5 – 10 (без монітора)	Побутова електромережа	Використовується в приміщенні для обладнання робочих місць
Переносні Laptop	2,5 - 5	Побутова електромережа або акумуляторні батареї	Призначені для використання в поїздках, забезпечуючи широкі функціональні можливості; можливе підключення до обчислювальних мереж
Блокнотні Notebook	0,7 – 2,5	Акумуляторні батареї або перетворювач напруги	Призначений для використання у поїздках. Забезпечує скорочені функціональні можливості
Електронний секретар, електронна записна книжка (PDA, Personal Digital Assistant)	Менше 0,7	Батареї або перетворювач напруги	Вміщається у кишені, можна тримати в руці. Набір функцій дозволяє записувати текст, деякі обчислення, вести розклад, переводити фрази з іноземної мови та інше

За основними напрямками використання ПК поділяють на три види: домашні (побутові), офісні (ділові), професійні.

Побутові призначені для автоматизації побутової сфери діяльності людини: ведення сімейних баз даних (словників, архівів, щоденників), планування сімейного бюджету, навчання, розваг (ігри) та інше. Мають невисокі характеристики і відповідну комплектацію, яка схиляється до мультимедійної (кольоровий монітор, звукова плата, аудіо система, дисковод CD – ROM та інші).

Офісні орієнтовані на автоматизацію конторської роботи: складання, редагування та оформлення текстів; ведення баз даних; ділове листування; виконання табличних обчислень; робота із графічною інформацією. Мають характеристики не нижче середніх і відповідну комплектацію (сканери, принтери, модеми, мережні карти та інше).

Професійні призначені для автоматизації праці інженерів, наукових працівників у САПР і автоматизованих системах наукових досліджень. Мають високі характеристики загальних пристроїв і необхідний набір висококласних спеціалізованих пристроїв (наприклад, дуже великий монітор, графопобудовник, професійний сканер, цифрову камеру і таке інше).

Питання для самоперевірки

1. Дайте означення інформатики як комплексної наукової та інженерної дисципліни.
2. Які події в суспільстві обумовили виникнення інформатики.
3. Дайте означення інформації. У якому вигляді вона існує?
4. Дайте означення економічної інформації. Які особливості економічної інформації?
5. Дайте поняття інформаційної технології.
6. Яку класифікацію ЕОМ можна дати за масогабаритними даними?
7. Яка класифікація персональних ЕОМ існує в наш час?
8. Яке призначення побутових, офісних та професійних персональних комп'ютерів? Яка робота автоматизується за їх допомогою?

1.2 Персональні комп'ютери. Особливості їх складу, конструкції і конфігурації.

Персональний комп'ютер – загальнодоступна та універсальна у застосуванні настільна або переносна ЕОМ. Це основна технічна база ІТ.

Архітектуру ПК зумовлюють потреби користувача. Головне – це структура та функціональні можливості машини, які можна поділити на основні та допоміжні.

Структура ПК – це модель, що встановлює склад, порядок та принципи взаємодії компонентів.

Основні функції – зумовлюють призначення ЕОМ: обробку та зберігання інформації, обмін інформації із зовнішніми об'єктами.

Допоміжні функції – підвищують ефективність виконання ЕОМ основних функцій: забезпечують ефективні режими її роботи, діалог із користувачем, високу надійність. Ці функції реалізуються за допомогою її компонентів і програмних засобів.

Достоїнствами ПК є:

- Низька вартість;
- Автономність експлуатації без спеціальних вимог до навколишнього середовища;
- Гнучкість архітектури, що забезпечує її адаптивність до різноманітних застосувань у сфері управління, науки, освіти, побуту;
- Дружність ОС та іншого програмного забезпечення, що зумовлює можливість роботи з нею користувача без спеціальної професійної підготовки;
- Висока надійність роботи (більш як 5 тис. годин експлуатації до відмови).

Склад персонального комп'ютера можна представити структурною схемою, зображеною на Рис. 1.1.

Мікропроцесор (МП) – є центральним блоком ПК і призначений для управління роботою усіх блоків комп'ютера і виконання арифметичних і логічних операцій над інформацією. До його складу входять логічні блоки (АЛП, КП) і мікропроцесорна пам'ять (МПП), що забезпечує короткочасне зберігання, запис та видачу інформації, яка використовується в обчисленнях у найближчій такти роботи комп'ютера. МПП будується на регістрах і використовується для забезпечення високої швидкодії ЕОМ, адже основна пам'ять не завжди забезпечує необхідну для швидкодіючого МП швидкість запису, зчитування, пошуку інформації. **Регістри** – найшвидкодійніші елементи пам'яті довжиною 1-4 байти або більше. **Інтерфейсна система** МП забезпечує зв'язок з іншими пристроями ПК (через системну шину). **Генератор тактових імпульсів** – генерує послідовність електричних імпульсів, частота яких зумовлює тактову частоту комп'ютера. Проміжок часу між імпульсами становить такт.

Системна шина – це основна інтерфейсна система ПК, яка забезпечує взаємодію усіх його пристроїв. Складається з шини даних, шини адрес, шини керування і шини живлення, забезпечуючи три напрямки передачі інформації:

- Між МП та оперативною пам'яттю;
- Між МП та портами введення/виведення зовнішніх (відносно процесора та шин) пристроїв;

- Між основною пам'яттю та портами введення/виведення зовнішніх пристроїв (у режимі прямого доступу до пам'яті).

Усі блоки, а точніше їх порти введення/виведення, через відповідні уніфіковані роз'ємні з'єднання підключаються до шини безпосередньо або через пристрої з'єднання – контролери (адаптери). У сучасних ПК керування шиною здійснюється контролером шини.

Основна пам'ять складається з ПЗП та ОЗП. ПЗП призначений для зберігання програми тестування ПК, програми початкового завантаження, базової системи введення/виведення (BIOS). ОЗП служить для оперативного запису, зберігання і зчитування інформації (програм і даних). У ОЗП висока швидкодія звернення до комірок пам'яті (прямий адресний доступ). Після вимикання ПК інформація в ОЗП знищується. Об'єм ОЗП сучасного ПК може складати 16, 32, 64, 128, МБ та більше.

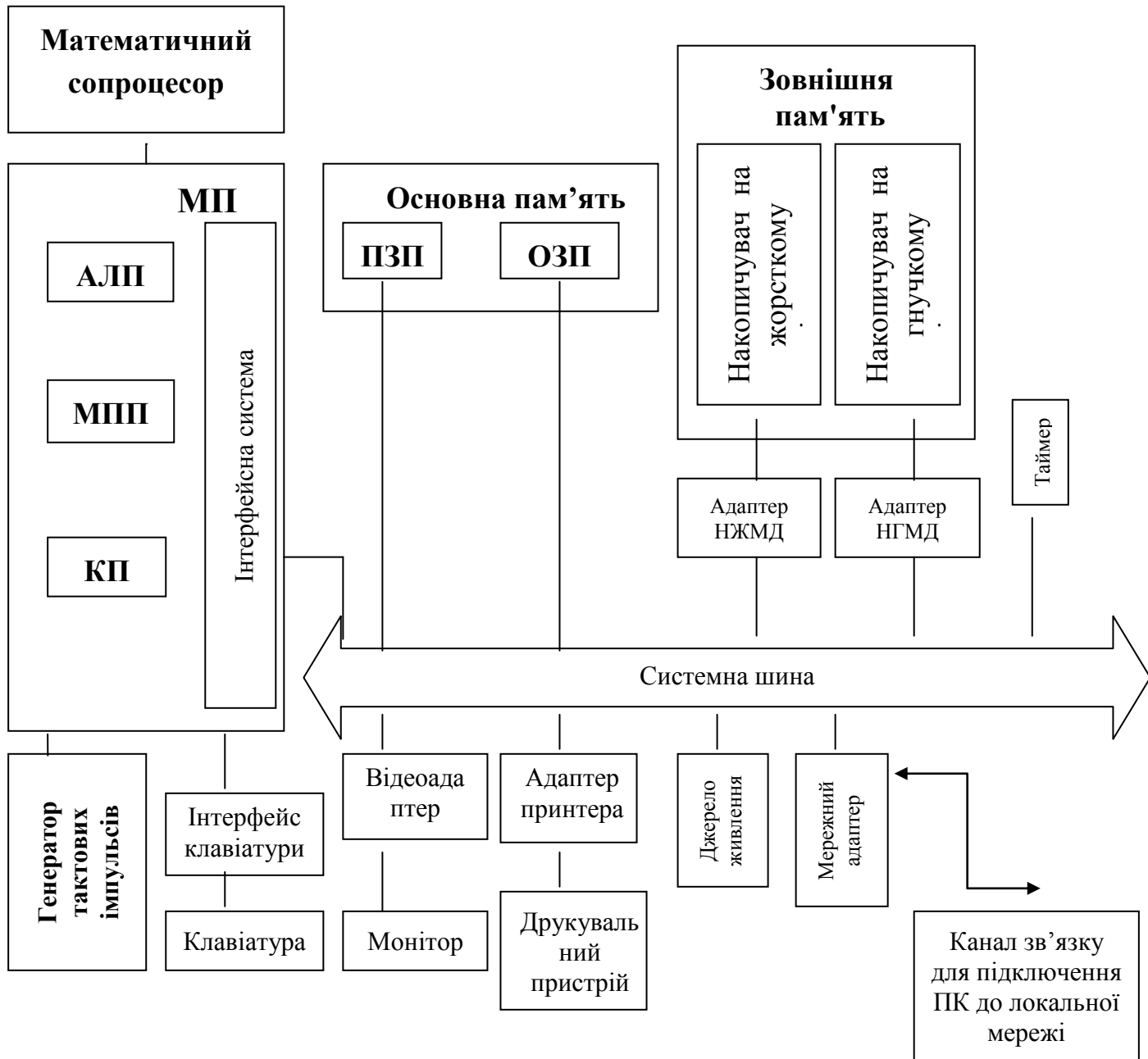


Рис. 1.1 Структурна схема ПК

Зовнішня пам'ять призначена для довготривалого зберігання даних. У зовнішній пам'яті зберігається усе програмне забезпечення ПК. Найпоширеніші пристрої - це накопичувачі на жорстких магнітних дисках (НЖМД) і накопичувачі на гнучких магнітних дисках (НГМД). Відрізняються НЖМД та НГМД тільки конструктивно, об'ємом інформації, яку вони можуть зберігати, часом пошуку, запису, зчитування. Також використовуються касетні магнітні стрічки, стримери, накопичувачі на оптичних дисках (CD – ROM – компакт-диск), флеш диски і таке інше.

Джерело живлення – це блок, який вміщує системи автономного і мережного енергоживлення ПК.

Таймер – внутрішній електронний годинник. Під час вимкнення ПК підключається до автономного джерела живлення і продовжує працювати.

Зовнішні пристрої за призначенням поділяються на такі групи:

- Зовнішні запам'ятовуючі пристрої виконані як окремий конструктивний блок і, як правило, із власним блоком живлення. Вони мають велику ємність, іноді складаються з декількох накопичувачів в одному корпусі (наприклад, пристрій для перезапису компакт-дисків)
- Діалогові засоби користувача: монітор – засіб для відображення інформації; мікрофонні акустичні системи зі складним програмним забезпеченням, яке дозволяє розпізнавати слова, перетворювати мову на текст і так далі; синтезатори звуку, які перетворюють цифрові коди на літери і слова. Мова відтворюється за допомогою динаміків або звукових колонок, з'єднаних з ПК.
- Пристрої введення інформації: клавіатура; графічні планшети (діджитайзери) – пристрої для ручного введення графічної інформації, зображень за допомогою переміщення по планшету пера (вказівника) з одночасним автоматичним зчитуванням координат його місцезнаходження та введення цих координат у ПК; маніпулятори – це пристрої вказівники – джойстик, миша, трекбол, світлове перо та інші; сенсорні екрани – пристрої для введення окремих елементів зображення, програм або команд з поліекрану дисплея у ПК; сканери- пристрої для автоматичного причитування з паперових носіїв і введення в ПК текстів, графіків, малюнків, креслень.
- Пристрої виведення інформації: принтери, графопобудовники (плотери) – пристрої для виведення графічної інформації (графіки, креслення, малюнки).
- Пристрої зв'язку і телекомунікацій: використовуються для зв'язку з пристроями і іншими засобами автоматизації (інтерфейсні адаптери, цифро – аналогові і аналого – цифрові перетворювачі та інше) і для підключення ПК до каналів зв'язку, іншими ПК і комп'ютерними мережами (мережеві інтерфейсні плати, модеми).

Засоби мультимедіа – це комплекс апаратних і програмних засобів, які дозволяють людині спілкуватися з комп'ютером, використовуючи звук, відео, графіку, тексти, анімацію та інше. До них належать: сканери; відео – і звукові плати; плати відео захвату (video grabber), які знімають зображення з відеоманітофону або відеокамери і вводять його в ПК; акустичні та відео відтворюючі системи з підсилювачами, звуковими колонками, великими відеоекранами. До засобів мультимедіа відносяться також запам'ятовуючі пристрої великої ємності на оптичних дисках, які використовуються для запису аудіо і відеоінформації.

Додаткові схеми - до системної шини і МП можуть бути підключені також додаткові плати з інтеграційними мікросхемами, які розширюють та покращують її функціональні можливості: математичний сопроцесор, контролер прямого доступу до пам'яті, сопроцесор введення/виведення, контролер переривань та інше.

- **Математичний сопроцесор** використовується для прискорення виконання операцій над числами, а також для обчислення деяких трансцендентних, у тому числі тригонометричних, функцій. Математичний сопроцесор працює паралельно з МП, але під його керівництвом. Прискорення операцій відбувається в декілька разів. Останні моделі ПК, починаючи з МП 486 DX, мають сопроцесор.

- **Контролер прямого доступу до пам'яті** звільняє МП від прямого керівництва накопичувачами на магнітних дисках, що суттєво підвищує ефект швидкодії ПК. Без контролера обмін даними між зовнішніми запам'ятовуючими пристроями і ОЗП здійснювався через регістри МП.
- **Сопроцесор введення/виведення** завдяки паралельній роботі з МП значно прискорює виконання процедур введення/виведення інформації при обслуговуванні декількох зовнішніх пристроїв (дисплей, принтер, НЖМД, НГМД).

- **Контролер переривань**

Переривання – це тимчасова зупинка виконання однієї програми з метою оперативного виконання іншої, більш пріоритетної на даний момент. Переривання виконується постійно. Контролер переривань обслуговує процедури переривань, приймає запит на них від зовнішніх пристроїв, визначає рівень пріоритету цього запиту і передає сигнал переривання в МП, який, отримавши цей сигнал, зупиняє виконання поточної програми, переходячи на виконання спеціальної програми обслуговування того переривання, яке запитав зовнішній пристрій. Після завершення програми обслуговування відновлюється виконання перерваної програми.

Розглянута структурна схема ПК є спрощеною, вона стосується першого (РС, ХТ) і частково другого (перші машини АТ) поколінь, відрізняючись від структурних схем сучасних ПК реалізацією системного інтерфейсу в середині машини.

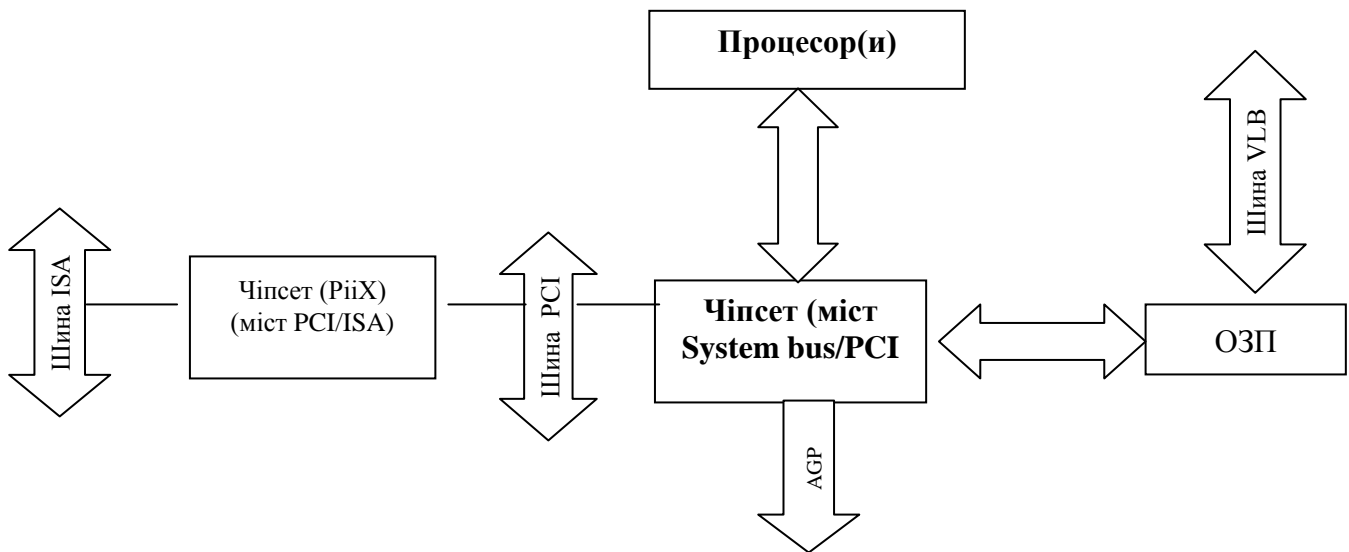


Рис. 1.2 Схема реалізації інтерфейсу сучасного ПК

Процесор, оперативна та кеш – пам'ять сполучені між собою найшвидкодійною і найширшою шиною (66 – 133 МГц, 64 біт). Саме цю область ПК називають системною шиною. Мікросхеми чіпсету надвеликої інтеграційної схеми керування на системній платі виконують функції мостів між шинами з більшою і меншою пропускними можливостями.

Питання для самоперевірки

1. Дайте означення структури ПК.
2. Призначення мікропроцесора.
3. Які логічні блоки входять в мікропроцесор?
4. Призначення мікропроцесорної пам'яті.
5. Призначення генератора тактових імпульсів.
6. Яке призначення системної шини? Які напрямки передачі інформації вона забезпечує?
7. Призначення ПЗП та ОЗП.
8. Які пристрої представляють зовнішню пам'ять? Призначення зовнішньої пам'яті.

9. Дайте класифікацію зовнішніх пристроїв по їх призначенню.
10. Зовнішні запам'ятовуючі пристрої. Їх призначення.
11. Діалогові засоби користувача.
12. Пристрої введення інформації. Принципи їх дії.
13. Пристрої виведення інформації. Їх призначення.
14. Пристрої зв'язку та телекомунікації. Їх призначення.
15. Засоби мультимедіа. Їх призначення. Які пристрої належать до засобів мультимедіа.
16. Призначення додаткових схем. Яке призначення математичного сопроцесора, контролера прямого доступу до пам'яті, сопроцесора введення-виведення, контролера переривань?

1.3 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: системний блок, блок живлення, мікропроцесор, материнська плата, оперативна пам'ять

ПК складається з декількох блоків, які з'єднані між собою кабелями. Склад блоків може мінятися, але мінімальний комплект складають: системний блок, клавіатура, монітор, маніпулятор (миша). У числі додаткових пристроїв можуть бути: принтер, сканер та інші.

Системний блок – прямокутний каркас з кришкою або кожухом, в якому розміщені усі основні вузли ПК: материнська плата, адаптери, блок живлення, один або два НГМД, один (або більше) НЖМД, динамік, дисковод для компакт-дисків, органи керування. Серед органів керування можуть бути: кнопка ввімкнення / вимкнення електроживлення (на передній панелі); кнопка загального скидання RESET; перемикач тактової частоти; кнопка («сну»), яка дозволяє знизити енергоспоживання, коли ПК не використовується; індикатори живлення та режимів роботи. З тильного боку системного блоку розташовані штепсельні роз'єми – порти для підключення шнурів живлення і кабелів зв'язку із зовнішніми пристроями. В середині системного блоку розташовані плати поєднання пристроїв з ЦП та іншими пристроями на материнській платі (адаптери або контролери і плати розширення).

Існують також формати системного блоку: вежа (tower), міні-вежа (mini – tower), плоский (desktop), надплоский (slim, ultraslim). Багато сучасних корпусів мають формат ATX. Вони відрізняються декілька більшими габаритами, але разом з тим більш «інтелектуальною» системою живлення, роз'ємними з'єднаннями для миші і клавіатури стандарту PS/2.

Блок живлення перетворює змінний струм стандартної мережі електропостачання (220В, 50Гц) на постійний струм низької напруги. Він має декілька виходів на різноманітні напруги (12 та 5В), які забезпечують живленням відповідні пристрої. Звичайна потужність блоків живлення ПК складає 150 – 230Вт для мережного сервера. Вона може бути значно більше.

Системна (материнська) плата – велика дукована плата, що містить на собі головні компоненти комп'ютерної системи: ЦП; ОП; кеш - пам'ять; комплект мікросхем логіки, які підтримують роботу плати чіпсет; центральну магістраль або шину; контролер шини і декілька роз'ємних гнізд (слотів), які служать для підключення до материнської плати інших плат (контролерів, плат розширення та інше). Кількість та вид слотів є однією з найважливіших характеристик системної плати, адже в подальшому ПК можливо доукомплектовувати або модернізувати. На материнській платі знаходяться перемикачі (switches), за допомогою яких відбувається налаштування плати.

Ще один важливий елемент, який встановлюється на системній платі – мікросхема BIOS (базова система введення/виведення). Вона є ПЗП, в якій записані програми, які реалізують функції введення/виведення; програму тестування ПК в момент його включення (POST, Power On Self Test), програму налаштування параметрів BIOS та системної плати та інші спеціальні програми. В роботі BIOS використовують інформацію про апаратну конфігурацію комп'ютера, яка зберігається на ще одній мікросхемі CMOS RAM. Ця пам'ять постійно підживлюється від батарейки, яка також знаходиться на системній платі. Ця батарейка також живить схему кварцового годинника.

Мікропроцесор. На одному кристалі надчистого кремнію за допомогою складного, багатоступеневого і високоточного технологічного процесу створено кілька мільйонів транзисторів та інших елементів, з'єднувальні проводи і вузли і точки підключення зовнішніх виводів. У сукупності вони створюють усі логічні блоки, тобто арифметичний пристрій, КП, регістри та інше. Лідером з виробництва ЦП є компанія INTEL

Табл. 1.3. Технічні дані деяких ЦП компанії INTEL:

Вид ЦП	Покоління	Кількість транзисторів на кристалі, тис	Розрядність основних регістрів	Розрядність шини даних	Розрядність шини адрес	Пам'ять, що адресується	Тактова частота, МГц
I8086	1	29	16	16	20	1Мб	4,77,8,10
I80286	2	134	16	16	24	16Мб	8,12,16
I80386DX	3	275	32	32	32	4Гб	20,25,33,40
I486DX	4	1200	32	32	32	4Гб	25-100
Pentium	5	3100	32	64	32	4Гб	60-200
Pentium MMX	5		32	64	32	4Гб	166-233
Pentium Pro	6	5500	32	64	36	64Гб	160-200
Pentium II	6		32	64	36	64Гб	233-500
Celeron	6	7500	32	64	36	64Гб	233-766 і вище
Pentium III	6		32	64	36	64Гб	450-933 і вище

Основними параметрами МП є: набір команд, розрядність, тактова частота.

Набір команд (або система команд) постійно вдосконалюється, з'являються нові команди, які змінюють серії примітивних команд – мікропрограми. На виконання нової команди потрібно менша кількість тактів ніж на мікропрограму. Сучасні МП можуть виконувати до декількох сотень команд (інструкцій).

Розрядність – показує, скільки двійкових розрядів (бітів) інформації обробляється або передається за один такт, а також скільки двійкових розрядів може бути використано у МП для адресування оперативної пам'яті, передачі даних та інше. Кількість пам'яті, що адресується, (адресний простір) залежить від кількості ліній шини адреси МП. Якщо таких ліній 20, то адресний простір складатиме $2^{20} = 1$ Мбайт, якщо ліній 24, то $2^{24} = 16$ Мбайт і так далі.

Тактова частота вказує, скільки елементарних операцій (тактів) МП виконує за секунду, вимірюється у мегагерцах (1МГц = 1000000 Гц). Тактова частота є тільки відносним показником продуктивності МП. Через архітектурні відмінності у деяких з них за один такт виконується робота, на яку інші витрачають кілька тактів. Важливими характеристиками сучасних МП, що впливають на їхню продуктивність, є ємність і швидкість функціонування вмонтованої кеш-пам'яті, оскільки тактова частота МП в кілька разів вища ніж частота синхронізації системної шини, по якій відбувається обмін інформацією з відносно повільним ОЗУ. Без внутрішньої кеш – пам'яті (яка має особливо високу швидкодію) МП часто працював би вхолосту, чекаючи чергової інструкції від ОЗП або закінчення операції запису у пам'ять. Кеш – пам'ять першого рівня (L1) реалізована в самому кристалі МП, а кеш – пам'ять другого рівня (L2) більш повільна і більшої ємності, може знаходитися на системній платі. Типова для сучасного МП ємність кеш-пам'яті L1 складає 32 Кбайт (по 16 Кбайт для копіювання команд та

даних); ємність L2 від 256 або 512 Кбайт для звичайних ПК до 2-3 Мбайт для потужних серверів.

Зовні МП виглядає, як прямокутна пластмасова або керамічна пластина розміром 5x5x0,3. На сучасні МП встановлюються радіатор зі сплаву з високою теплопровідністю та вентилятор для охолодження. МП легко встановлюється у спеціальне прямокутне рознімне з'єднання – сокет. Тип сокету має відповідати типу МП.

Оперативна пам'ять реалізована на надвеликих інтегральних схемах. Існує два типи мікросхем пам'яті: статична та динамічна. У перших елементарну комірку утворюють тригерні схеми. Така схема встановлюється в 0 або в 1 вхідним імпульсом і зберігає свій стан до наступного імпульсу або вимкнення ПК. При причитуванні записаного в комірку значення її стан також не змінюється. Динамічна пам'ять складається із мікроскопічних конденсаторів, кожний з яких може знаходитися у стані «заряджений» (1) або ні (0). Конденсатори необхідно підживлювати. Тому динамічний ОЗП більш повільний, але він менш енергоємний. Дані у динамічному ОЗП також знищуються після вимкнення ПК. Конструктивно ОЗП (типу SIMM, Single in-line Memory Module) має вигляд невеликої довгастої друкованої плати з розміщеними на ній мікросхемами. В останній час використовують в основному 72-контактні (72-pin) 36-бітові модулі. На сучасних системних платах використовують в основному 168 – контактні 64 – розрядні модулі DIMM, що надає можливість краще використовувати можливості системної шини. На системних платах є два типи рознімних з'єднань: SIMM та DIMM. Швидкість доступу (час циклу зчитування з ОЗП або запису в нього даних) складає 50-70 нс для асинхронної пам'яті (SIMM) і 15-45 нс для синхронної (DIMM). Розрядність ОЗП може сильно позначатися на продуктивності системи. Тому 32-розрядні додатки Windows треба використовувати на 32-розрядному ПК, а не, наприклад, на 16-розрядному.

У сучасному ПК використовується новий тип пам'яті EDO (Extended Data Output), в якій використовується принцип конвеєризації: під час читання записаного в банк байта мікросхема також виконує вибірку наступного байта або зберігає його у вихідному регістрі, завдяки чому час вибірки наступного значення значно скорочується. Загальний вигреш у швидкодії 5-15%. З 1997 року на вітчизняному ринку з'явилися модулі ОЗП типу CD-RAM (синхронний динамічний ОЗП), доступ до якого може здійснюватися ще швидше. Модулі CD-RAM мають ємність 16, 32, 64, 128, 256 Мбайт.

Системна магістраль (шина) та шина розширення

Системна магістраль (шина) – це група електричних з'єднань (провідників) для обміну даними, адресами, сигналами між різними агрегатами комп'ютера.

Системна магістраль поєднує МП з оперативною пам'яттю та кеш-пам'яттю і, як правило, є найбільш швидкодіюю з магістралей. Системні шини працюють на частоті 66, 75, 100, 133 МГц. Більшість IBM-сумісних комп'ютерів мають шину розширення стандарту ISA (Industry Standard Architecture – стандартна індустріальна архітектура). Шина містить 16 ліній для передачі даних, 24 ліній для передачі адрес, 15 ліній для апаратних переривань і 7 ліній для організації прямого доступу до пам'яті. Крім того, кілька провідників призначені для розведення електроживлення та службових сигналів. Шина працює на частоті 8 МГц. Максимальна швидкість передачі даних по шині сягає 16 Мб/с.

У сучасних ПК проміжне місце між системною шиною та шиною ISA займає Mezzanine-шина, тобто шини – прибудови. Вони не залежать від типу ЦП та його тактової частоти. Найпоширенішим стандартом на такі шини є PCI (Peripheral Component Interconnect, зв'язок периферійних компонентів). До шин PCI можна підключати до 10 пристроїв. На шині PCI може бути 124-контактне (32-розрядна передача даних) або 188-контактне (64-розрядна передача) рознімне з'єднання, при цьому теоретично можлива швидкість обміну даними складає відповідно 132 та 264 Мб/с. Стандартна тактова частота шини – 33 МГц. Одним із головних користувачів шини є графічний адаптер. Для використання сучасних графічних програм, які надто перевантажують шину, передбачено спеціальну шину AGP (Accelerated Graphic Port – прискорений графічний порт), розроблену фірмою Intel. Ця шина схожа на PCI шину, але працює на частоті 66 МГц і має архітектурні особливості.

Плати розширення. Шини розширення на системній платі мають вигляд 8-, 16- або 32-розрядних рознімних з'єднань розширення (слотів), в які вставляють плати розширення. Конструктивно плата розширення – це друкована плата стандартного розміру з рознімним з'єднанням типу ISA та PCI. На платі встановлено необхідні мікросхеми. Плата розширення встановлюється у вільний слот системної плати. Іноді платами розширення називають тільки додаткові електронні пристрої (модем, звукову плату та інше), які не входять до комплексу поставки ПК.

Питання для самоперевірки

1. Які основні вузли ПК знаходяться в системному блоці? Які формати системного блока ви знаєте?
2. Призначення блоку живлення.
3. Які компоненти комп'ютерної системи знаходяться на системній платі?
4. Опишіть мікропроцесор з точки зору конструктивного виконання. Які технічні дані характеризують мікропроцесор?
5. Дайте характеристику таким основним параметрам мікропроцесора як набір команд, розрядність, тактова частота.
6. Дайте характеристику мікросхемам статичної та динамічної пам'яті.
7. Який принцип роботи використовується в пам'яті типу EDO?
8. Призначення системної шини.
9. Яке призначення шини стандарту ISA?
10. Яке призначення шини AGP?
11. Яке призначення плат розширення?

1.4 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: накопичувачі, адаптери

Накопичувачі - це запам'ятовуючі пристрої, призначені для довготривалого зберігання великих об'ємів інформації, коли ємність накопичувача в десятки, сотні разів перевищує ємність ОЗП або взагалі необмежена (якщо мова іде про змінні носії). Накопичувач потрібно розглядати як сукупність носія та відповідного приводу. Привід – це об'єднання механізму читання – запису із відповідними електронними схемами керування. Його конструкція визначається принципом дії і видом носія. Носій, що є середовищем зберігання інформації, за зовнішнім виглядом може бути дисковим або стрічковим; за принципом запам'ятовування – магнітним, магнітно – оптичним, оптичним.

Інформація на дискових носіях зберігається у секторах (як правило, по 512 байтів). Сектори розташовуються вздовж концентричних кіл – доріжок. Для дискети запис ведеться на двох сторонах, тому сукупність доріжок з однаковими номерами називається циліндром. Сектори і доріжки утворюються під час форматування носія.

НГМД – це пристрій, який працює із змінним 3,5 дюймовим (89мм) носієм (зроблені зі спеціальної плівки -майлара), який вкритий феромагнітним шаром і поміщений у спеціальний конверт (оболонка дискети). Стандартна ємність дискети 1,44 Мбайт. Флеш-дискети вставляється у спеціальний роз'єм на задній або передній панелі системного блоку і мають ємність від 500 Мбайт.

НЖМД – це пристрій з незмінним носієм. НЖМД повинен забезпечувати у сотні разів більшу ємність та швидкість обміну даними. Тому інформація записується не на один, а на набір дисків, який складається із декількох пластин, ідеально плоских із відполірованим феромагнітним шаром. При цьому запис відбувається на обидві поверхні кожної пластини (окрім крайніх). Пакет дисків обертається без зупинки і з великою частотою (до 7500, а в окремих моделях до 10000 об/хв) доки ПК ввімкнений. Десятки фірм виробляють накопичувачі. Для забезпечення взаємозамінності розроблені стандарти на їхні габарити та електронні характеристики. Найрозповсюдженіші стандарти IDE (Integrated Drive Electronics) та SCSI (Small Computer Sysytem Interface, читається як «сказі»).

Табл. 1.4 Технічні характеристики НЖМД

Модель НЖМД	Фірма – виробник	Інтерфейс	Ємність, Мбайт	Частота обертання, об/хв.	Час доступу, мс
M2714	Fujitsu	IDE	1080	3600	12
M2952		SCSI	2200	7200	8
M2949		SCSI	9100	7200	10
HP SuperStore 1080A	Hewlett – Packard	IDE	1080	4480	12
HP SureStore 1600D		IDE	1626	4480	12
HP SureStore 2000LP		SCSI	2140	5400	9,5
Cabo CFS1081A	Seagate	IDE	1080	3600	14
Medalist 2140		IDE	2113	5400	10
Cabo CFP1080		SCSI	1080	5400	11
Barracuda 4LP		SCSI	2250	7200	8
Bigfoot 1280	Quantum	IDE	1280	3600	15,5
Sirocco 1700		IDE	1700	5400	10
Atlas XP34300		SCSI	4300	7200	8,5

На швидкодію НЖМД впливають такі характеристики: частота обертання шпинделя; ємність кеш – пам'яті – в усіх сучасних дискових накопичувачах встановлюється кеш – буфер, який дозволяє прискорити обмін даними з жорстким диском (ємність буфера від 64 Кбайт до 2 Мбайт); час пошуку (час доступу) – це час у мілісекундах, необхідний для позиціонування блока голівок на потрібний циліндр; час затримки – це час від моменту позиціонування блока голівок на потрібний циліндр до позиціонування конкретної голівки на конкретний сектор; швидкість обміну – кожен жорсткий диск має два таких показники – перший визначає швидкість пересилання даних між пластинами диска і кеш – буфером (внутрішня швидкість обміну), а другий – між кеш - буфером диска і контролером на материнській платі (зовнішня швидкість). Зовнішня швидкість залежить від режиму, який використовується. В режимі РІО використовуються регістри ЦП. Робота в режимі DMA (Direct Memory Access) дає можливість безпосередньо передавати дані від диска до ОП, міняючи ЦП.

Накопичувачі CD-ROM - здатні тільки прочитувати дані, занесені на диск. Мають велику ємність (640 Мб) та високу швидкість зчитування. На відміну від вінчестера, доріжки якого мають вигляд концентричних кіл, компакт – диск має одну доріжку у формі спіралі. Зчитування інформації відбувається за допомогою лазерного променя.

Схожими на CD-ROM та сумісними з ними за форматом запису є накопичувачі CD-R, вони дозволяють зробити один запис на спеціальний диск, а потім необмежену кількість зчитувань. Накопичувачі CD-R сумісні зі звичайними компакт – дисками, але дають можливість формувати, записувати, перезаписувати та читати спеціальні диски.

Накопичувачі DVD – пристрій для читання цифрових відеозаписів, дозволяє перезапис інформації, можливо розміщувати до 17 Гбайтів інформації.

У роботі накопичувачів приймає участь контролер дисків - він керує роботою механічно рухомих частин та формує електричні імпульси під час запису та зчитування інформації. Він містить:

- Генератор, який живить перемінним струмом двигун дисків;
- Сервосистему, що керує пристроєм позиціонування блоку голівок на потрібну доріжку (циліндр);
- Підсилювачі запису, які формують електричні імпульси, що подаються на магнітні голівки під час запису інформації;
- Підсилювачі зчитування.

Адаптери. Форми подання даних та керуючих сигналів, що використовуються у різних пристроях ПК, істотно різні, оскільки різними є функції пристроїв, фізичні принципи їх роботи. Для підтримання взаємодії різних пристроїв необхідно виконувати перетворення форм подання

інформації – для цього використовуються адаптери. Конструктивно – це друкована плата, яка з одного кінця має стандартний роз'єм для з'єднання із шиною, а з другого – специфічний роз'єм для зв'язку з відповідним пристроєм. На цій платі знаходяться мікросхеми, що виконують відповідні перетворення. Сформувалась тенденція відмовитись від адаптерів, оскільки функції адаптера переносяться безпосередньо на самий пристрій (наприклад, для накопичувачів), деякі функції адаптера забезпечують мікросхеми, встановлені на системній платі. Зараз стійко фігурують: відеоадаптери (вони ж – відеокарти, відеоплати), адаптери портів введення – виведення, мережні адаптери (карти), звукові плати (аудіокарти), внутрішні модеми.

Відеоадаптер (відеокарта) – пристрій, який перетворює набір даних, які характеризують зображення на екрані монітора, на відеосигнал, що передається по кабелю. Відеоадаптер встановлюється у слот на системній платі. Зображення на екрані формується окремими точками (пікселями), чим більше їх кількість, тим вища якість зображення: цю характеристику монітора, а також відеоадаптера називають роздільною здатністю і виражають у вигляді добутку: *число точок у рядку x кількість рядків*. Зображення на екрані поновлюється як мінімум 25-30 разів на секунду, кожен раз відеосигнал формується заново, звертаючись до оперативної пам'яті за початковими даними. Частоту поновлення екрану підвищують до 100 Гц та більше, щоб послабити мерехтіння екрану. Щоб цей процес не заважав роботі ЦП, необхідні для введення та виведення на екран дані зберігаються у спеціально відведених для цього відеопам'яті – в ОЗП або у складі відеоадаптера присутні спеціальні мікросхеми пам'яті (для сучасних ПК). Вимоги до ємності відеопам'яті зростають із збільшенням роздільної здатності та кількості кольорів.

Табл. 1.5 Технічні дані деяких адаптерів

Модель відеоадаптера	Фірма – виробник	Ємність відеопам'яті	Роздільна здатність
S3 Trio 64 V+	S3	1 або 2	1280 x 1024
S3 Virge		2 або 4	1600 x 1200
Stealth 64 Video 3000	Diamond Multimedia	2 або 4	1600 x 1200
MGA Millennium	Matrox Graphics	2,4 або 8	1600 x 1200
Matrox Mystique		2 або 4	1280 x 1024
Imagine 128 Series	Number Nine	2 або 4	1600 x 1200
9FX Motion 331		1 або 2	1280 x 1024

Швидкість роботи – важливий показник відеоадаптера. Чим «вище» відеорежим за кольоровістю та за роздільною здатністю, тим більший об'єм даних потрібно передавати у відеопам'ять та зчитувати з неї.

Відео карти стандарту PCI та AGP забезпечують необхідну швидкість. У сучасних відео картах використовується графічна акселерація – режим, при якому робота ЦП під час побудови окремих елементів зображення (лінії, прямокутники, графічний курсор) передається спеціалізованому процесору карти. Окрім таких «прискорених» відео карт, популярними є спеціальні карти – прискорювачі, що виконують тільки цю функцію і працюють спільно з відео картою – 3D-акселератори.

Табл. 1.6 Стандарти на відеоадаптери

Назва стандарту	Роздільна здатність графічного режиму, пікселів	Кількість кольорів, що відображаються	Необхідна ємність відео пам'яті	Частота горизонтальної розгортки, кГц	Частота вертикальної розгортки, Гц	Примітка
CGA	320x200, 640x200	2-16	16 Кбайт	15	50 interlaced	У наш час не використовуються
EGA	До 640x350	До 16 із 64	64-256 Кбайт	15;22	60	У наш час не використовуються
VGA	До 640x480	До 256 із 4096	256 Кбайт	31,5 та ін.	60	Виходить із використання

SVGA	800x600, 1024x768, 1280x1024 і вище	До 16,7 млн	Від 512 Кбайт до 8 Мбайт і більше	31,5 та ін. (залежно від режиму)	70-120 і вище	Сучасний відео стандарт
------	--	----------------	--	--	------------------	----------------------------

Адаптер портів (контролер введення-виведення) – це пристрій, що обслуговує різні зовнішні пристрої (принтери, маніпулятори і таке інше). Розрізняють паралельні та послідовні порти. Паралельний порт дозволяє передавати за один такт 1 байт, оскільки для передачі кожного біта відводиться один провідник (контакт) і, таким чином, усі складові байта передаються одночасно – паралельно. Послідовний порт має тільки одну пару провідників, і тому біти одного байта проходять через порт послідовно. Паралельні порти іменують LPT-1 – LPT-3, а послідовні – COM1-COM4.

Мережні адаптери (плати) – призначені для з'єднання ПК з фізичним каналом передачі даних, наприклад, із коаксіальним кабелем. Вони здійснюють прийом сигналу із каналу і передачу його на шину ПК або навпаки. Існують мережні карти для різних типів слотів (ISA, EISA, VESA, PCI). Сучасні карти підтримують швидкість роботи до 100 Мбіт за секунду.

Питання для самоперевірки

1. Дайте поняття накопичувача як сукупності носія та приводу.
2. Яким чином зберігається інформація на дискових носіях?
3. Конструкція НЖМД та його ємність.
4. Які характеристики впливають на швидкодію НЖМД?
5. Дайте характеристику НГМД.
6. Накопичувачі CD-ROM.
7. Особливості накопичувачів CD-R та CD-RW.
8. Накопичувачі DVD.
9. Призначення контролерів дисків.
10. Призначення адаптерів.
11. Призначення відеоадаптера. Дайте поняття роздільної здатності відеоадаптера.
12. З якою метою використовується режим графічної акселерації в відеокартах?
13. Яка різниця між послідовними та паралельними портами введення та виведення інформації?
14. Призначення мережних адаптерів.

1.5 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: звукові карти, монітор, клавіатура

Звукові карти – ПК має стандартний канал керування звуком Speaker (спікер), розрахований на підключення невеликого динаміка. Звук формується з тонального сигналу від таймера, роботою якого можна програмно керувати. Частоту (тон) сигналу можна змінювати. Фактично стандартизованим засобом для роботи з аудіо сигналом є сучасний звуковий аудіо канал, реалізований на звукових платах. Одним із перших таких пристроїв була плата Sound Blaster. Цифровий аудіоканал забезпечує можливість моно– та стереофонічного запису та відтворення аудіо файлів. Запис проводиться оцифруванням (аналого – цифровим перетворенням) вибірок миттєвого значення сигналу з частотою дискретизації 5 – 44,1 кГц. Розрядність перетворювачів складає 8,16,32 біта та більше. Якість звуку тим вища, чим більша частота дискретизації та розрядність. Для передачі потоку даних по шині у звукових платах використовуються канали DMA і тому цими даними ЦП не завантажується.

Окрім цифрового аудіо каналу на звукових платах також є:

- Мікшер, який змішує та регулює вхідні сигнали від різних пристроїв (мікрофона, CD-ROM, цифрового аудіоканала та інших);

- Еквалайзер, який регулює тембр на низьких та високих частотах;
- Синтезатор (MIDI – синтезатор), який забезпечує імітацію різноманітних музичних інструментів (є FM-синтезатори та більш якісні і відповідно більш дорогі хвильові WaveTable – синтезатори);
- MIDI-порт для підключення електромузичних інструментів. Часто рознімне з'єднання цього порту виконує функцію ігрового порту, встановленого на платі, і використовується для підключення ігрових маніпуляторів - джойстиків;
- Вбудований підсилювач, до рознімного з'єднання якого підключають навушники або колонки;
- Рознімні з'єднання для підключення мікрофону, CD-ROM, лінійні вхід та вихід.

Монітор - пристрій для відображення текстової та графічної інформації. В текстовому режимі монітор умовно розбивається на окремі ділянки – знакомісця (найчастіше – на 25 рядків по 80 символів у кожному). На кожному знакомісці може бути відображений один із 256 заделегідь визначених символів. Загальна номенклатура символів не обмежується числом 256. Одному й тому ж коду можуть відповідати різні символи в залежності від записаної в пам'ять відеоадаптера таблиці (латинь, кирилиця, псевдографіка та інше).

В графічному режимі екран монітора - це по суті растр, який складається із точок (пікселів). Роздільна здатність зумовлюється розміром точки (зерна) екрану. Реальна роздільна здатність залежить від режиму, заданого відеокартою. Розмір зерна сучасного екрану не перевищує 0,28 мм при вищій характеристиці істотно зростає вартість пристрою. Якість зображення залежить також від діагоналі екрану. Розповсюджені монітори із діагоналлю 14,15 та 17 дюймів. Для якісного зображення вистачає режиму 800x600, 15 дюймів, 0,28 мм. Важливою характеристикою моніторів є частота регенерації зображення або частота кадрової розгортки (refresh rate). Чим вище цей показник, тим менше буде помітне мерехтіння зображення і тим менше будуть втомлюватися очі користувача. Підвищення частоти регенерації встановлює достатньо жорсткі вимоги до апаратного монітору; так, у моніторах попередніх поколінь для режимів з максимальною розрізняльною здатністю застосовувалась розгортка через рядок (interlace), як у звичайних телевізорів, так, що зображення на екрані формувалося за два проходи променя: за перший прохід він утворював парні ряди, за другий – непарні. При цьому підсилювалося мерехтіння зображення, а якість його була недостатньою. Тому сучасний монітор, який працює у режимі високої розрізняльної здатності, повинен також працювати у режимі прогресивної розгортки (non - interlaced; NI). Кращі моделі моніторів забезпечують частоту регенерації до 200 Гц.

Маса та геометричний об'єм рідкокристалічних моніторів на порядок менша, вживають на два порядки менше енергії, але вони приблизно в 5 разів дорожчі, оскільки застосовуються в основному, у переносних комп'ютерах.

В останніх моделях моніторів передбачено цифрову настройку з пам'яттю для 10-20 відорежимів. Це необхідно для роботи у різних графічних режимах, як правило, з розрізняльною здатністю 800x600 пікселів та більше, оскільки у протилежному разі користувач при зміні режиму повинен би був постійно регулювати геометричні характеристики зображення.

Клавіатура – це пристрій, призначений для введення у комп'ютер інформації та команд керування. Клавіатура сучасних ПК, як правило, - це самостійний конструктивний блок. У переносних ПК клавіатура входить у склад корпусу. Клавіатура має 101-104 клавіші, розташовані за стандартом QWERTY (у верхньому лівому куті клавіатури знаходяться клавіші Q, W, E, R, T, Y).

Усю сукупність клавіш клавіатури розбито на декілька груп:

- Символьні (алфавітно – цифрові);
- Цифрові;
- Функціональні;

- Керування курсором;
- Спеціальні.

Основне призначення символічних клавіш – введення тексту (літер, цифр, спеціальних символів). Група цифрових клавіш призначена для введення чисел. Призначення функціональних клавіш – подавати команди. Призначення команд в повній мірі зумовлюється активною в даний момент програмою (12 функціональних клавіш). Наприклад, F1 - допомога при роботі конкретної програми.

Клавіші керування курсором дають команди на його пересування по екрану відносно поточного зображення. Конкретне значення команд може залежати від програми.

Клавіші зі стрілками призначені для керування курсором <Page Up> та <Page Down> для прокручування тексту на сторінку вгору або вниз, <Home> встановлює курсор на початок рядка, <End> - на кінець.

Спеціальні клавіші: для введення літер у верхньому регістрі використовується клавіша <Caps Lock>, <Enter> - для введення команди у комп'ютер або перехід на новий рядок. Клавіші <Ctrl>, <Alt> та <Shift> використовують разом з іншими клавішами. Значення комбінацій цих клавіш залежить від конкретної програми. Наприклад, Ctrl+C – копіює відокремлений фрагмент тексту у буфер обміну; Ctrl+Shift - змінює мову (алфавіт) введення інформації і таке інше. <Print Screen> - дозволяє друкувати зміст екрану на принтері або скопіювати у буфер обміну, <Pause> - призупиняє виконання програми; <Esc> надає можливість відмінити дію; та <Back Space> надають можливість стирати введенні символи; <Insert> потрібна для перемикання режимів вставки і заміни під час введення тексту.

Питання для самоперевірки

1. Які перетворення проводять з аудіоінформацією при її записі та зчитуванні з носіїв?
2. Які пристрої знаходяться на аудіокартах ?
3. Яке призначення мікшера та еквайзера?
4. Яке призначення синтезатора?
5. Яке призначення монітора?
6. Які особливості розмітки екрана монітора в текстовому режимі?
7. Які особливості розмітки екрана монітора в графічному режимі?
8. Дайте поняття "частота кадрової розгортки" монітора.
9. Яке призначення клавіатури?
10. На які групи розбито усі клавіші клавіатури?
11. Яке призначення функціональних клавіш?
12. Яке призначення клавіш керування курсором?
13. Яке призначення спеціальних клавіш?

1.6 Призначення, конструкція і характеристики пристроїв персональних комп'ютерів: принтер, модеми, сканери, джерела безперебійного живлення

Принтер – це пристрій, призначений для виведення на папір (тверді носії) результатів роботи програми.

Існує велика кількість різноманітних моделей принтерів, які відрізняються принципами дії, інтерфейсом, продуктивністю, функціональними можливостями.

За способом формування зображення принтери поділяються на **контурні** (ударні) та **растрові**.

У контурних принтерах зображення символу має вигляд безперервної лінії та формується ударом по паперу (через фарбувальну стрічку) деталі з рельєфом відповідної форми (принцип друкуючої машинки).

У **растрових** принтерах зображення складається з безлічі дрібних (0,01 – 0,3 мм) точок. За принципом нанесення фарбуючих точок растрові принтери можна поділити на **матричні**,

струминні та **лазерні**. **Матричні** принтери застосовуються у даний час дуже мало. У **струминних** принтерів зображення формується мікрокраплями спеціального чорнила, що викидаються на папір через мініатюрні сопла. Швидкість друку від 10 до 60 с на сторінку.

Лазерні принтери забезпечують високу, близьку до друкарського способу, якість друку. В цих принтерах використовується принцип ксерографії: зображення формується на спеціальному барабані у вигляді сукупності електричних зарядів; до заряджених точок поверхні барабану прилипає тонко дисперсійний барвник, внаслідок чого зображення стає видимим; потім зображення відтисненням переноситься на папір і закріплюється на ньому сильним, але короткочасним прогріванням. Відмінність від звичайного ксерокопіювального апарату полягає в тому, що електричний рельєф друкуючого барабану формується за допомогою лазера, промінь якого моделюється за командою ПК. Швидкість друку лазерних принтерів від 5 до 15 с на сторінку при виведенні текстів. На виведення сторінок з великими малюнками потрібно декілька хвилин.

Модем - це пристрій для з'єднання ПК з телефонною лінією. ПК виробляє дискретні електричні сигнали (послідовність 0 та 1), а по телефонних лініях інформація передається в аналоговій формі (тобто у вигляді сигналу, який змінюється безперервно). Модеми виконують цифро – аналогове і обернене перетворення. При передачі даних модеми накладають цифрові сигнали комп'ютера на безперервну носійну частоту телефонної лінії (модулюють її), а при їх прийманні демодулюють інформацію і передають її у цифровій формі в комп'ютер. Модеми передають інформацію зі швидкістю 300 – 56000 біт за секунду, а по орендованих (виділених) каналах ця швидкість може бути більш високою.

Табл. 1.7 Характеристики деяких принтерів

Модель принтера	Фірма-виробник	Тип	Швидкість друку	Формат паперу	Кольоровість	Роздільна здатність, точок/дюйм	Маса, кг	Ємність пам'яті буфера друку
Laser Jet 4MP	Hewlett-Packard	Л	4 ст/хв	A4	М	600х600	9	6-22 Мбайт
Laser Jet 4P		Л	4 ст/хв	A4	М	600х600	9	2-26 Мбайт
DeskJet 550C		ТС	3-0,25 ст/хв	A4	К	300х300	6,5	128 Кбайт
DeskJet 1200/PS		ТС	7-0,25 ст/хв	A4	К	600х300	13,4	2-26 Мбайт
LX – 100	Epson	I 9	180 ст/с	A4	М	240х216	5,9	3 Кбайт
LQ – 100		I 24	200 ст/с	A4	М	360х360	4,5	11 Кбайт
Stylus – 1000		ПС	2,5 ст/хв	A3	М	360х360	8,6	128 Кбайт
Stylus Color 800		ПС	6-2,5 ст/хв	A4	К	1440х720	6,5	32 Кбайт
Mannesmann Tally 360	Mannesmann Tally	I 24	700 ст/с	A3	М	360х360	20	128 Кбайт
BJC – 800	Canon	ПС	0,7 ст/хв	A3, A4	К	360х360	10	7 Кбайт
LBP – 8IV		Л	8 ст/хв	A4	К	600х600	17	10 Мбайт

Табл. 1.8 Технічні дані деяких модемів

Модель модема	Фірма – виробник	Тип (модем –М, факс модем – ФМ)	Стандарт (протокол)	Швидкість приймання та передачі даних, біт/с
3260 Fast	Motorola	М	v.34	24000
3220 Plus	Codex	М	v.32bis	14400
Courier Dual Standard	U.S.Robotics	М	v.34	16800
Sportster		ФМ	v.32bis	14400
U – 1496	Zyxel	ФМ	v.34	16800
U – 1496 Plus (E, B, R)		ФМ	v.34	19200

Сканер – це пристрій, який дає змогу вводити у ПК чорно-біле або кольорове півтонове зображення, прочитувати графічну та текстову інформацію.

Існує багато моделей сканерів, які відрізняються методом оптичного «промацування» зображення, допустимим розміром оригіналу, якістю оптичної системи. За способом організації переміщення зчитуючого вузла відносно оригіналу сканери поділяються на планшетні, барабанні та ручні.

В планшетних сканерах оригінал укладають на скло, під яким рухається оптико-електронний зчитувальний пристрій.

У барабанних сканерах оригінал через вхідну щілину втягується барабаном у транспортний тракт і пропускається повз нерухомий зчитувальний пристрій. Барабанний сканер не дозволяє сканувати книжки, переплетені брошури.

Ручний сканер необхідно плавно переміщувати вручну по поверхні оригіналу. Сканер, як і фотокопіювальний пристрій, освітлює оригінал, а його світлочутливий датчик із певною частотою вимірює інтенсивність відбитого оригіналом світла. У процесі сканування пристрій перетворює інтенсивність світла на двійковий код, який передається у пам'ять комп'ютера. Якщо сканер при кожній вибірці реєструє один біт інформації, то він розрізняє або чорний, або білий колір. В залежності від кількості бітів, що відповідають одній вибірці, сканер може розрізняти більшу або меншу кількість відтінків від чорного до білого. Наприклад, восьми бітові сканери забезпечують реєстрацію 256 рівнів сірого.

Існує три види сканування для отримання чорно – білих зображень: сканування штрихового рисунка (реєструються тільки чорні та білі кольори), сканування півтонового зображення (проміжні відтінки сірого імітуються угрупованням точок чорного кольору з різною щільністю. Отримане зображення називається змазаним півтоновим), сканування зображення з використанням шкали яскравості (сірої шкали) – кожен елемент зображення описується певною групою бітів, які кодують конкретний відтінок сірого. Півтоновий растр накладається на зображення з великою кількістю градацій яскравості в момент виведення зображення на друк. Людське око не може розрізнити більш 256 відтінків сірого, тому при забезпеченні такого рівня зображення виглядає цілком природно.

ARTEC VIEWSTATION AS6E

Табл. 1.9 Планшетний TWAIN-сумісний сканер

Максимальний розмір оригіналу	216x297 мм
Оптична роздільна здатність	300 точок/дюйм
Інтерполяційна роздільна здатність	4800x4800 точок/дюйм
Глибина кольору	16,7 млн (24 біт)
Кількість рівнів сірого	256 (8біт)
Інтерфейс	LPT

ARTEC VIEWSTATION AT12 SCSI**Табл. 1.10 Планшетний TWAIN-сумісний сканер**

Максимальний розмір оригіналу	216x297 мм
Оптична роздільна здатність	600x1200 точок/дюйм
Інтерполяційна роздільна здатність	4800x4800 точок/дюйм
Глибина кольору	16,7 млн (24 біт)
Кількість рівнів сірого	256 (8біт)
Інтерфейс	SCSI

ARTEC VIEWSCAN A2000D**Табл. 1.11 Ручний TWAIN-сумісний сканер**

Інтерполяційна роздільна здатність	До 800 точок/дюйм
Глибина кольору	16,7 млн (24 біт)
Кількість рівнів сірого	256 (8біт)
Інтерфейс	SCSI

BOEDER COMPACT COLOUR 1200 (65137)**Табл. 1.12 Планшетний TWAIN-сумісний сканер**

Максимальний розмір оригіналу	216x297 мм
Оптична роздільна здатність	300x600 точок/дюйм
Інтерполяційна роздільна здатність	1200x1200 точок/дюйм
Глибина кольору	16,7 млн (24 біт)
Кількість рівнів сірого	256 (8біт)
Інтерфейс	LPT

Кольорове сканування – це сканування у «сірому» режимі із різними фільтрами (червоним, синім, зеленим). Зображення, яке утворюється внаслідок сканування 256 відтінків по кожному компоненту, дають в сумі 16,7 млн можливих комбінацій кольорів (24 - бітне зображення). У 24 – бітному зображенні не відбувається точне представлення всіх відтінків, оскільки при подальшому коректуванні гами кольорів, яскравості або контрастності розмір палітри значно зменшується. Це зумовило появу моделей сканерів із глибиною кольору 30 і навіть 36 біт. При неминучій втраті даних запасні біти дають можливість отримати природне 24 – бітне зображення. Звичайна розрізняльна здатність сканера складає 300 -720 точок на дюйм. Більша їх кількість (більше 1000) відображує інтерполяційну розрізняльну здатність, яка досягається апаратно – програмними засобами із використанням математичної обробки параметрів розташованих поряд точок зображення. Якість відсканованого матеріалу залежить від оптичної розрізняльної здатності сканера, але не тільки. Результат роботи сканера (система «сканер – дешифратор») залежить від якості оригіналу (чіткість символів, стабільність їх розмірів у рядку, насиченості кольору).

Джерела безперебійного живлення (UPS) – призначені захищати обчислювальні пристрої, зокрема ПК, від перебоїв електропостачання. До складу UPS входить акумулятор, який зберігає заряд, якого достатньо для живлення ПК, на протязі деякого часу, в момент вимкнення електромережі.

Джерела безперебійного живлення випускають на різні потужності (від 250 Вт), з різною тривалістю автономного живлення (від декількох хвилин до декількох годин) і з різною мірою автоматизації.

Вибір конфігурації ПК

Вивести аналітичну залежність між процесами обробки даних та технічними характеристиками ПК ще нікому не вдалося. В реальному житті передусім необхідно прагнути до якомога більшої відповідності між споживчими потребами та характеристиками ПК.

Обираючи тип процесора (може бути і тип ПК) потрібно мати на увазі, що з часом буде складно перейти до більш потужної конфігурації: заміна процесора потребує заміни дорогого елемента – системної плати.

Потрібно подумати і про економічну доцільність модернізації, оскільки сукупна ціна вибулих, застарілих комплектуючих, як правило, невисока. Деякі фірми займаються апгрейдом (upgrade - модернізація) комп'ютерів і можуть задовольнити будь-яке замовлення користувача, включаючи старі комплектуючі і вартість послуг. Але послуги таких фірм достатньо дорогі. Інші види комплектуючих потребують розкриття системного блоку, розбирання деяких його пристроїв, використання спеціального інструмента, знання апаратної частини ПК.

При виборі варіанта купівлі або модернізації ПК потрібно враховувати, що ціни на ПК швидко падають. Тому купівля не вигідного на певний момент часу комплекту з підвищеними характеристиками може стати вигідною і обґрунтованою пізніше.

Питання для самоперевірки

1. Яке призначення принтера?
2. Яким чином формується зображення на екрані під час друку на струминному принтері?
3. Яким чином формується зображення на екрані під час друку на лазерному принтері?
4. Яке призначення модемів?
5. Які основні принципи роботи модемів?
6. Яке призначення сканерів?
7. Які основні принципи роботи планшетних сканерів?
8. Які основні принципи роботи барабанних сканерів?
9. Які відмінності сканування в режимах штрихового рисунка, полу тонового зображення, з використанням шкали яскравості?
10. Які принципи кольорового сканування?
11. Яке призначення джерел безперебійного живлення?

2. Операційна система windows xp. основні поняття на принципи роботи у системі.

2.1 Історія Windows

Перша система Windows з'явилася на світ у 1986 році та являла собою сукупність програм, що розширювали можливості існуючих на той час операційних систем задля більшої зручності у роботі. Через декілька років вийшла друга версія, але особливої популярності вона не отримала. Однак у 1990 році вийшла нова версія – Windows 3.0, яка почала використовуватися на багатьох персональних комп'ютерах. Популярність нової версії Windows обумовлювало декілька причин. Графічний інтерфейс дозволяв працювати з об'єктами вашого комп'ютера не за допомогою команд, а за допомогою наочних та зрозумілих дій зі значками, що позначали ці об'єкти. Можливість одночасної роботи з декількома програмами суттєво підвищила зручність та ефективність роботи. Крім того, зручність та легкість написання програм для Windows сприяли появі значної кількості програм, що працювали під керуванням операційної системи Windows 3.0. Також набагато краще було організовано роботу з різноманітним комп'ютерним обладнанням, що також визначило популярність системи. Зміни у наступних версіях Windows були спрямовані на підвищення надійності системи, а також на підтримку засобів мультимедіа (версія 3.1) та роботу у комп'ютерних мережах (версія 3.11).

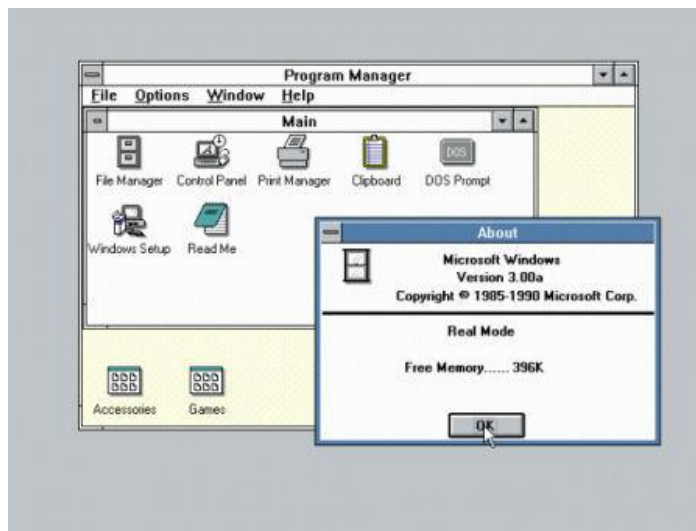


Рис. 2.1. Інтерфейс операційної системи Windows 3.11

Паралельно з розробкою Windows компанія Microsoft у 1988 році почала роботу над новою операційною системою, якій було дано назву Windows NT. Однією з головних вимог до нової системи було істотне підвищення надійності та ефективної підтримки роботи у мережі. При тому інтерфейс мав не відрізнятися від інтерфейсу Windows 3.0. Цікаво, що найбільш розповсюдженою версією Windows NT виявилась третя версія. У 1992 році з'явилася версія Windows NT 3.0, а у 1994 році – Windows NT 3.5.

Процес розвитку операційних систем продовжується, та у 1995 році з'явилася система Windows 95, яка являла собою новий етап у історії Windows. Порівняно із Windows 3.1 істотно змінився інтерфейс та зросла швидкість роботи програм у системі. Однією з нових можливостей Windows 95 була можливість автоматичної настройки додаткового обладнання комп'ютера за відсутності конфліктів з обладнанням, що було встановлено раніше. Наступною суттєвою особливістю системи була можливість роботи в мережі Інтернет без використання додаткових програмних засобів.

Інтерфейс Windows 95 являв собою основу для усіх операційних систем родини Windows, та у 1996 році з'являється допрацьована версія Windows NT 4.0, як мала саме такий інтерфейс, як й Windows 95. Продовженням розвитку Windows 95 була операційна система, що з'явилася у 1998 році. Усе, окрім збереженого інтерфейсу, було суттєво змінено, найбільші зміни торкалися

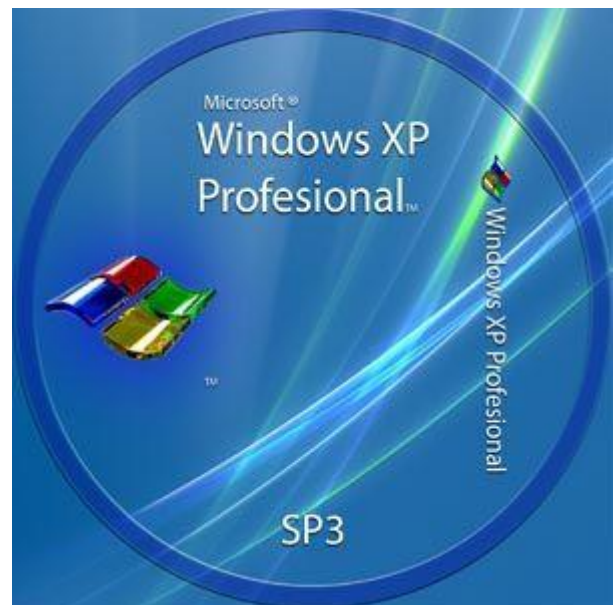
внутрішньої структури системи. Багато уваги було приділено роботі з Інтернет, а також підтримці сучасних протоколів передачі інформації – стандартів, що забезпечують обмін інформацією між різним обладнанням. Окрім того, відмінністю Windows 98 від її попередників була можливість роботи з використанням декількох моніторів одночасно.



Наступним етапом у розвитку Windows була поява Windows 2000 та Windows Me (Millennium Edition – редакція тисячоліття). Система Windows 2000 була розроблена на основі Windows NT та наслідувала від неї високу надійність та захист інформації від стороннього втручання. Операційна система Windows Me являла собою наступницю Windows 98, але отримала багато нових можливостей.

Перш за все, це покращена робота з засобами мультимедіа, можливість записувати не тільки аудіо, а й відеоінформацію, потужні засоби відновлення інформації після збоїв на багато іншого. Поступово різниця між різними системами Windows зникає, та наступна система Windows XP призначена для заміни як Windows 2000, так й Windows Me. Версія Windows XP є найбільш розповсюдженою та об'єднує в собі переваги операційних систем попереднього покоління, які знані та шановані користувачами, а також надійність та багатofункціональність Windows 2000.

Операційну систему Microsoft Windows XP створена на основі технології NT є безпосередньою наступницею системи Windows 2000. Разом з тим усі найкращі нововведення, що були додані у Windows Me, можна знайти й у Windows XP. Зі збереженням високих показників надійності, безпеки та швидкодії, система стала більш простою для освоєння, у ній з'явилося безліч засобів, призначених для індивідуальних користувачів, що частіше за все працюють вдома.



В цій новій версії особлива увага приділяється роботі з малюнками, аудіо та відео. Версія Microsoft Windows XP Professional призначена, як це зрозуміло з назви, для професіоналів. Ця версія найбільш часто використовується в установах. Але якщо ви вдома виконуєте складні роботи зі створення та редагування зображень, моделювання та конструювання, чи будь-які інші складні роботи, то версія також задовольнить вас на домашньому комп'ютері. Версія Microsoft Windows XP Server призначена для встановлення на сервері – потужному комп'ютері, що забезпечує роботу багатьох користувачів в комп'ютерній мережі.

У версії Windows XP суттєво змінився зовнішній вигляд системи. Разом з тим реалізована можливість використовувати також й старий інтерфейс, якщо ви до нього звикли. Особливо варто відмітити роботу програм у режимі сумісності з попередніми версіями Windows. Ви маєте можливість працювати з програмою, яку було написано для Windows 95 та яка не працює в Windows 2000. Усі версії Windows XP мають безліч нововведень. Підтримується значно

більше різноманітних пристроїв, система дозволяє легко та зручно обробляти відеофільми, фотографії, зображення та музику. Тепер за допомогою Windows XP будь-хто може налаштувати домашню мережу у складі двох-трьох комп'ютерів, забезпечуючи сумісне використання файлів, папок, принтера, факсу та доступу в Інтернет.

При роботі з Windows XP для запису інформації на компакт-диски чи зовнішні носії вам не потрібно встановлювати додаткові програми. Записувати можна безпосередньо з провідника Windows. До речі, провідник досить суттєво змінився. Окрім підтримки стиснутих папок, особливих папок для зберігання зображень, музики та відео, було додано панель з командами, склад яких змінюється в залежності від команд, які вами виконуються.

Цікавим компонентом системи є редактор відеофільмів. Тепер ви маєте можливість виконати професійний нелінійний монтаж власних записів. Значно легше тепер можна працювати зі сканерами та цифровими фотоапаратами. Вам не потрібна ніяка програма, щоб увести фотографію в комп'ютер, трохи перетворити її та роздрукувати на принтері. Універсальний програвач аудіо та відео тепер підтримує більше форматів та дозволяє змінити свій зовнішній вигляд. Також ви маєте можливість самостійно створювати звукові файли у популярному форматі mp3. Цей універсальний програвач відтворить для вас зміст DVD-дисків, що дозволить насолодитися найвищою якістю зображення та звуку під час перегляду сучасних фільмів. Для розваг в Windows XP додано декілька нових ігор, частина з яких дозволяє вам грати через Інтернет.

Суттєво покращено й систему захисту. Відтепер, при випадковому видаленні важливих системних файлів, вони будуть автоматично відновлені. В наяві є можливість також відновити систему до попереднього стану, після встановлення нових програм та обладнання. Покращена підтримка технології Plug & Play дозволяє приєднати до комп'ютера багато з сучасних побутових пристроїв.

Подальший розвиток отримали й засоби роботи з Інтернет. Істотно перероблено додаткову систему, покращено систему безпеки. Багаточисельні зміни торкнулися засобів адміністрування та управління роботою користувачів у локальній комп'ютерній мережі.

В системі в наявності також безліч інших новацій, але спочатку рекомендуємо вам ознайомитися з початковими поняттями, що використовуються у Windows XP.

2.2 Файлова система та структура.

Інформація у комп'ютері зберігається в пам'яті чи на різноманітних носіях, таких як жорсткі диски, компакт-диски, зовнішні HDD USB жорсткі диски, накопичувачі, фотобанки, цифрові фотоальбоми та інші. Коли живлення комп'ютери вимикається, то уся та інформація, що зберігалася у пам'яті, зникає, а та, що зберігалася на носіях - ні. Для впевненої роботи за комп'ютером слід знати основні принципи збереження інформації на внутрішніх та зовнішніх носіях.

Уся інформація, що призначена задля довгострокового використання, зберігається в файлах. Файл являє собою послідовність байтів, які об'єднуються за деякою ознакою та ця сукупність має власне ім'я. Система збереження файлів та роботи з ними має назву файлова система. Для зручності файли зберігаються у різних папках, що розташовані на дисках. В комп'ютері може бути декілька дисків. Будь-який жорсткий диск, компакт-диск, цифровий відеодиск чи мережевий диск частіш за все називають саме диском, бо принципи збереження файлів на них ідентичні. Кожному диску ставиться у відповідність одна з літер латиниці від A до Z, але існують деякі стандарти позначення. Наприклад, літерами A та B позначаються гнучкі диски (у теперішній час ці типи носіїв майже не використовуються, але за традицією перші дві літери латиниці резервуються саме під ці носії), літерою C позначається головний диск вашого комп'ютера, що міститиме систему Windows. Літера D та наступні використовуються для логічних дисків та інших. Після літери, що позначає диск, ставлять символ „:”, щоб показати, що ця літера позначає саме диск (наприклад, C: чи K:). Окрім літери, кожен диск має власне автентичне ім'я, яке теж називають міткою. Частіш за все при посиланні на диск використовують мітку на літеру-позначення диска. Наприклад, підпис **Main (C:)** означає, що головний диск C вашого комп'ютера має мітку **Main**.

На кожному диску можна зберігати багато різних файлів, кількість залежить від розміру диска та файлів. Кожен файл може бути розташований чи безпосередньо на диску, чи у деякій папці, яка, в свою чергу, теж може бути розташована на диску чи у іншій папці. .

Той факт, що файли можуть бути розташовані у різних папках, дозволяє нам створити декілька файлів з однаковими назвами, але у різних папках. Структура зберігання інформації на диску, при якій папки можуть бути розташовані в інших, має назву *ієрархічної* чи *деревоподібної*. Насправді, така структура схожа на дерево, на котрому кожне листя являє собою файл, а гілка - папку. Листя може зростати як з гілки, так й безпосередньо зі стовбура. Також може бути, що від гілки відходить ще одна гілка, а вже на ній росте листя. Для того, щоб однозначно визначити конкретний файл, потрібно задати його назву та місце розташування, тобто назву диска та вкладених папок, у котрих цей файл міститься. Така структура визначення положення файла має назву *повне ім'я файла* чи *шлях до файла*.

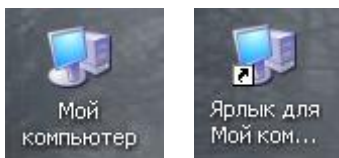
Якщо ми вказуємо шлях до файла, то імена папок відокремлюються одне від одного та від імені диска за допомогою символу оберненої нахиленої риски "\", наприклад, **C:\Мої документи\Мої рисунки\Осінь.bmp**. Такий запис означає, що файл з ім'ям **Осінь.bmp** розташований у папці **Мої рисунки**. Ця папка міститься у папці **Мої документи**, що розташована на диску **C**:

Зверніть увагу, що в розглянутому прикладі ім'я файла містить у собі символ крапки та неначе складається з двох частин: до крапки та після неї. Та частина імені, яку розташовано після крапки, має назву розширення та використовується задля визначення типу інформації, що зберігається в файлі. Наприклад, розширення .txt позначає текстовий документ, .wav – файл, що містить звуки, а розширення .bmp – зображення. В Windows XP багато з розширень файлів не відтворюються на екрані, тому, скоріш за все, на моніторі назва цього файла буде мати вигляд лише **Осінь**, але Windows буде визначати файл як файл зображення.

Важним поняттям в Windows XP являється поняття ярлика. На любой объект Windows можно сослаться из другого места - такая ссылка и называется ярлыком. Например, в какой-то папке расположена часто используемая программа. Для быстрого доступа к этой программе из разных мест можно поместить в эти места ярлыки, содержащие адрес реального местонахождения файла, запускающего выполнение программы.

Важливым понятием в Windows XP є поняття ярлика. На будь-який об'єкт Windows можна зробити посилання з іншого місця – таке посилання має назву *ярлик*. Наприклад, в деякій папці розташовано програму, яка доволі часто використовується. Для швидкого доступу до неї з різних місць можна розташувати в цих місцях ярлики, що містять адрес реального місцезнаходження файла, який запускає виконання програми.

Не потрібно копіювати програми та дані в різні папки, достатньо лише розмістити ярлики, що посилаються на потрібний файл, у деяких місцях. Усі такі ярлики будуть вказувати на оригінальний файл. Видалення на пересування ярлика не впливає на розташування оригінального файла, тому використання ярликів може забезпечити додатковий захист.



Зовні сам ярлик відрізняється від файла, на який він посилається, наявністю стрілки у лівому нижньому куті іконки. Саму ж іконку іноді називають *піктограма*.

Рис. 2.2. Піктограма та ярлик

Запитання для контролю.

1. Які дії можна виконати за допомогою миші та які функції вони виконують?
2. На які групи поділяються клавіші на клавіатурі?
3. Як за допомогою клавіатури визвати на екран системну довідку, закрити вікно, відкрити меню Пуск, переключити російську чи англійську розкладки клавіатури?

4. Як змінити розміри вікна?
5. Як максимізувати вікно?
6. Як згорнути вікно?
7. Як зробити вікно активним?
8. Як перейти від однієї програми, що виконується, до іншої?
9. Як завершити роботу програми?
10. Що таке ярлик та чим він відрізняється від файла?

2.3 Основні принципи роботи з системою

Для спілкування з користувачем в Windows XP використовується графічний інтерфейс, який відображує інформацію на екрані у вигляді різноманітних малюнків. Багато з функцій системи пов'язано з невеликими малюнками, що мають назву піктограм.

В Windows XP широке розповсюдження отримало поняття об'єкта. Об'єкти – це файли та папки, а також деякі фізичні пристрої, наприклад, диски та принтери. На екрані монітора усі об'єкти мають вигляд піктограм.

Для того, щоб керувати системою, немає потреби вводити з клавіатури складні та довгі команди. Керування здійснюється за допомогою різних операцій над піктограмами, для чого використовується комп'ютерна миша чи клавіатура. Коли ви відокремлюєте, переміщуєте чи виконуєте інші операції з піктограмами, ви насправді видаєте команди операційній системі, котра виконує деякі операції з об'єктами, що пов'язані з цими піктограмами.

Однак не всі операції, які потрібно, можуть бути зроблено за допомогою піктограм. Іноді легше лише вказати на необхідну дію, обрав її зі списку. При цьому нам також не буде потрібно вводити команди. Система запропонує декілька варіантів роботи, а ви вкажете саме на той варіант, який потребуєте. Списки дій, що можливо виконати, мають назву *меню*, а робота з меню навіть більш поширена ніж робота з піктограмами.

Слово windows перекладається з англійської як вікна. Вікно являє собою відокремлену область на екрані, що пов'язана з деякою програмою. Після запуску операційної системи Windows XP увесь екран монітора займає особливий малюнок, що має назву **Робочий стіл** чи, по-англійськи, **Desktop**. Усі вікна розташовуються на робочому столі та їх можна пересувати, збільшувати в розмірах чи видаляти (закривати). Вікна можуть частково чи повністю перекривати одне одного. Робочий стіл дозволяє запускати програми, налаштовувати систему та виконувати інші дії. Розміри та кількість малюнків та надписів на робочому столі можуть значно відрізнятися в залежності від особливостей налаштування даної копії операційної системи на вашому комп'ютері.

Однак у будь-якому випадку в нижній частині робочого стола розташовано полосу, що має назву **Панель задач (Taskbar)**. Головне призначення Панелі задач – відображення програм, що запущено, у вигляді кнопок та піктограм, а також більш швидке перемикання між ними. Окрім того за допомогою панелі задач можна запускати деякі додаткові програми.



Рис. 2.3. Панель задач з кнопками програм

У лівій частині панелі задач знаходиться рамка, у внутрішній частині якої написано слово **Пуск(Start)**. В системі Windows такі рамки з текстом та малюнками називаються кнопками, тому надалі будемо казати, що на панелі задач розташована кнопка **Пуск (Start)**. Призначення цієї кнопки – запуск програм та налаштування комп'ютера. Можна навіть сказати, що будь-яка робота в Windows починається з натискання цієї кнопки. Меню „Пуск” – другий за важливістю (після Панелі керування) засіб керування будь-якою версією Windows. Якщо Панель керування займається внутрішніми ресурсами Windows, то кнопка „Пуск” несе відповідальність за усі встановлені в системі програми. Розглянемо зміст меню Пуск.



Рис. 2.3. Робочий стіл Windows XP

Кнопка Інтернет запускає оглядач Internet Explorer – програму, що дозволяє переглядати зміст веб-сторінок. Поряд з ІЕ існує ще велика кількість програм - оглядачів (Opera, Firefox Mozilla, Safari), які ніяк не поступаються за якістю роботи ІЕ, однак Internet Explorer є програмним продуктом компанії Microsoft, тому він присутній в меню. Однак після призначення іншої програми-оглядача „броузером за умовчанням” зміниться також й цей пункт меню.

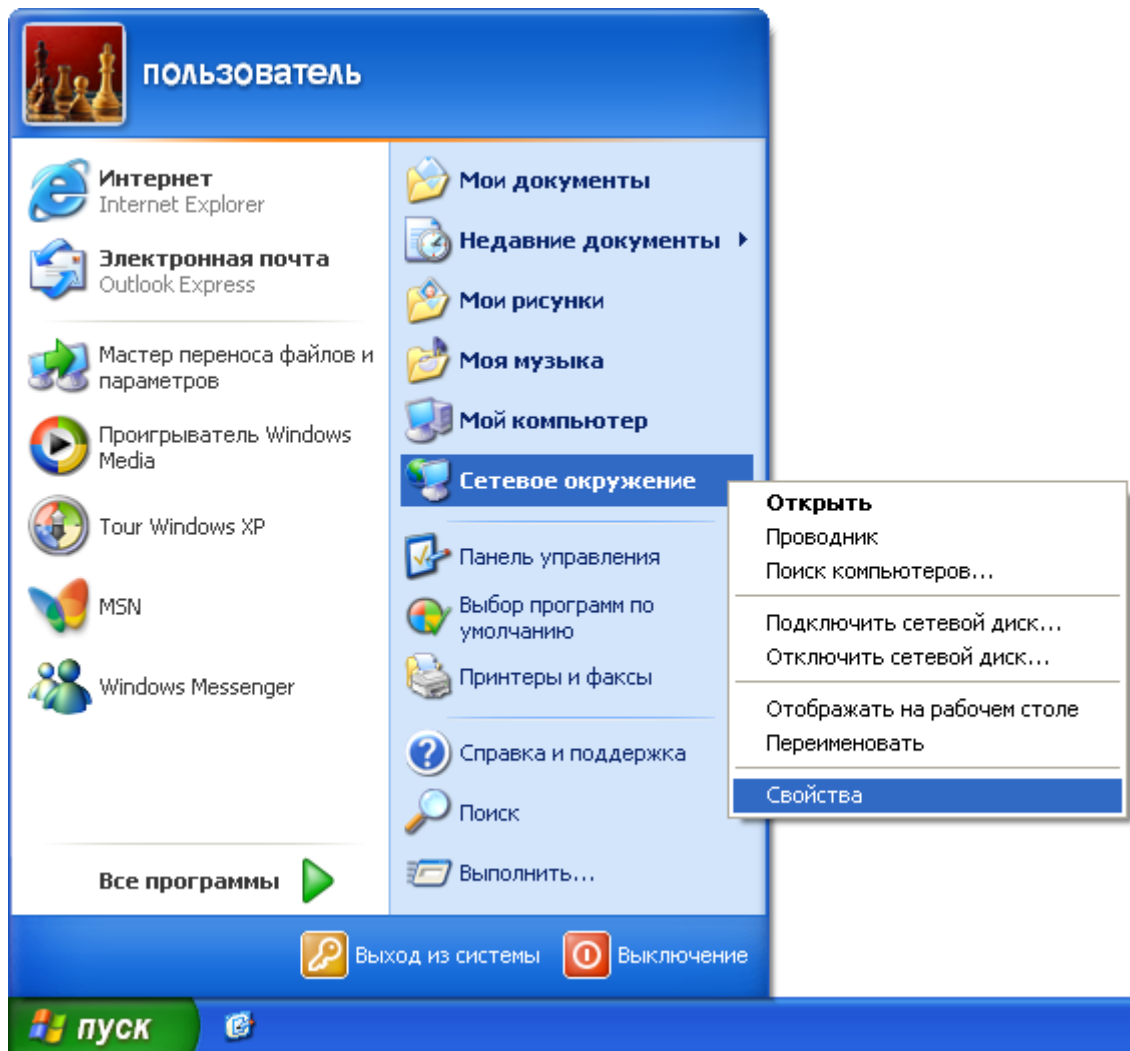


Рис. 2.5. Зовнішній вигляд меню Пуск. Для елемента Мережеве Оточення надано його контекстне меню.

Електронна пошта призначена для перегляду листів, отриманих за допомогою інтернет-пошти (так званий e-mail)

Далі йде перелік програм, що використовувалися частіше за інші. Створення такого переліку мало на меті спростити доступ до програм та служб, які частіше за всіх запускалися на даному комп'ютері. Однак насправді цей перелік вельми далекий від того, що бажає бачити в ньому користувач, тому легше лише виносити ярлики до програм, які часто використовуються, на робочий стіл, та таким чином полегшити собі користувацьке життя.

Розглянемо тепер меню *Всі програми*. До нього ви будете звертатися частіш за інші меню, бо звідси можна здійснювати доступ до усіх встановлених в Windows програм.

Основна папка – **Програми (Programs)** – містить в собі безліч програмних груп, тобто підпапок з ярликами, кожна з яких відповідає визначеній програмі. На початку переліку знаходиться декілька додаткових можливостей для користувача Windows XP. Найбільш важливою серед них є Windows Update – безкоштовне оновлення системи та деяких програм безпосередньо від виробника. Однак при використанні цієї можливості треба бути обережним – у деяких випадках некоректне встановлення системи та подальше оновлення може спричинити блокування роботи системи чи навіть перехід системи до неробочого стану.

Усі базові програми Windows, - тобто ті програми, що встановлюються разом із самою операційною системою, - ви знайдете у папці „*Стандартні (Accessories)*”. Тут ви побачите простий текстовий редактор Word Pad, графічний редактор Paint, віртуальний калькулятор, режим роботи з командним рядком та багато чого іншого.

В групі *Службові (System Tools)* знаходяться програми для обслуговування жорсткого диска комп'ютера – Перевірка диску (Scandisk), Очищення диску (Disk CleanUp), дефрагментація диска (Disk Defragmenter), Майстер обслуговування (Optimization Wizard), меню Інформація про систему (System Information).

Група „Розваги” (*Entertainment*) об'єднує програми для роботи зі звуком та відео. В Windows XP їх декілька – Універсальний програвач та Програвач компакт-дисків.

Група „Мережа” (**Communications**) містить в собі програми для з'єднання комп'ютера з іншими ПК за допомогою телефонної лінії, домашньої локальної мережі чи мережі Інтернет.

Група „Ігри” (**Games**) не потребує додаткових коментарів. Для відпочинку вам пропонується розкласти декілька пасьянсів, перевірити, чи насправді „Сапер” помиляється лише один раз, чи поганяти кулі у „Пінбол”. Також є можливість перевірити ваш рівень інтелекту та виграти в інтернет-шашки чи нарди в суперника з мережі Інтернет.

У папці **Автозавантаження (Startup)** знаходяться ярлики до тих програм, що завантажуються безпосередньо після старту Windows XP. Наприклад, деякі користувачі додають до цієї папки ярлик від текстового редактора – у цьому випадку він буде розгортати власне вікно одразу ж після включення комп'ютера та завантаження операційної системи.

Також крім вище перелічених груп у меню Пуск знаходяться папки з ярликами до всіх програм, що встановлені на вашому комп'ютері. Тому якщо на робочому столі ви не бачите ярлика для потрібної вам програми – відкривайте меню Програми – він заховався від вас саме там.

Розглянемо ще декілька груп, що знаходяться у меню Програми.

Мої документи, Мої рисунки, Моя музика дозволяють отримати швидкий доступ до змісту однойменних папок в комп'ютері (інакше ви маєте пройти весь шлях C:\Documents and Settings\User\...\Мої документи\Моя музика безпосередньо з клавіатури).

Нещодавні документи містять ярлики на зображення чи документи, які було нещодавно відкрито на цьому комп'ютері.

Мій комп'ютер містить в собі ярлики на усі локальні диски, зовнішні та приєднані до цього комп'ютера прилади. Дійсно, якщо в вас відразу ж 5 дисководів та 21 локальний диск - навіщо переносити на робочий стіл усі ці ярлики? Лишень один клік по піктограмі Мій комп'ютер дозволить вам добратися до будь-якого з них.

Мережеве оточення відкриває доступ до програм та документів, що знаходяться на всіх комп'ютерах, під'єднаних до локальної мережі. Звичайно, якщо вам це дозволено.

Панель керування – найважливіший елемент меню Пуск. Основні налаштування Windows згруповані в спеціальній папці, яка має назву *Панель керування (Control Panel)*. Для того, щоб відкрити цю папку, слід обрати однойменну команду в головному меню Windows. Панель керування може відображатися в одному з двох різних режимів. Перший режим, що використовується за умовчанням, групує значки різних налаштувань. При цьому зовнішній вигляд панелі керування схожий на Web-сторінку. Під час переходу за посиланнями, ви обираєте групу налаштувань та знаходите потрібну програму

Другий режим, який має назву класичного, відображує усі піктограми разом, не відокремлюючи окремі групи. У цьому режимі панель керування не відрізняється від звичайної папки. Режим відображення панелі у вигляді Web-сторінки є більш зручним, але деякі користувачі вважають за краще користуватися саме класичним режимом. Для того щоб перейти з одного режиму до іншого потрібно використати посилання у лівій частині вікна панелі керування, на панелі задач.

мишу, клавіатуру, принтери, модеми, сканери та інше. Для налаштування обраного обладнання чи групи пристроїв слід обрати відповідне посилання чи двічі натиснути потрібну піктограму.

Дата, час, мова та регіональні налаштування (Date, Time, Language and regional settings). У цій групі ви можете налаштувати деякі параметри, що пов'язані з особливостями мови та країни, де ви мешкаєте. Наприклад, ви маєте можливість встановити формат чисел та дат, обрати потрібну грошову одиницю. Також ви встановлюєте поточну дату та час. Окрім того, у вас є можливість використовувати декілька мов під час роботи, наприклад, українську, англійську, російську та башкирську. Саме тут ви обираєте додаткові мови, а також встановлюєте засіб перемикання між мовами.

Облікові записи користувачів (User Accounts). Windows XP дозволяє працювати на одному комп'ютері декільком користувачам, кожен з яких буде мати власні налаштування та зовнішній вигляд робочого столу. Можна швидко перемикатися між різними користувачами. В даній групі ви можете додати нового користувача, а також змінити налаштування тих користувачів, що вже існують.

Спеціальні можливості (Accessibility Options). За їх допомогою Windows налаштовується для людей із порушеннями зору, слуху чи рухливості рук. Ви можете обрати великий розмір літер у надписах, високу контрастність зображень, звукові ефекти, а також налаштувати інтерактивну клавіатуру.

Робота з налаштування системи Windows XP за допомогою панелі керування є досить зручною. Усі схожі налаштування зосереджено в одному місці, там же є посилання на пов'язані за сенсом налаштування з різних груп. Вибір потрібних налаштувань зводиться до обрання групи та наступного обрання потрібної задачі.

Запитання для контролю.

1. Що таке робочий стіл?
2. Як змінити налаштування робочого столу?
3. Яка структура стандартного вікна Windows XP?
4. Що таке Панель задач та які її функції?
5. Яким чином можна сховати Панель задач з робочого столу?
6. Як викликати системну довідку Windows XP?
7. Як завершити сеанс роботи у Windows XP?
8. Як завантажити програму?
9. Як завантажити документ?
10. Як викликати системне меню?
11. Як викликати контекстне меню*
12. Які групи входять до меню Пуск?
13. Як за допомогою меню Пуск можна проглянути документи, що недавно відкривалися?
14. Група Програми. Її призначення та робота з програмами, що недавно було використано.
15. Для чого потрібна Панель керування? Чим відрізняються категорії зовнішнього вигляду даної панелі?
16. Опишіть вміст панелі керування та надайте стислий опис її компонент.
17. Як за допомогою Панелі керування додати нову мову?
18. Як завантажити Блокнот, Калькулятор?
19. Як переналаштувати системну дату та час?
20. Як визначити системну дату та час?
21. Як визначити поточну розкладку клавіатури?
22. Дайте назву головних елементів панелі інструментів.

2.4 Робота з файлами та папками.

Файлом будемо називати визначений обсяг інформації одного типу, що зберігається на деякому фізичному носії та має власне ім'я та розширення. Розширення файла призначено для однозначної та вичерпної ідентифікації типу файлового об'єкта, воно записується правобіч від імені файла та відокремлюється від нього крапкою.

Папкою в операційній системі Windows XP називають спеціальний файловий об'єкт, що грає роль контейнера для зберігання інших папок і файлів, та відображується на екрані за допомогою спеціального значка, що має вигляд канцелярської папки.

Взагалі усі значки, що знаходяться всередині ікон Windows, мають назву *піктограма*.

Атрибути файлових об'єктів

Кожен файл або папка в Microsoft Windows XP може мати власний набір атрибутів, що вказують на деякий з можливих методів використання даного файлового об'єкта. У файловій системі FAT/NTFS розрізняють три основні атрибути файлів і папок:

Тільки читання – режим захисту від випадкових змін, редагування чи знищення інформації, яка у файлі зберігається. Якщо для файла чи папки увімкнено режим „Тільки читання”, то запис до нього буде унеможливлено.

Архівний – атрибут файла чи папки, який означає, що даний файловий об'єкт підлягає компресії та його зміст може бути стиснутий за допомогою спеціальних програм. Треба прийняти до уваги, що у випадку зміни даного параметра для групи об'єктів атрибут може бути поза доступом у випадку, коли деяка частина файлів з відокремленого масиву має бути заархівована, а лишок - ні.

Прихований – параметр, що вказує на те, що даний файловий об'єкт є прихованим та за звичайних умов в операційній системі Windows XP відображатися не буде. Приховані файли та папки неможливо використовувати, змінювати чи відкривати, якщо їх ім'я невідоме.

Атрибути, що призначені для кожного з файлових об'єктів, позначаються у Windows XP за допомогою спеціальних символів:

R – Тільки читання, H - Прихований, A - Архівний.

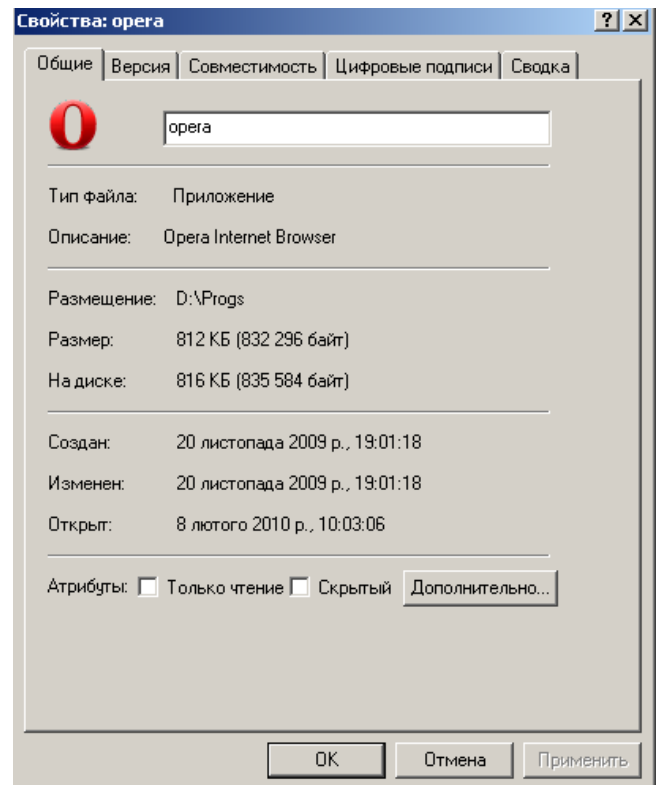


Рис. 2.7. Вікно властивостей файла (контекстне меню ► пункт Властивості)

Змінити атрибути файла чи папки можна наступним чином: клацніть по значку файлового об'єкта правою клавішею миші та у контекстному меню, що з'явилося, оберіть пункт Властивості. У нижній частині вікна властивостей файла чи папки ви побачите перелік призначених для даного файлового об'єкта атрибутів. Призначення чи зняття атрибутів файлів чи папок здійснюється методом встановлення чи скидання відповідних позначок.

Створення папок та файлів

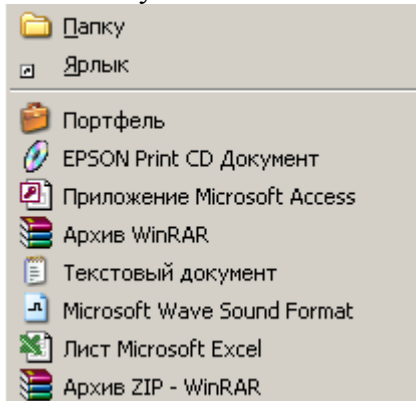
Для того щоб створити новий файл чи папку в будь-якій папці на диску вашого комп'ютера потрібно здійснити наступну послідовність дій:

1. Перейдіть до папки, в якій ви бажаєте створити нову папку чи файл. Це може бути й в кореновому розділі вашого диска;

2. Клацніть правою клавішею миші на будь-якому вільному від занчка місцю папки, яку вами було відкрито в головному робочому вікні. Провідника;

3. У вікні, що з'явилося, оберіть пункт Створити та вкажіть в ньому тип файлового об'єкту, який ви бажаєте створити.

За умовчанням Windows пропонує створити наступні типи файлових об'єктів:



Папка - нова папка
 Ярлик - новий ярлик
 Портфель - портфель
 Рисунок - графічний файл
 Текстовий документ - текстовий документ
 Звук WAV - звуковий файл
 Зжата ZIP-папка - папка, зжата за допомогою вбудованого у Windows XP архіватора WinZip.

Рис. 2.8. Меню, яке з'являється при створенні файла безпосередньо за допомогою контекстного меню (контекстне меню вікна ► створити)

Якщо в вашій системі встановлено додаткові програми та програмні комплекси, то в переліку можуть з'явитися також ці програми (в даному випадку – програма для принтера Epson Print CD)

Перейменування файла чи папки в Windows XP можна виконати трьома різними способами:

Перший спосіб

- Відокремте мишкою файл чи папку, яку ви бажаєте перейменувати;
- Клацніть на цьому об'єкті мишою ще раз після нетривалої паузи, таким чином, щоб виділенням виявився лише підпис значка - зовнішній вигляд об'єкта перед вводом його нової назви вказано на полях.

Другий спосіб

- Виділіть кліком миші файл чи папку, яку ви бажаєте перейменувати;
- Оберіть в лівому командному меню Провідника пункт Задачі для файлів чи папок, та далі Перейменувати файл натисканням на кнопку;
- Введіть нову назву об'єкта.

Спосіб третій

- Клацніть на об'єкті, який ви бажаєте перейменувати, лівою клавішею миші;
- У вікні, яке з'явиться, оберіть пункт Перейменувати;
- Уведіть нову назву об'єкта.

Нову назву об'єкта може бути задано як на латиниці, так й на кирилиці, та вона має містити від 0 до 255 символів.

Переміщення та копіювання папок та файлів

Переміщення того чи іншого файлового об'єкта має на увазі зміну його місцезнаходження на диску вашого комп'ютера, під час якої оригінальний файл копіюється у вказану користувачем папку чи на обраний їм носій, а сам оригінал знищується.

Для того, щоб перенести кудись файл чи папку, потрібно здійснити наступну послідовність дій:

- Виділіть кліком миші файл чи папку, яку ви бажаєте перенести;

- Оберіть пункт Перемістити файл чи папку в командному меню Задачі для файлів та папок програми Провідник, натиснув на відповідну кнопку;
- Вкажіть системі, куди слід перенести обраний ним файл чи папку

Для того щоб скопіювати деякий файл чи папку в іншу папку чи на інший диск, потрібно виконати наступну послідовність дій:

- Виділіть кліком миші файл чи папку, до якої ви бажаєте копіювати
- Оберіть пункт Копіювати файл чи папку у командному меню Задачі для файлів та папок програми Провідник, для чого натисніть на відповідну кнопку
- Вкажіть системі, куди слід скопіювати обраний файл чи папку

Другий спосіб копіювання файлів та папок здійснюється шляхом перетягання файлового об'єкту за допомогою миші з одночасним утриманням в натиснутому стані клавіші Ctrl:

- Клацніть на об'єкті, який ви бажаєте перемістити, мишею;
- Натисніть та утримуйте клавішу Ctrl;
- Перетягуйте об'єкт по екрану, „кинувши” його над зображенням папки, до якої необхідно перемістити цей об'єкт, та одночасно відпустіть клавішу Ctrl.

Видалення та відновлення папок та файлів

Видалити деякий файл чи папку до Корзини можна одним з трьох способів:

Клацніть на файлі, що належить видалити, мишею та, утримуючи її ліву кнопку, “перетягніть” об'єкт по екрану, “кинувши” його над зображенням Корзини в вікні програми Провідник або на Робочому столі Windows.

Виділіть об'єкт, що тріб видалити, мишею в вікні Провідника та клацніть мишею на позначку „Видалити файл або папку” в командному меню Задачі для файлів та папок програми Провідник, натисканням на кнопку;

Клацніть на об'єкті, що видаляється, правою кнопкою миші та, в меню, яке з'явилося, оберіть пункт „Видалити”. У випадку, якщо ви випадково видалили у Корзину ще потрібний вам в роботі файл чи папку, можна відновити даний об'єкт з Корзини у ту папку, звідки він був видалений. Для цього:

- Перейдіть до папки Корзина за допомогою кліка мишею на зображення Корзини в робочому вікні Провідника або подвійним кліком на її позначці, розташованій на робочому столі.
- Виділіть об'єкт, що підлягає відновленню, кліком мишу та потім клацніть на пункті Відновити об'єкт у командному меню Задачі для Корзини. Відновити усі об'єкти, що зберігаються в корзині, можна кліком мишею на пункті Відновити всі об'єкти.

Надалі при бажанні ви можете видалити з Корзини всі об'єкти, що залишились, якщо клацнете мишею на пункті Очистити Корзину в командному меню Задачі для Корзини.

Операції по переміщенню, копіюванню, видаленню, зміні атрибутів файлових об'єктів можна одночасно проводити з цілою групою папок та файлів.

Створення ярликів.

Ярлики є пусковими елементами програми, або графічними значками програм, які розташовано в інших розділах диску файлових об'єктів, що пов'язані з ними динамічним посиланням. Існують три методи створення ярлика деякої програми, файла або папки:

- Клацніть на позначці відповідного файлового об'єкту правою кнопкою миші та в контекстному меню, що з'явиться, оберіть пункт Створити ярлик;
- Перемістить утворений вами ярлик до потрібної папки на диску.

Другий спосіб створення ярликів має на увазі застосування метода перетягування:

- Виділіть потрібний об'єкт кліком миші;
- Натисніть та утримуйте у натиснутому стані клавішу Alt;

- Перетаскуйте створений ярлик по екрану, „кинувши ” його над зображенням папки, до якої необхідно перемістити цей ярлик, та одночасно відпустити клавішу Alt.

Третій спосіб дозволяє створити ярлик для будь-якого файлового об'єкта через контекстне меню Windows XP:

- Перейдіть до папки, в якій ви плануєте створити ярлик деякого об'єкта;
Клацніть в будь-якій вільній від позначок точці папки правою кнопкою миші та в контекстному меню, яке з'явиться, оберіть пункт Створити ► Ярлик;
- На екрані з'явиться вікно мастера Створення ярлика. Додайте до поля Вкажіть розташування об'єкта повний шлях до того файлу, до якого ви бажаєте утворити ярлик.

Програма «Провідник»

Для керування файловими об'єктами, що зберігаються на локальних та з'ємних носіях, а також задля полегшення навігації по файловій системі комп'ютера, у складі Microsoft Windows XP маємо спеціальна програма – файловий менеджер Провідник. Щоб запустити Провідник, потрібно виконати команди Пуск►Усі програми►-Стандартні►Провідник або, клацнувши лівою кнопкою миші на кнопці Пуск, обрати з контекстного меню пункт Провідник. Також можна запустити Провідник, якщо натиснути на панелі інструментів кнопку папка

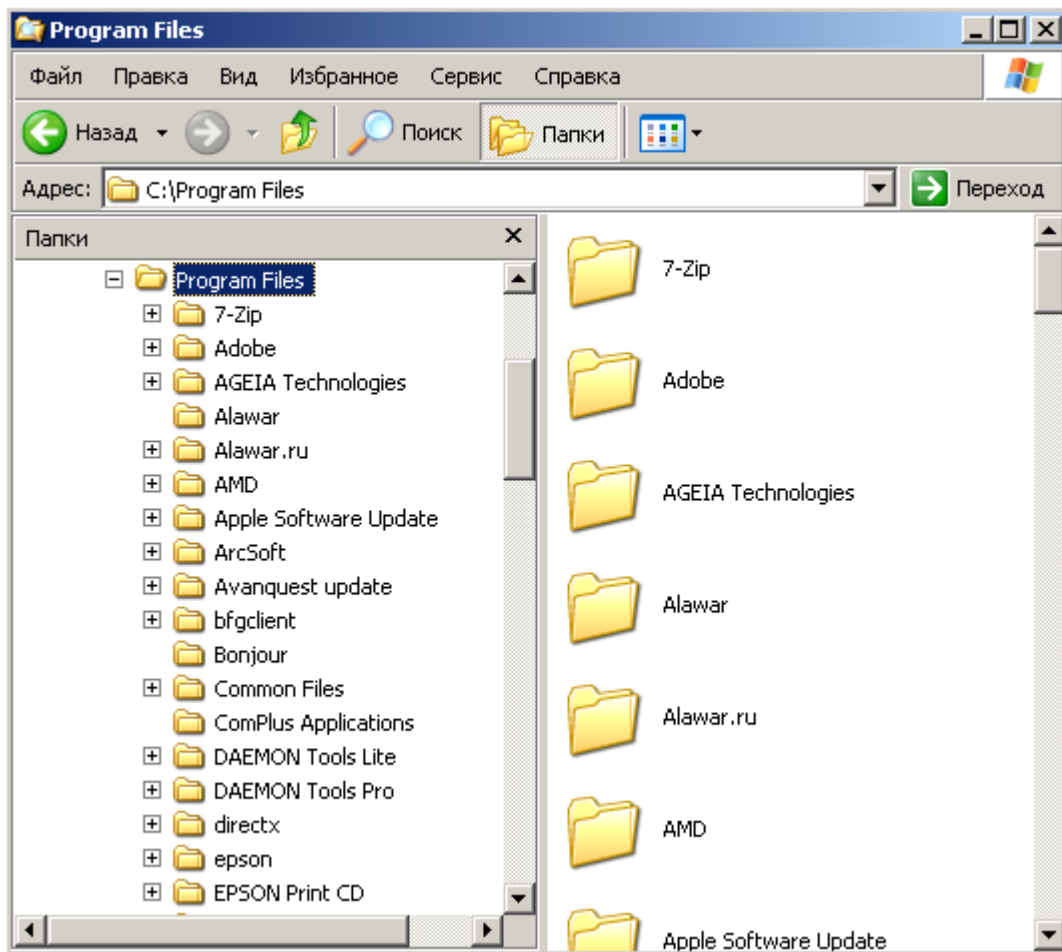


Рис. 2.9. Окно Program Files. В лівій частині відкрите вікно провідника

Робочий простір програми Провідник поділено на два основних поля:

- У лівому відображається вертикальна панель, що надає доступ до різних операцій з файловими об'єктами
- У правому – склад обраного диска чи папки.

Переміщення між папками та дисками виконується за допомогою правого кліка мишою на позначці об'єкта файлової системи у правому робочому полі.

Програма Провідник надає можливість установити своєрідні „закладки” в папках, до яких ви частіш за інші звертаєтесь, завдяки чому ви маєте можливість перейти до потрібної папки за допомогою одного кліка мишою, обравши її назву у меню Вибране.

Зверніть увагу, що біля деяких папок в вікні Провідника стоять позначки „+”. Це означає, що усередині даних папок є вкладені підпапки.

Пошук файлів та папок

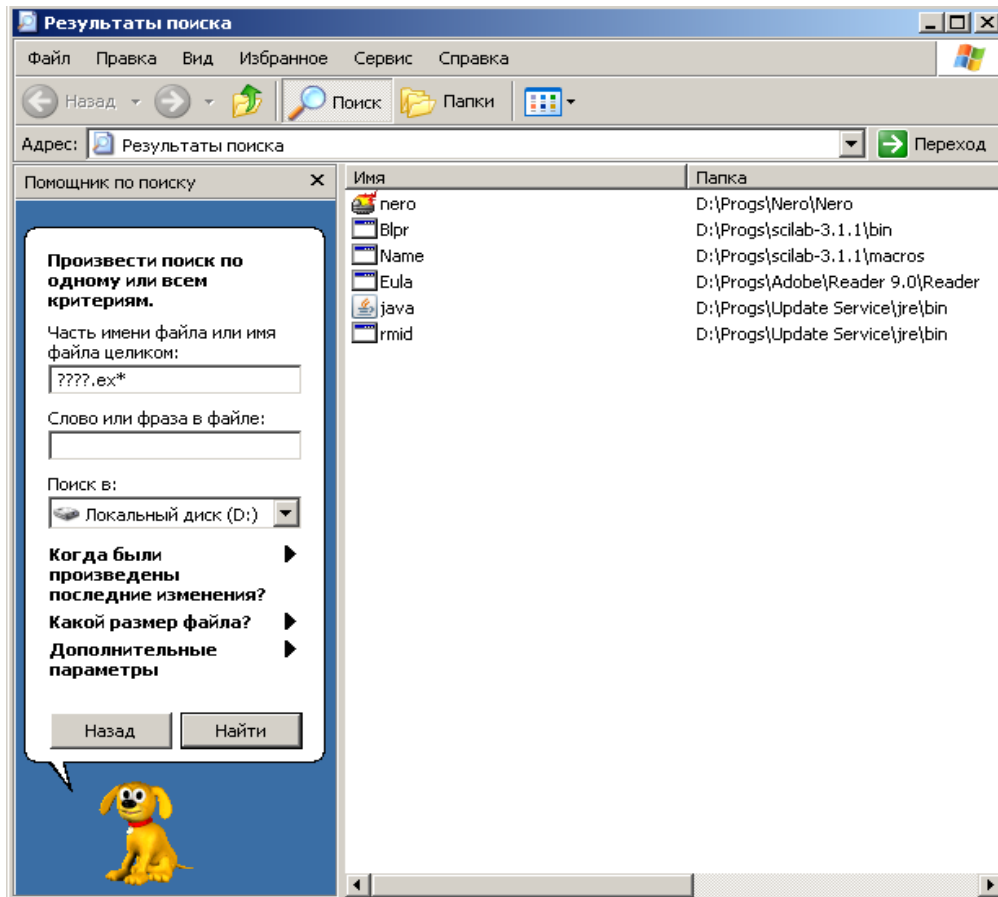


Рис. 2.10. Приклад вікна Пошук. Критерії пошуку – усі файли за ім'ям, що складається з чотирьох літер, розширення починається з літер ex та складається з трьох літер. Пошук проводиться на локальному диску D.

Якщо ви зберігаєте на жорсткому диску велику кількість файлів та папок, то ймовірність серед такої купи легко загубити щось дійсно потрібне. Для пошуку файлів та папок на дисках та в каталогах до програми Провідник Windows XP додано спеціальну утиліту Помічник по пошуку

Викликати на екран вікно локальної пошукової системи Windows Помічник по пошуку можна трьома способами:

1. Клацнувши на пункт *Пошук* в Головному меню Windows
2. Виділивши кліком миші один з дискових накопичувачів, оберіть пункт *Пошук файлів та папок* в меню Системні задачі робочого вікна Провідника чи системного вікна Мій комп'ютер.
3. Клацніть мишею на кнопку *Пошук* на панелі інструментів Провідника.

Пошук файлів та папок за допомогою програми Помічник по пошуку дуже простий. Спочатку необхідно обрати один з режимів пошуку, після чого ввести до відповідного поля ім'я шуканого файла чи папки.

Програма Помічник по пошуку підтримує режим пошуку за маскою, тобто з використанням символів ? та *. Наприклад, якщо ви бажаєте знайти файл на ім'я name та довільним розширенням, то маєте можливість ввести до поля пошуку строку name.*, а якщо забажаєте знайти усі файли с розширенням .doc, скористуйтесь запитом *.doc. Фактично символ ? замінює в строчці запиту будь-який єдиний символ, а на його місце підставляється довільне значення. Наприклад, якщо ви введете до поля програми запит file.?x?, то у вікні результатів пошуку будуть відображені файли, що мають вигляд file.exe чи file.txt.

Запитання для контролю.

1. Які існують дії з файлами та папками?
2. Як зберегти документ Блокнот на зовнішній носій?
3. Як перенести файл з однієї папки до іншої?
4. Як скопіювати файл з однієї папки до іншої?
5. Як створити папку, ярлик?
6. Як видалити файл?
7. Як відновити видалений файл?
8. Як перейменувати файл чи папку?

Приклад контрольного завдання за темою Основи роботи в ОС Windows XP

1. Створіть папку з ім'ям F1.
2. Відкрийте папку F1 та створіть у ній папку F2.
3. Запустіть Блокнот та створіть документ з довільним текстом.
4. Збережіть цей файл як Lab до папки F1.
5. Створіть копію цього файла у папці F2
6. Перейменуйте файл Lab у папці F2 у файл NewLab
7. Переместить файл NewLab з папки F2 до папку F1
8. Видалите файл Lab з папки F1
9. Видалите папку F1, не додаючи її до корзини.
10. Відновіть файл Lab. Де ін буде знаходитися після відновлення?
11. Перенесить папку F1 на зовнішній носій..

3. ВВЕДЕНИЕ В MICROSOFT OFFICE.

Ми починаємо знайомство з найбільш популярним програмним пакетом - Microsoft Office, який працює з Windows. Він включає наступні програми: Word (текстовою редактор), Excel (електронні таблиці); Access (бази даних). Крім того, в пакет входять такі програми як: PowerPoint (творець слайдів), Outlook (записник -ежедневник) - і безліч маленьких корисних програм : ClipArt (галерея картинок), Equation (редактор математичних формул), Word - Art (творець "фігурних" текстів).

Чому ці програми об'єдналися в один пакет? По-перше, можна створити документ, який як єдине ціле містить текст, таблицю, заголовок, малюнок, діаграму.. По-друге, вікна програм мають схожий вигляд, та і основні правила роботи в них схожі. Крім того, перехід з одного застосування в інше можна виконати швидко і просто.

Microsoft Office бывает разных версий. Версия 4.0 работает из 3.11. Версия 7.0 работает из Windows 95. В 1997 г. появилась версия 8.0, более известное название Word 97, в 2000г. появился Word 2000, с появлением операционной системы Windows XP появился Microsoft Office XP.

Після установки файли Microsoft Office розташовуються в теку Program Files\Microsoft Office. Тут в теці Office розташовуються запускаячі програми пакету. Ось вони:

Winword.exe - текстовий процесор
 Excel.exe - електронні таблиці
 Msaccess.exe - редактор бази даних.
 Powerpnt.exe - творець слайдів.
 Outlook.exe - записник.

У Windows для запуску пакету Microsoft Office зручно використовувати панель Microsoft Office, якщо вона встановлена. Відповідальність за неї несе файл msoffice.exe.

Головне призначення панелі Microsoft Office - зручність роботи з документами. З її допомогою можна створювати нові документи і відкривати старі, не замислюючись, в якій програмі, вони

створюватимуться і редагуватимуться.

Коли Windows відкриває який-небудь файл, він дивиться на його розширення і по ньому визначає яку програму треба запустити. Розглянемо які розширення файлів відповідають, яким програмам пакету.

Розширення	Програма-застосування
doc	Редакторы Слово и WordPad
mdb	База даних Access
ppt	Дизайнер слайдів PowerPoint
xls	Електронна таблиця Excel

3.1 MICROSOFT WORD.

3.1.1 ЗАПУСК І ОЗНАЙОМЛЕННЯ З ПАКЕТОМ

Запуск Microsoft Word. Меню і панелі інструментів Word. Клавіші управління текстом.

3.1.1.1 Введення в Microsoft Word.

Microsoft Word є поширеним і багатовіконним текстовим редактором з достатньо великими можливостями. Далі розглядатиметься Microsoft Word 2000.

Запустити редактор можна декількома способами:

1. Натиснути кнопку Microsoft Word  на панелі Microsoft Office.

2. Вибрати пункт меню Пуск □ Программы □ Microsoft Word.
3. Знайти файл C:\program Files\ Microsoft Office\office\winword.exe і запустити його.
4. Запустити потрібний файл з розширенням *.doc.

Після запуску Microsoft Word з'являється вікно нового документа (див. Рис. 3.1.1.1).

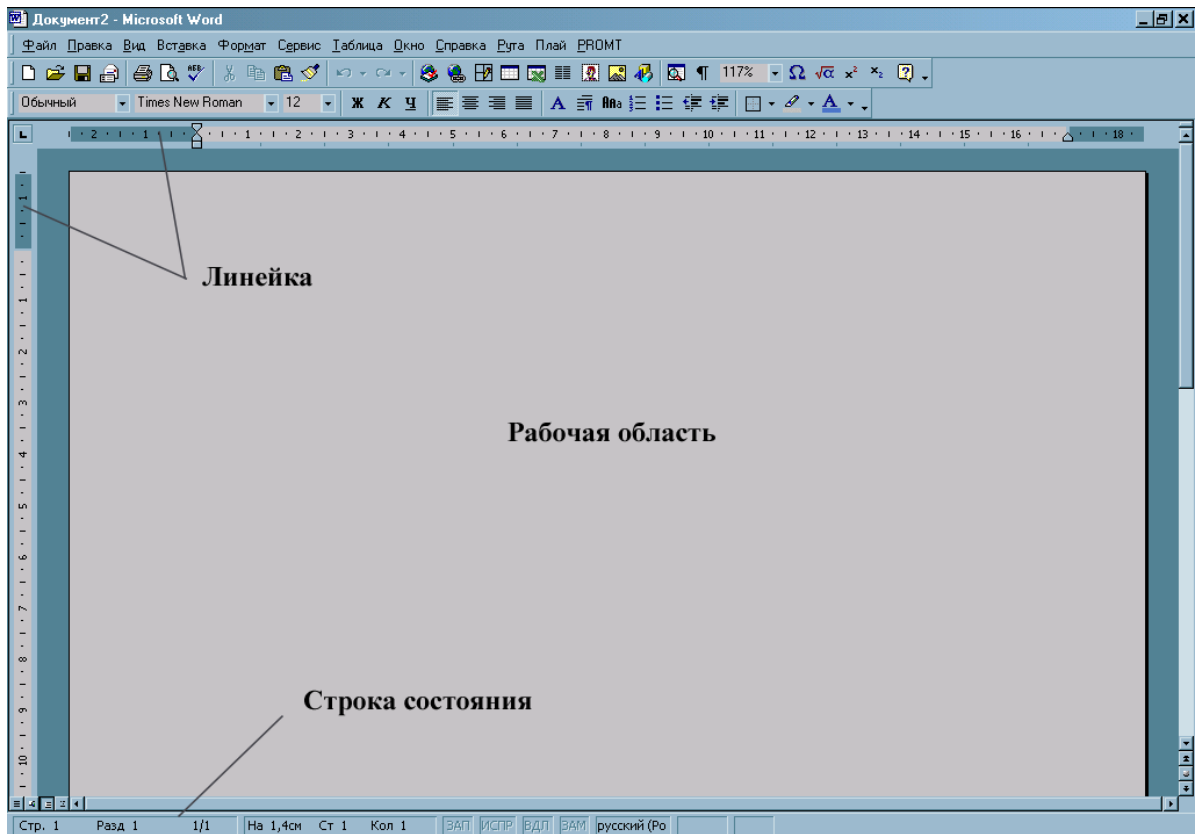


Рис.3.1.1.1 Документ Microsoft Word

Як і будь-яке вікно, вікно нового документа має заголовок, меню, панелі інструментів, робочу область, смуги прокрутки (скролінг).

3.1.1.2 Меню і панелі інструментів Word.

Microsoft Word має наступні пункти меню:

- **Файл** - містить команди для роботи з файлами, налаштування параметрів сторінки, налаштування параметрів друку і друк документа.
- **Правка** - містить команди для відміни дій, для копіювання і пошуку інформації.
- **Вид** – містить команди налаштування виду вікна.
- **Вставка** – містить команди для вставки об'єктів, номера сторінок, символів, малюнків і т.п.
- **Формат** – містить команди по оформленню фрагментів текстів.
- **Сервис** – містить команди для налаштування програми.
- **Таблицы** – містить команди для створення і оформлення таблиць.
- **Окно** – містить команди для дій з вікнами.
- **Справка** – містить довідкову інформацію про програму і роботу з нею.

Включити або вимкнути яку-небудь панель можна через меню Вид ☐ Панели инструментов, відзначивши символом ☒ необхідну панель. Всі панелі мають якусь кількість кнопок - інструментів. Вид кнопок, їх кількість можна змінити, можна також створити свою нову панель.

Щоб створити свою панель натисніть Вид ☐ Панели инструментов ☐ Настройка. У вікні, що з'явилося, перейти на вкладку **Панелі інструментів** і натиснути кнопку **Створити** (див. рис. 3.1.1.2). Потім необхідно дати ім'я новій панелі. Після цього в робочій області з'явиться ваша панель, але поки вона буде без кнопок. Перетягнете нову панель до що вже існує.

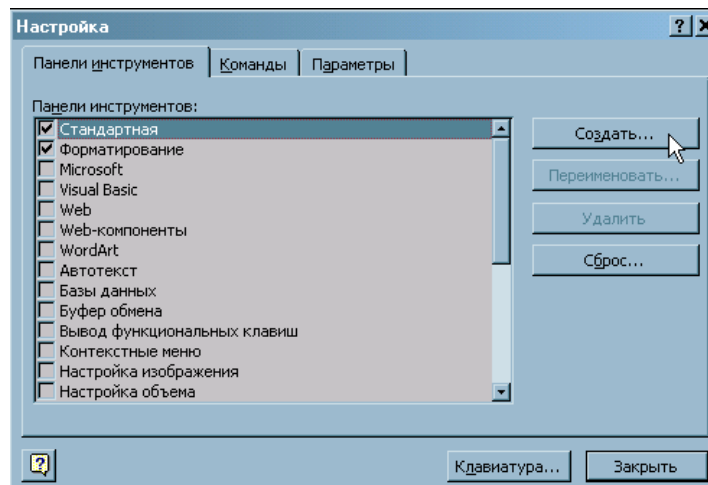


Рис. 3.1.1.2 Вікно налаштування панелі інструментів.

Для додавання кнопок в нову панель необхідно у вікні налаштування панелі інструментів перейти на вкладку **Команди** (див. рис. 3.1.1.3). Вибираєте категорію, а потім команду, яку хочете винести на панель. Методом перетягування переносите команду на нову панель. Можна змінити значок кнопки або перейменувати її. Для цього виділите необхідну кнопку і натисніть на кнопку **Змінити виділений об'єкт**.

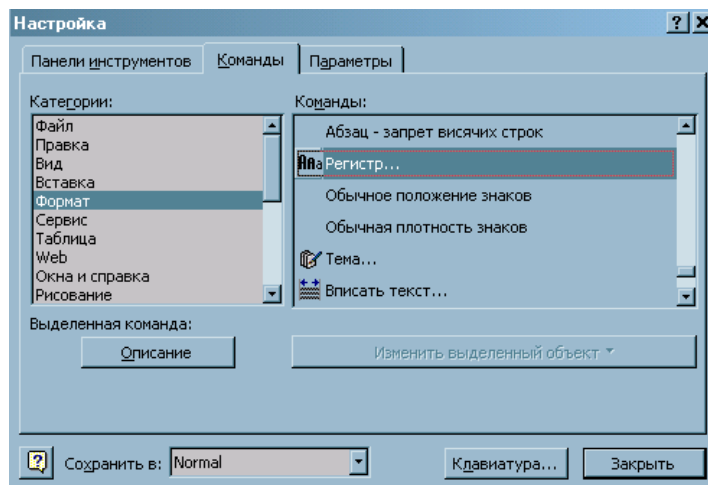







Рис. 3.1.1.3 Налаштування і додавання кнопок

3.1.1.4 Початок роботи з документом.

Для того, щоб створити новий документ необхідно зайти в Файл  Создать або натиснути на кнопку  на панелі інструментів Стандартна. Новий документ носитиме ім'я ДокументN, де N - деяке число

Щоб відкрити існуючий документ необхідно зайти в Файл  Открыть або натиснути на кнопку  на панелі інструментів Стандартна.

Зберегти документ можна в двох режимах: Зберегти  і Зберегти як. У першому випадку документ зберегтися під поточним ім'ям, в другому - можна задати інше ім'я.

Основні клавіші для роботи з текстом:

- ←↑→↓ - клавіші управління курсором;
- Enter – перехід на новий рядок;
- Home, End – перемістити курсор в початок і кінець рядка відповідно.
- Page Up, Page Down – перегортання сторінок вгору і вниз відповідно.
- Insert – режим заміщення;
- Delete – видалення символу, що стоїть після курсора;
- Shift + буква - друк заголовної букви;
- Caps Lock – режим написання заголовних букв;
- Tab – табулювання рядка, слова або символу.

Контрольні питання:

1. Перерахуйте програми, що входять в склад Microsoft Office.
2. Якій програмі відповідає розширення doc?
3. Перерахуйте способи запуску Word.
4. Який пункт меню відповідає за форматування фрагмента тексту?
5. Як створити нову панель інструментів?
6. Як додати кнопки на панель інструментів?
7. Назвіть основні клавіші для роботи з текстом.


3.1.2. НАЛАШТУВАННЯ ПАРАМЕТРІВ СТОРІНКИ

Режими сторінки. Встановлення масштабу. Попередній прогляд підгонка сторінок. Налаштування параметрів сторінки. Вставка номерів сторінок. Верхній і нижній колонтитули.

3.1.2.1 Завдання режиму сторінки

При роботі з документом користувач залежно від виконуваних дій задає відповідний режим перегляду і редагування. У режимі **Звичайний** досягається найбільша швидкість введення і редагування тексту документа. Режим **Електронний документ** орієнтований на зручне читання документа. Режим **Розмітка сторінки** порівняно з попередніми режимами точніше відображає остаточний зовнішній вигляд сторінок документа, крім того, цей режим обов'язковий при роботі з графічними об'єктами, рамками багатоколонним тестом. Режим **Структура** зручний для перегляду і управління структурою документа в цілому.

Для завдання режиму можна використовувати один з наступних способів:

1. Встановити курсор миші на одну з перших чотирьох кнопок, розташованих на горизонтальній лінійці прокрутки . Вибрати потрібний режим можна, натиснувши ліву кнопку миші.

2. Вибрати в меню **Вигляд** необхідний режим за допомогою однойменної команди (див. рис. 3.1.2.1).

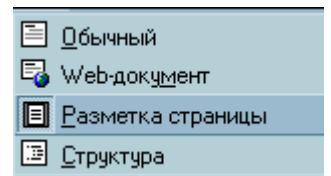
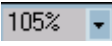


Рис. 3.1.2.1 Режими сторінки в пункті меню Вигляд.

3.1.2.2 Налаштування масштабу документа

Якщо текст на екрані виглядає дуже дрібно або крупно, то потрібно набудувати масштаб. У меню **Вигляд** вибрати команду **Масштаб**. У вікні Масштаб встановити перемикач в потрібний масштаб (див. рис. 3.1.2.2).

Більш зручніше використовувати список, що розкривається, Масштаб на панелі інструментів Стандартна. Якщо бажаного масштабу немає в списку (наприклад, 105%), ввести потрібне значення безпосередньо в список  і натиснути клавішу Enter.

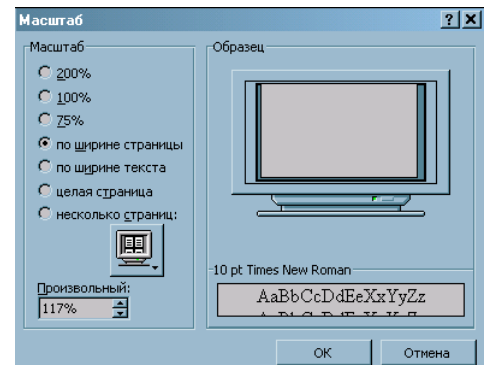



Рис. 3.1.2.2 Діалогове вікно для завдання масштабу

3.1.2.3 Попередній перегляд файлу

Проглядання документа перед друком виконується по команді **Попередній перегляд**  у меню **Файл**. Ця команда перемикає текстовий процесор в однойменний режим (див. рис. 3.1.2.3). У цьому режимі можна отримати узагальнене уявлення про остаточний зовнішній вигляд документа і виконати операцію підгонки рядків тексту.

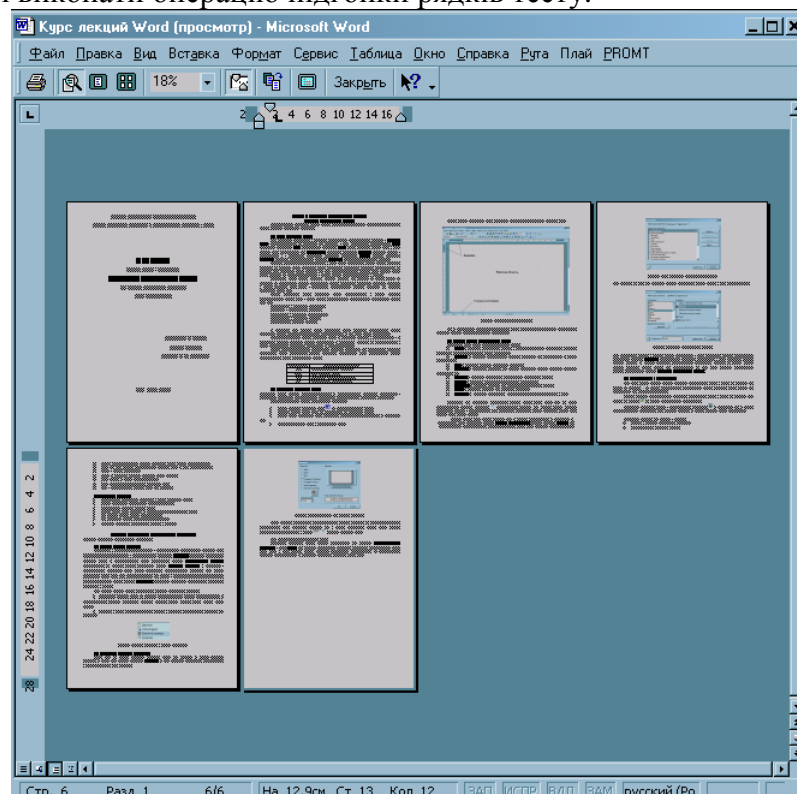





Рис. 3.1.2.3 Вид вікна в режимі попереднього перегляду

Для того, щоб проглянути і підігнати сторінку необхідно виконати наступні дії:

1. Перемкнутися в режим **Попередній перегляд**.
2. Натиснути на панелі інструментів кнопку **Декілька сторінок** .
3. На панелі, що розкрилася, встановити число сторінок документа, що одночасно відображаються (від 1 до 12).
4. Оцінити зовнішній вигляд документа.
5. За наявності одной-двох рядків на окремій сторінці натиснути кнопку **Підгонка сторінок** . В результаті буде виконана підгонка рядків тексту і скорочення числа сторінок документа на одну.
6. За допомогою кнопки **Закреть** повернутися до попереднього режиму перегляду і редагування документа.

3.1.2.4. Друк документа

Щоб надрукувати поточний документ повністю в одному екземплярі, натисніть кнопку **Друк**  на **Стандартній панелі інструментів**. Для того, щоб подивитися, як виглядатимуть надруковані сторінки, натисніть кнопку **Попередній перегляд**.

Якщо необхідно надрукувати не весь документ або декілька копій, то необхідно вибрати команду **Друк** в меню **Файл**.

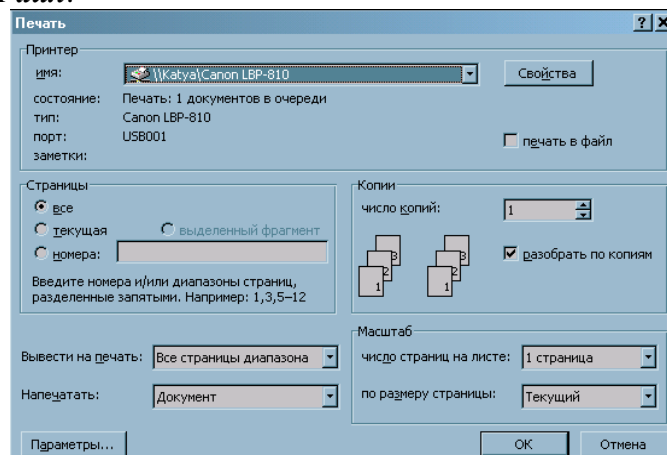


Рис. 3.1.2.4 Вид вікна друку

Вкажіть частину документа, яку потрібно надрукувати, в полі **Сторінки** (див. рис. 3.1.2.4). При виборі значення **Номера** необхідно також вказати номери або діапазони сторінок, які потрібно надрукувати. При виборі значення **Поточна** буде роздрукована сторінка, на якій в даний момент знаходиться курсор.

Якщо потрібно надрукувати декілька копій, то в полі **Число копій** введіть потрібне число копій.

Якщо необхідно друкувати текст з двох сторін сторінки, то необхідно вибрати в полі **Вивести на друк** значення **Непарні сторінки**. Знову закласти ці сторінки в принтер і вибрати значення **Парні сторінки**.

3.1.2.5. Установка параметрів сторінки

Сторінка складається з робочого поля, де можна друкувати текст і полів. Поля розташовуються зліва, вгорі, внизу і справа. Щоб встановити розмір листа документа і поля необхідно зайти в пункт меню **Файл** і вибрати команду **Параметри сторінки**.

Вікно **Параметри сторінки** складається з декількох вкладок (див. рис. 3.1.2.5). На вкладці **Поле** встановлюється розміри кожного поля в сантиметрах (від краю листа до краю тексту). Змінюючи параметри сторінки відразу ж можна подивитися на **Зразок**. Якщо документ

після роздруку брошурують, то встановите прапорець в полі *Дзеркальні поля*. Тоді ліве і праве поля перетворяться на внутрішнє і зовнішнє.

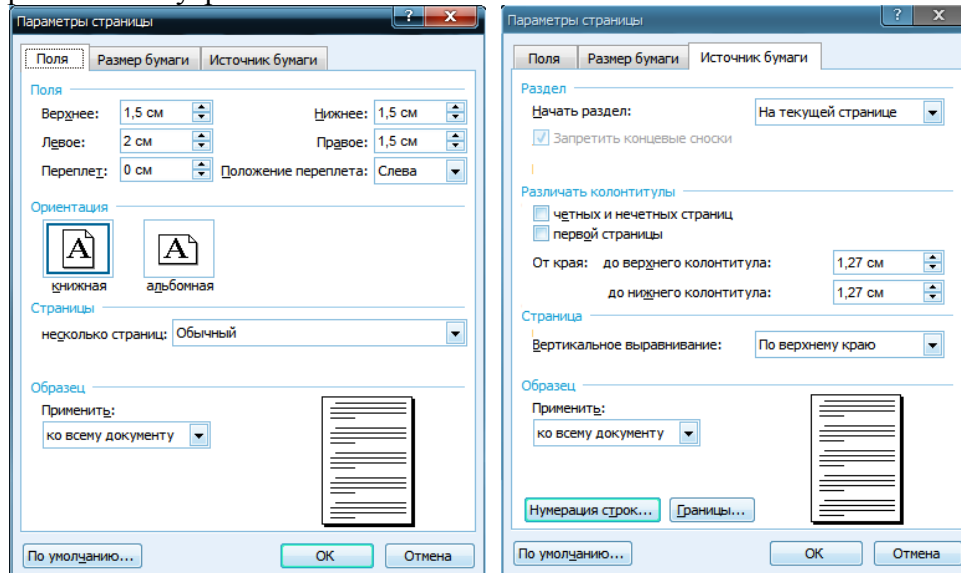


Рис. 3.1.2.5 Вікно Параметри сторінки

На вкладці *Розмір паперу* можна встановити розмір паперу із списку, його ширину і висоту.

Якщо на вкладці *Джерело паперу* в його полях встановлено значення «за умовчанням», то при роздруку використовуватимуться налаштування принтера. Можна скористатися командою *Нумерація рядків*, натиснувши на відповідну кнопку. В результаті з'являється вікно (див. рис. 3.1.2.6), в якому можна задати, з якої сторінки починати нумерацію, з яким кроком нумерувати і де починати нумерацію.

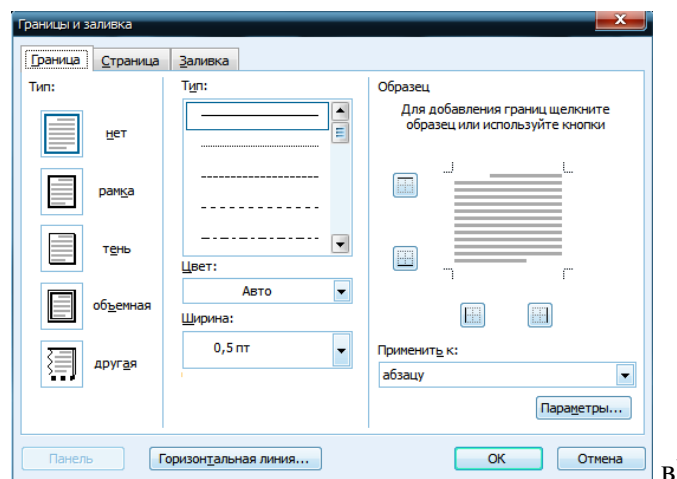
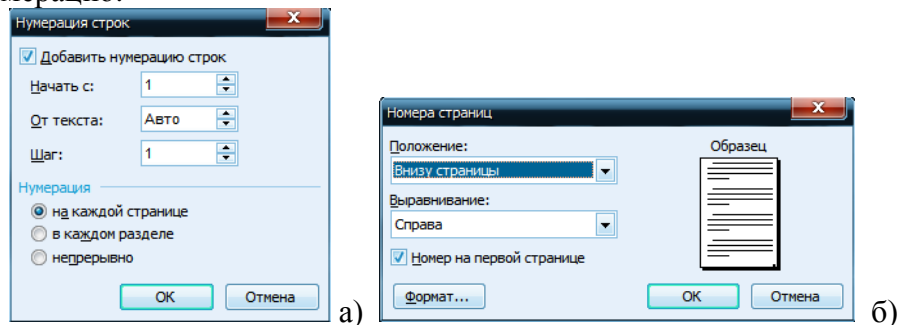


Рис. 3.1.2.6 Вікно Нумерація сторінок

(а - у вікні параметри сторінки, б - через меню вставка, в - Джерело паперу)

Команду **Нумерація сторінок** можна викликати також і в спрощеному варіанті, скориставшись командою, **Номери сторінок** в меню **Вставка**. Також на вкладці **Джерело паперу** можна скористатися командою **Межі**. (аналогічна команді Межа і заливка в меню Формат).

3.1.2.6 Додавання колонтитулів








У верхньому і нижньому полях листа часто друкують службову інформацію. Цю інформацію називають верхнім і нижнім колонтитулом. Так, наприклад, номери сторінок розташовуються або у верхньому або в нижньому колонтитулі.

Щоб перейти в режим роботи з колонтитулом необхідно відкрити меню **Вигляд** і скористатися командою **Колонтитули**. Якщо ж в полі колонтитулу вже є яка-небудь інформація, то тоді достатньо двічі клацнути по ньому мишею. В результаті відкриється панель **Колонтитули** і курсор з'явиться в полі колонтитулу. Текст документа буде недоступний. Щоб перейти в режим роботи з документом, необхідно натиснути кнопку **Закрити** на панелі.



Рис. 3.1.2.7 Панель Колонтитули

Розглянемо деякі кнопки панелі:

- **Верхній/ніжній колонтитул**,  – перехід від верхнього колонтитулу до нижньому і на оборот.
- **Перейти до попереднього**,  – перехід до колонтитулу попередньої сторінки.
- **Перейти до наступного**,  – перехід до колонтитулу наступної сторінки.
- **Номер сторінки**,  – вставити номер сторінки.
- **Дата**,  – вставити поточну дату.
- **Час**,  – вставити поточний час.
- **Параметри сторінки**,  – відкрити вікно Параметри сторінки.

Контрольні питання:

1. Які режими сторінок ви знаєте?
2. Як переключитися на масштаб, який не вказаний в списку?
3. Для чого використовується попередній перегляд?
4. Як роздрукувати поточну сторінку?
5. Як задати орієнтацію листа?
6. Що таке колонтитули і для чого вони використовуються?

3.1.3. НАБІР І РЕДАГУВАННЯ ТЕКСТУ. ШРИФТ.

Установка нових шрифтів. Зміна шрифту тексту. Вставка символу. Декілька способів виділення фрагмента тексту.

Введення тексту виконується в режимі вставки і заміщення. У режимі вставки розташування курсора, що вводяться по місцю, символи автоматично зрушують текст, що знаходиться справа. У режимі заміщення розташування курсора, що вводяться по місцю, текст заміщає символи, що знаходяться праворуч від курсора.

Перемикання між режимами вставки і заміщення виконується за допомогою клавіші Insert. При первинному введенні тексту режими вставки і заміни рівносильні один одному.

Припустимо, що є набраний текст, і ви хочете різноманітнити. Для цього знадобляться засоби для роботи з текстом. Одними з таких засобів є шрифт і абзац.

3.1.3.1 Налаштування шрифтів

У стандартному постачанні Windows є обмежена кількість шрифтів. А Word працює з шрифтами, встановленими в Windows. Виникає питання чи можна збільшити число шрифтів. Відповідь на це питання позитивне.

Для установки нових шрифтів необхідно виконати наступні дії:

1. Відкрити в панелі управління Windows ярлик **Шрифти** (Пуск □ Настройка).
2. Виберіть команду **Встановити новий шрифт** в меню **Файл**. В результаті відкриється вікно **Додавання шрифтів**.
3. Виберіть диск і теку, в якій знаходяться нові шрифти (список шрифтів - це спеціальні файли з розширенням ttf).
4. Виберіть необхідні шрифти і натисніть Ок. Відбудеться установка шрифту (Windows копіює ttf-файли в каталог Windows\Fonts і робить зміни в своїх службових файлах).
5. Після цього шрифти можна використовувати як в самому Windows, так і в інших застосуваннях.

3.1.3.2 Робота з шрифтами

Зміни параметрів шрифту зібрані в команді **Шрифт** в пункті меню **Формат**. Вибравши цю команду, з'явиться вікно **Шрифт** (див. рис. 3.1.3.1).

Вікно складається з трьох вкладок: Шрифт, Інтервал, Анімація.

На вкладці **Шрифт** можна вибрати потрібний шрифт, його зображення і розмір, його колір і тип підкреслення, видозміни. Всі внесені зміни відображаються в полі **Зразок**.

У вкладці **Інтервал** можна набудувати і змінити інтервал між буквами.

У вкладці **Анімація** можна вибрати анімацію на ваш текст: червоні мурашки, мерехтіння, миготливий фон, неонна реклама, фейерверк, чорні мурашки.

Слід пам'ятати, для того, щоб змінити шрифт, необхідно спочатку виділити фрагмент тексту.

За допомогою команди Шрифт можна отримати такі результати:

Диалоговые окна **используют** только с т а н д а р т н ы й шрифт Office и не **используют** другие шрифты.

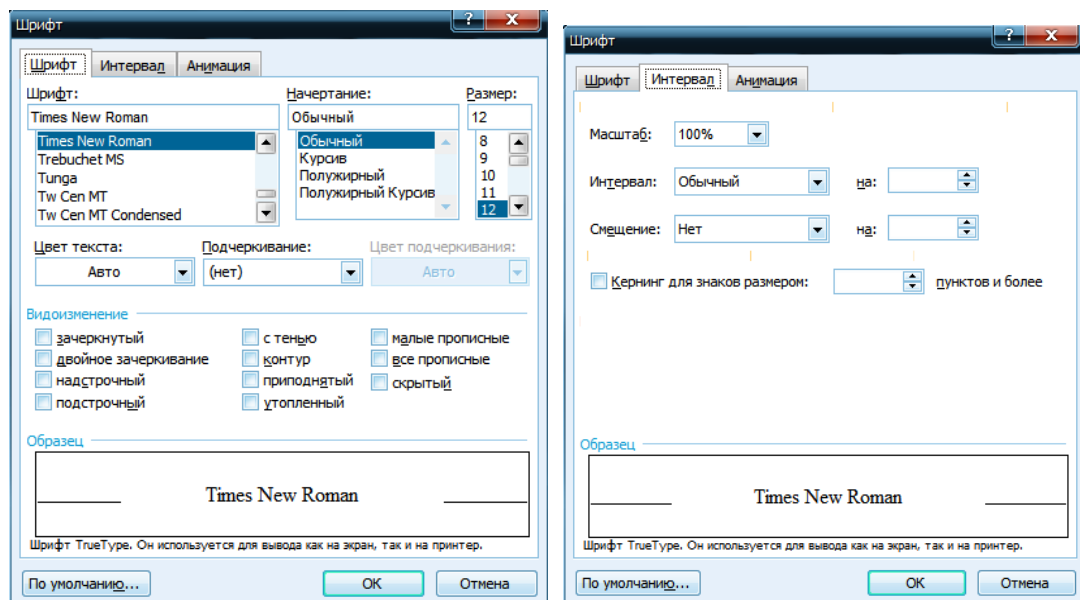


Рис. 3.1.3.1 Вікно Шрифт

Вкладка інтервал дозволяє налаштовувати наступні параметри:

Масштаб. Розтягує або стискає текст у вертикальному і горизонтальному напрямі у відсотках від поточного розміру тексту. Введіть або виберіть значення у відсотках між числами 1 і 600.

Інтервал. Збільшує або зменшує інтервал між знаками. Введіть або виберіть величину інтервалу в полі **на**.

Зсув. Піднімає або опускає текст по відношенню до опорної лінії. Введіть або виберіть

величину зсуву в полі **на**.

Кернінг для знаків розміром. Автоматично змінює інтервал між певними поєднаннями знаків так, щоб зображення цілого слова виглядало більш рівномірним. Цей параметр застосовується тільки для шрифтів Truetype і Adobe Type Manager. Щоб використовувати цей параметр, введіть або виберіть найменший розмір шрифту, для якого слід застосувати кернінг, в полі **пунктів і більш**. Microsoft Word автоматично вибере кернінг для всіх шрифтів вказаного розміру і великих.

3.1.3.3 Вставка символу в документ

При необхідності вставити в документ символ, відсутній на клавіатурі, використовують команду **Символ** в меню **Вставка**. В результаті відкриється вікно для вставки символів (див. рис. 3.1.3.2).

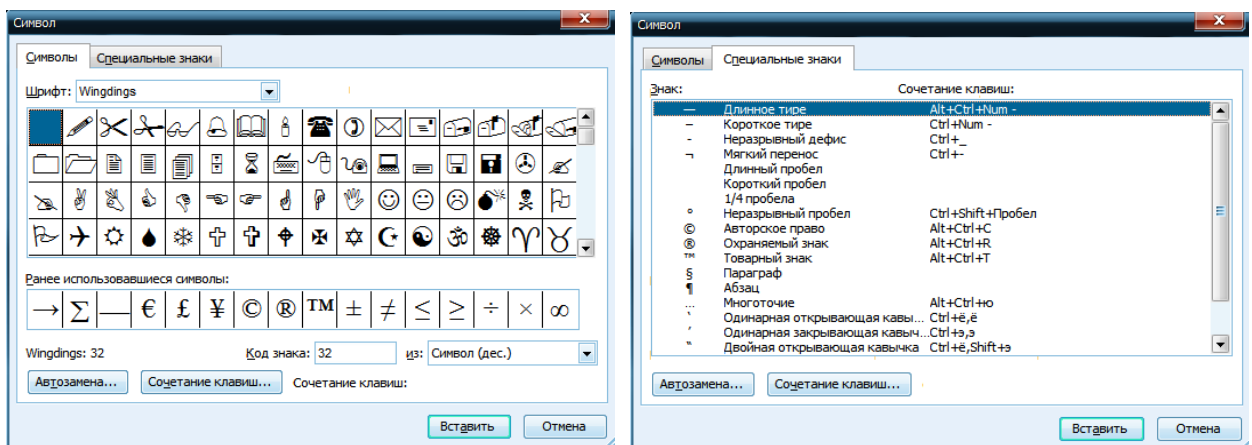


Рис. 3.1.3.2 Вид діалогового вікна вставки символу

Вікно складається з двох вкладок: символи і спеціальні символи. У вкладці **Символи**, в полі **Шрифт** вибирається потрібний шрифт і символ. Можна до будь-якого символу приписати комбінацію клавіш для його виклику. Для цього натисніть кнопку **Клавіша** і вкажіть в полі **Нове поєднання клавіш** необхідне поєднання клавіш.

У вкладці **Спеціальні символи** можна вибрати із списку потрібний спеціальний символ. Наприклад, , ©, ®, ™, §, ¶ і т.п.

3.1.3.4. Виділення фрагмента тексту

Виділення фрагмента документа в звичайному режимі:

1. Помістити покажчик в початок або кінець фрагмента, що виділяється.
2. Далі виконаєте одну з наступних дій:
 - натиснути ліву кнопку миші і виділити фрагмент протяганням покажчика миші в потрібному напрямі;
 - натиснути клавішу **{Shift}** і, не відпускаючи її, за допомогою клавіш управління курсором розширити виділення фрагмента в потрібному напрямі;
 - натиснути клавішу **{Shift}** і, не відпускаючи її, клацнути мишею в кінці області, що виділяється.

Виділення фрагмента в режимі розширення виділення:

1. Помістите покажчик в початок або кінець фрагмента, що виділяється.
2. Включити режим покрокового розширення виділення натисненням клавіші **{F8}** або подвійним клацанням миші в полі **ВДЛ** рядки стану. При цьому букви поля стають чорними.
3. Тепер при послідовних натисненнях клавіші **{F8}** відбувається виділення слова, пропозиції, абзацу і документа в цілому.


Щоб зменшити область виділення (у зворотному порядку: документ, розділ, абзац), досить при натиснутій клавіші **{Shift}** натиснути клавішу **{F8}** потрібне число разів.

Контрольні питання:


1. Які режими введення тексту ви знаєте? Чим вони відрізняються?
2. Як встановити нові шрифти?
3. Яку необхідно викликати команду по роботі з шрифтом? Що з її допомогою можна зробити?
4. Які дії необхідно виконати, щоб вставити в текст символ, відсутній на клавіатурі?
5. Які способи ви знаєте для виділення фрагмента тексту.

3.1.4. НАБІР І РЕДАГУВАННЯ ТЕКСТУ. АБЗАЦ.

Вирівнювання абзаців. Завдання міжрядкового інтервалу і інтервалів перед і після абзаців. Налаштування відступів за допомогою горизонтальної лінійки. Оформлення тексту в декілька колонок. Розривши сторінки. Відміна і повтор дій. Використання буфера обміну. Копіювання і переміщення інформації.

Зміна параметрів абзаців виконується за допомогою діалогового вікна **Абзац** (рис. 3.1.4.1), що викликається по команді  **Абзац...** у меню **Формат**.

3.1.4.1. Завдання вирівнювання і інтервалів

Варіант вирівнювання абзаців зручно встановлювати за допомогою кнопок  панелі інструментів **Форматування**, які задають вирівнювання по лівому краю, по центру, по правому краю і ширині відповідно.

Також можна скористатися відповідним елементом в діалоговому вікні **Абзац** -

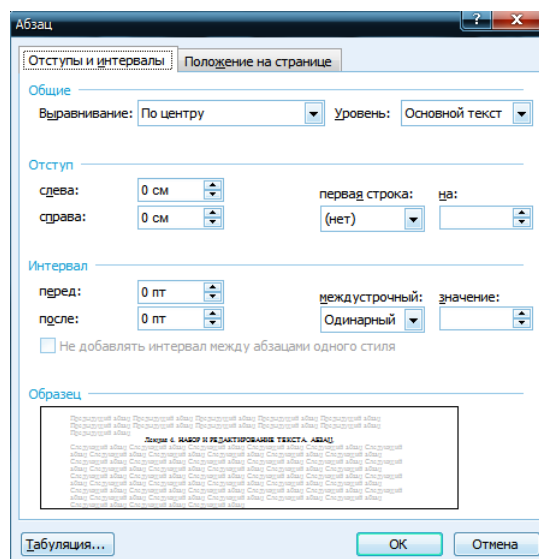
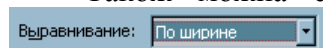


Рис. 3.1.4.1 Діалогове вікно Абзац

Встановити для виділених абзаців необхідний міжрядковий інтервал можна таким чином:

1. Виділити потрібні абзаци і вибрати команду **Абзац**
2. У діалоговому вікні, що відкрилося, вибрати вкладку **Відступи і інтервали**.
3. У списку **міжрядковий** вибрати потрібне значення.

Міжрядковий інтервал може мати одне з наступних значень: одинарний, полуторний, подвійний, мінімум, точно, множник.

Для того, щоб задати інтервали і після абзаців, необхідно в списках *перед* і *після* вибрати або ввести необхідні значення інтервалів в пунктах.

Результат впливу заданих параметрів абзаців можна побачити в полі *Зразок* діалогового вікна Абзац.

3.1.4.2. Завдання відступів

Встановити відступи абзаців зліва і справа, а також відступ або виступ першого рядка.

1. Виділити потрібні абзаци і задати команду *Абзац*....
2. У діалоговому вікні, що відкрилося, вибрати вкладку *Відступи і інтервали*.
3. У списках *зліва*: і *справа*: вибрати або ввести необхідні значення інтервалів в пунктах.
4. У списку *перший рядок*: вибрати *Відступ* (червоний рядок) або *Виступ* і справа в полі *на*: задати величину відступу або виступу першого рядка кожного з абзаців.

Відступи зліва і праворуч від абзацу, а також відступ першого рядка поточного абзацу зручно налаштовувати за допомогою горизонтальної лінійки. Маркери горизонтальної лінійки мають наступні назви: лівий верхній - маркер (або фіксатор) першого рядка, лівий ніжній, - маркер лівого відступу, правий ніжній, - маркер правого відступу.

Налаштування відступів абзацу за допомогою горизонтальної лінійки:

1. Помістити курсор в абзац, що налаштовується.
2. За допомогою команди *Лінійка* в меню *Вигляд* встановити горизонтальну лінійку у вікні документа, якщо вона не була встановлена раніше.
3. Переміщаючи маркер першого рядка (див. рис. 3.1.4.2), встановити потрібний відступ першого рядка.
4. За допомогою маркерів лівого і правого відступу (див. рис. 3.1.4.2) встановити лівий і правий відступи абзацу.

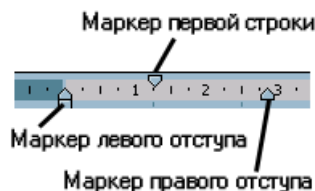


Рис. 3.1.4.2. Розташування маркерів на горизонтальній лінійці

3.1.4.3. Оформлення тексту в декілька колонок

У Word є можливість представляти текст в декількох колонках.

1 спосіб

1. У меню *Формат* вибрати команду *Колонки*.
У вікні *Колонки* (рис. 3.1.4.3), що відкрилося, ввести необхідне число колонок і задати їх тип.

2 спосіб

1. Клацніть кнопку *Колонки* на *Стандартній панелі інструментів*.



2. У вікні, що з'явилося вкажіть число колонок

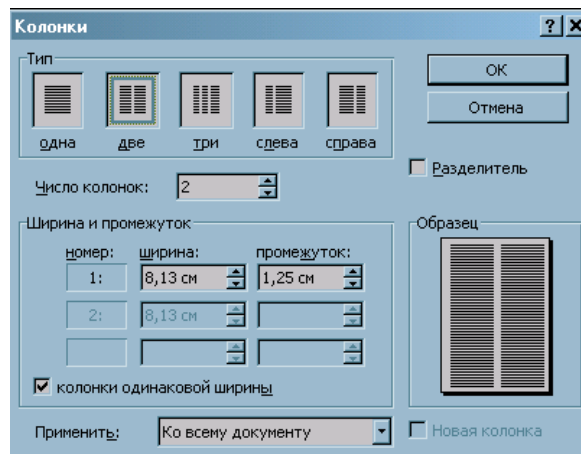


Рис. 3.1.4.3 Діалогове вікно Колонки

3.1.4.4. Створення списків

Перерахування, в якому кожне нове значення перерахування розташовується з нового рядка, називатимемо списком. У Word розрізняють нумеровані, маркіровані і багаторівневі списки.

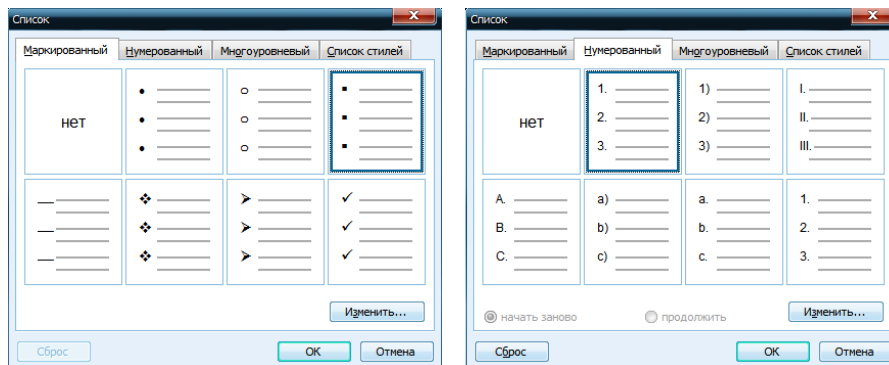


Рис. 3.1.4.4 Діалогове вікно для створення списку

Для того, щоб створити новий список, необхідно виконати наступні дії:

1. Викликати команду **Список** в меню **Формат**.
2. У діалоговому вікні, що з'явилося, вибрати необхідний вид списку (див. рис. 3.1.4.4) і натисніть ОК.

3. Якщо необхідно змінити вид списку, то скористайтесь кнопкою **Змінити**.

4. Введіть текст. Для переходу до наступного пункту натисніть Enter.

Якщо вже існує текст і його необхідно оформити у вигляді списку, то для цього виділіть необхідний фрагмент тексту і виконаєте всі дії описаний вище.

3.1.4.5. Вставка розриву сторінки або розділу

Буває така ситуація, коли необхідно, щоб деяка частина тексту починалася обов'язково з нового рядка. Для цього необхідно виконати наступні дії:

1. Помістити курсор в місце вставки розриву.
2. У меню **Вставка** вибрати команду **Розрив**.
3. У діалоговому вікні, що відкрилося, встановити потрібний перемикач (див. рис. 3.1.4.4) і натиснути ОК.

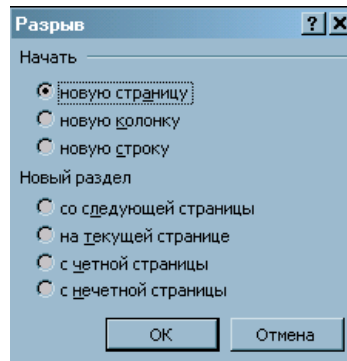


Рис. 3.1.4.5 Діалогове вікно Розрив

3.1.4.6. Відміна і повторення виконаних дій.

Word надає зручний засіб відміни помилково виконаних дій з редагування. Щоб відмінити останню виконану дію, досить виконати команду **Відмінити** в меню **Правка**. Для повторення останньої скасованої дії можна виконати команду **Повторити**. Вказані дії з відміни і повторення введення зручно виконуються клацанням миші відповідно на лівій і правій кнопках з дуговими стрілками на панелі інструментів **Стандартна**.

Word запам'ятовує в стеку до сотні останніх виконаних дій, які можна відмінити і повторити наново. Клацанням миші по одній з кнопок з трикутником можна проглянути весь список дій з редагування або відміни редагування. Вибравши елемент в одному із списків, і клацнувши по ньому мишею, можна відмінити або повторити цілу ланцюжок дій.

3.1.4.7. Копіювання і переміщення інформації.

При створенні документа часто виникає необхідність в копіюванні і переміщенні фрагментів тексту. Це можна виконати, використовуючи метод drag-and-drop (перенести і залишити) або використовуючи буфер обміну.

Переміщення і копіювання виділеного фрагмента за допомогою миші методом drag-and-drop (перенести і залишити):

1. Заздалегідь задати команду **Параметри...** у меню **Сервіс** і на вкладці **Правка** діалогового вікна **Параметри** встановити прапорець **Використовувати перетягання тексту при правці**.

2. Помістите покажчик миші в область виділеного фрагмента, і після ухвалення ним форми нахиленої вліво стрілки, натиснути ліву кнопку миші. В результаті у нижнього кінця покажчика з'явиться прямокутник, а текстовий курсор набуде форми вертикальної штрихової лінії.

3. Протягнути покажчик так, щоб текстовий курсор опинився в місці передбачуваної вставки виділеного фрагмента, і відпустити кнопку миші.

4. Якщо передбачуване місце вставки не відображається на екрані, протягнути покажчик до верхньої або нижньої межі вікна документа. Прокрутити, таким чином, текст у вікні в потрібному напрямі (вгору або вниз).

5. Копіювання виділеного фрагмента виконаємо аналогічно, але з натиснутою клавішею **{Ctrl}**. При цьому у прямокутника в нижньому кінці покажчика з'являється хрестик.

У будь-якому додатку Windows, включаючи Word, є команди **Вирізувати**, **Копіювати** і **Вставити**. Вони дозволяють спростити дії з редагування документів, пов'язані з вирізуванням, копіюванням і переміщенням різних фрагментів документа, включаючи текст, рисунки і таблиці.

При реалізації названих команд використовується буфер обміну. Він є ділянкою оперативної пам'яті, використовуваною для тимчасового зберігання інформації при передачі в межах однієї програми або між декількома програмами.

За допомогою команд **Вирізувати** і **Копіювати** копія виділеного фрагмента поміщається в буфер обміну (відмінність між командами полягає в тому, що в першому випадку виділений фрагмент вирізується (віддаляється), в другому - ні. Порцію інформації з буфера обміну можна

використовувати багато разів, вставляючи її в різні місця різних застосувань. Інформація в буфері обміну залишатиметься незмінною до тих пір, поки знову не буде виконана команда **Копіювати** або **Вирізувати** або не буде завершений сеанс роботи в Windows.

У Word існує так зване **контекстно-залежне меню**, яке висвічується при натисненні правої кнопки миші. Контекстно-залежне меню підвищує комфортність роботи, оскільки виключає необхідність пошуку необхідних операцій в меню. У контекстно-залежне меню включені всі команди, які можуть бути застосовані до активного (виділеному) об'єкту.

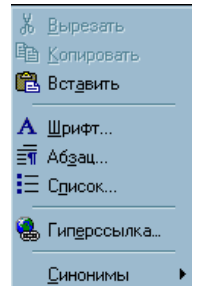


Рис. 3.1.4.6 Приклад контекстно-залежного меню



При виконанні клацання правою кнопкою миші на виділеному об'єкті прямо біля покажчика миші відкривається меню з основними доступними функціями, вживаними в даній ситуації, - в контексті.

Існують наступні способи копіювання інформації:

Спосіб 1

1. Активізувати документ-джерело і виділити в нім фрагмент, який потрібно скопіювати.
2. Виконати команду **Копіювати** в меню **Правка** і перемкнутися в документ-приймач.
3. Помістити покажчик в місце вставки копії фрагмента.
4. Виконати команду **Вставити** в меню **Правка**.

Спосіб 2

1. Активізувати документ-джерело і виділити в нім фрагмент, який потрібно скопіювати.
2. Вибрати кнопку **Копіювати**  на стандартній панелі інструментів і перемкнутися в документ-приймач.
3. Помістити покажчик в місце вставки копії фрагмента.
4. Вибрати команду **Вставити**  на стандартній панелі інструментів.

Спосіб 3

1. Активізувати документ-джерело і виділити в нім фрагмент, який потрібно скопіювати.
2. Над виділеним фрагментом викликати контекстно-залежне меню, в якому вибрати команду **Копіювати**.
3. Помістити покажчик в місце вставки копії фрагмента.
4. У місці вставки фрагмента викликати контекстно-залежне меню, в якому вибрати команду **Вставити**.

Контрольні питання:

1. Яка команда працює з абзацом?
2. Як можна задати вирівнювання абзацу?
3. Як встановити подвійний міжрядковий інтервал?
4. Як задати червоний рядок в абзаці?
5. Які способи ви знаєте по налаштування відступів зліва і справа?
6. Як перенести частину тексту на нову сторінку?
7. Як можна відмінити неправильно зроблену дію?
8. Які ви знаєте способи копіювання інформації?
9. Що таке буфер обміну?

3.1.5. ВСТАВКА ОБ'ЄКТІВ.

Використання верхнього і нижнього індексів для написання формул. Введення математичних формул за допомогою редактора формул. Створення об'єктів Wordart. Використання панелі Малювання.

3.1.5.1 Створення простих формул

Для введення простих формул досить використовувати вставку верхнього і нижнього індексів. Це ви зможете виконати наступними способами:

Спосіб 1

1. Увійти до меню **Формат** або в контекстно-залежне меню і вибрати команду **Шрифт**.

2. У вікні **Шрифт** вибрати вкладку **Шрифт** і в полі **Ефекти** покажчиком миші відзначити нижній індекс/верхній індекс як вид ефекту. При цьому у відповідному віконці з'явиться галочка.

3. Після закінчення введення нижнього індекса/верхнього індексу символів потрібно увійти до меню **Формат** або в контекстно-залежне меню і вибрати команду **Шрифт**. У вікні **Шрифт** в полі **Ефекти** покажчиком миші відмінити нижній індекс/верхній індекс вид ефекту.

Спосіб 2

1. Увійти до меню **Вигляд** і вибрати команду **Панелі інструментів**, потім команду **Налаштування**. В результаті відкриється діалогове вікно **Налаштування**.

2. У вікні **Налаштування** вибрати вкладку **Команди**. У полі **Категорії** вибрати **Формат**, а потім в полі **Команди** виділити піктограму x^2 верхній індекс (нижній індекс x_2) і перемістити на панель інструментів **Форматування** при натиснутій кнопці миші.

3. Для введення індексів натискаємо відповідні піктограми на панелі інструментів **Форматування**.

3.1.5.2. Запис формул за допомогою редактора формул

Запис і редагування складних формул в **Word** ведеться за допомогою програми **Редактор формул 3.0 (Equation Editor 3.0)**. Він дозволяє вставляти в документ математичні формули, що включають знаки операцій, вирази, дробі, ступені, інтеграли і тому подібне. При записі формули застосовується автоматичне застосування відповідних стилів для різних її компонентів (зменшений розмір шрифту для показників ступеня, курсив для змінних і т. п.). Освоїти роботу з редактором порівняно нескладно, хоча сам процес запису формул достатньо трудомісткий. З іншого боку, слід зазначити простоту зміни розміру готових формул.

Запуск редактора формул Microsoft Equation 3.0 (кнопка на панелі інструментів \sqrt{x}):

1. Помістити курсор на місце введення і редагування формули.

2. У меню **Вставка** вибрати команду **Об'єкт ...**

3. У діалоговому вікні, що відкрилося, **Об'єкт**, на вкладці **Створення** в полі **Тип об'єкту** вибрати **Microsoft Equation 3.0**. Натиснути ОК.

В результаті документ перейде в режим створення або редагування формул і з'явиться панель для роботи з редактором формул (див. рис. 3.1.5.1). У формулу за допомогою клавіатури можна ввести букви російського і латинського алфавітів, а також знаки простих математичних операцій «+», «-», «/» і операцій стосунків «<», «>».



Рис. 3.1.5.1 Панель для написання формул.

У верхньому рядку панелі інструментів зліва направо розташовані наступні палітри символів: **Символи стосунків**, **Пропуски і багатокрапки**, **Надрядкові знаки**, **Оператори**,

Стрілки, Логічні символи, Символи теорії множин, Різні символи і Грецькі букви - рядкові і прописні.

За допомогою шаблонів панелі інструментів виконується вставка у формулу знаків ряду математичних операцій, завдання символів інтегралів, сум, творів. Крім того, шаблони дозволяють задати форму математичного виразу (дробу, ступеня, індексу, матриці і тому подібне) для подальшого введення математичних символів в заготовку, отриману за допомогою шаблонів. Вставка шаблонів виконується аналогічно вставці математичних символів.

У нижньому рядку панелі інструментів зліва направо розташовані наступні палітри шаблонів: **Шаблони дужок, Шаблони дробів і радикалів, Верхні і нижні індекси, Суми, Інтеграли, Надчеркивання і підкреслення, Стрілки з текстом, Твори і символи теорії множин, Шаблони матриць.**

При наборі символів формули курсор введення має форму знаків \square або \square . Символ, що вводиться у формулу, розміщується справа або зліва від вертикальної межі і над горизонтальною лінією курсора введення.

Запуск редактора формул для коректування наявної формули виконується подвійним клацанням миші в полі формули.


Завершення і редагування або записи формули виконується клацанням миші за межами рамки для введення формули.

Зміна розміру формули виконується прямо в основному вікні документа Word після попереднього виділення її клацанням миші.

3.1.5.3. Вставка об'єктів Wordart.

Створення і налаштування текстових ефектів в Word виконується за допомогою вбудованих засобів.

Для завдання текстових ефектів створюються об'єкти **WordArt**. З їх допомогою можна задати всілякі варіанти оформлення текстових фрагментів: з різним розташуванням лінії тексту, використанням тіней, обертанням рядків тексту і нахилу букв, завданням обрамлення і заливки, розтяжкою букв по горизонталі і по вертикалі і тому подібне. У багатьох відношеннях налаштування параметрів об'єкту WordArt практично не відрізняється від налаштування параметрів графічного об'єкту Word.

Для створення об'єкту WordArt і налаштування параметрів текстового ефекту використовується панель інструментів WordArt . Вона викликається командою **Рисунок** \square **Об'єкт WordArt** у меню **Вставка**. В результаті відкриється вікно для роботи з WordArt (див. рис. 3.1.5.2).

Щоб вставити створити об'єкт WordArt, необхідно виконати наступні дії:

1. Відкрити діалогове вікно WordArt.
2. Клацанням миші вибрати потрібний варіант і натиснути ОК. При цьому відкриється діалогове вікно **Зміна тексту WordArt**.
3. У полі текст: ввести потрібний текст, до якого застосовуватиметься створюваний ефект Word Art.
4. У списку **Шрифт**: вибрати необхідну гарнітуру шрифту.
5. Уточнити розмір шрифту в пунктах, при необхідності встановити потрібне зображення (напівжирний і /ілі курсив). Натиснути ОК.

В результаті відбудеться вставка в документ відредагованої текстової фрази з вибраним варіантом і параметрами текстового ефекту.

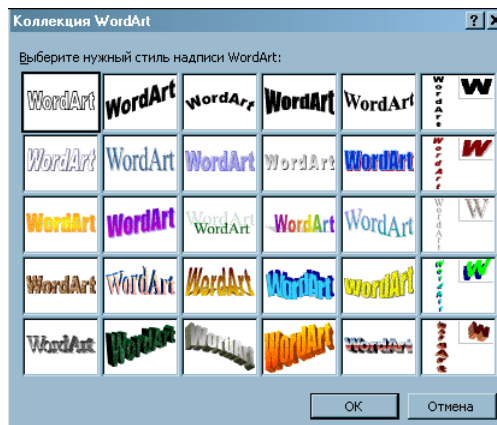


Рис. 3.1.5.2 Діалогове вікно Колекція WordArt.

3.1.5.4. Робота з ClipArt

Створення достатньо складних композицій може бути дуже трудомісткими. У таких випадках використовують готові бібліотеки (колекції) малюнків (*ClipArt*), які входять в комплект постачання пакету Microsoft Office.

Для вставки ClipArt використовують команду **Вставка, Рисунок, Картинки**. Діалогове вікно, що відкривається при цьому, називається **Вставка картинки** (див. рис. 3.1.5.3).

Графічні ClipArt зосереджені на вкладці **Рисунки**. Вони класифіковані по категоріях. Розшукавши потрібний рисунок, досить виділити його і вставити в документ клацанням на кнопці **Вставити**.

В результаті відбудеться вставка в документ вибраного рисунка.

Для зміни формату, необхідно виділити об'єкт, викликати контекстне меню і в ньому вибрати команду **Формат рисунка**.

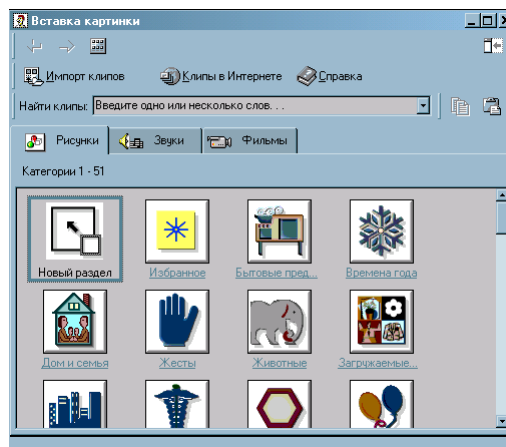


Рис. 3.1.5.3 Вікно Вставка картинки.

3.1.5.5. Створення графічних об'єктів.





Для створення своїх графічних об'єктів можна скористатися стандартною панеллю інструментів для малювання. Вона викликається натисненням на кнопку  у стандартній панелі інструментів.





Рис. 3.1.5.4. Панель Малювання.


Команда **Лінія** : Малювання лінії при переміщенні миші з натиснутою лівою кнопкою. Щоб кут нахилу лінії був кратний 15 градусам, при малюванні утримуйте натиснутою клавішу **Shift**. Щоб лінія продовжувалася в обидві сторони від початкової точки, при виконанні описаних вище дій утримуйте натиснутою клавішу **Ctrl**.

Команда **Стрілка** : Малювання ліній із стрілкою на кінці.


Команда **Прямокутник** : Малювання прямокутника. Для малювання квадрата, слід утримувати натиснутій клавішу **Shift**.


Команда **Овал** : Малювання овалу в поточному вікні. Для малювання круга, слід утримувати натиснутій клавішу **Shift**.


Команда **Напис** : Для створення напису в активному вікні. Написи використовуються для розміщення тексту, такого як заголовки або винесення, на малюнках і діаграмах.

Команда **Колір заливки** : Додавання, зміна або видалення кольору заливки для виділеного об'єкту. Заливка може бути градієнтною, текстурою, візерунковою або складатися з рисунків.


Команда **Колір ліній** : Додавання, зміна або видалення кольору ліній для виділеного об'єкту.


Команда **Колір шрифту** : Форматування виділеного тексту заданим шрифтом.


Команда **Товщина лінії** : Вибір типу і товщини вибраної лінії.

Команда **Тип штриха** : Вибір штрихової або штрих-пунктирної лінії для виділеної рамки або лінії.

Команда **Вигляд стрілки** : Вибір типу стрілки для виділеної лінії.

Команда **Тип тіні** : Вибір типу тіні для виділеного об'єкту.

Команда **Стилі об'єму** : Вибір стилю для об'ємного ефекту.

Команда **Вибір об'єктів** : Перетворення покажчика в стрілку, що дозволяє вибирати об'єкти. Для виділення одного об'єкту досить просто вибрати його. Для виділення декількох об'єктів слід виділити їх, натиснувши на кнопку **Вибір об'єктів** і обвівши їх покажчиком (при цьому з'явиться пунктирний квадрат - всі об'єкти, що виділяються, повинні знаходитися усередині нього).

Команда **Групувати** (меню **Дії**): Декілька виділених об'єктів збираються в єдиний об'єкт для того, щоб можна було їх змінювати або переміщати одночасно.

Команда **Розгрупувати** (меню **Дії**): Декілька зібраних в групу об'єктів розділяються так, щоб кожен з них міг бути змінений або переміщений окремо.

Команда **Перегрупувати** (меню **Дії**): Об'єднання в групу об'єктів, які були роз'єднані командою **Розгрупувати**.

Команда **На передній план** (меню **Дії**): Виділений об'єкт розташовується по верх всіх об'єктів, що перекриваються.

Команда **Перемістити вперед** (меню **Дії**): переміщення об'єкту або групи об'єктів на один рівень вгору (ближче до поверхні екрану) в наборі об'єктів, що перекриваються.

Команда **Перемістити назад** (меню **Дії**): переміщення об'єкту або групи об'єктів на один рівень вгору вниз (далі від поверхні екрану) в наборі об'єктів, що перекриваються.

Команда **Помістити перед текстом** (меню **Дії**): Виділений об'єкт поміщається попереду тексту.

Команда **Помістити за текстом** (меню **Дії**): Виділений об'єкт поміщається позаду тексту.

Команда **Сітка** (меню **Дії**): Установка режиму прив'язки до сітки, який дозволяє легко вирівнювати об'єкти при малюванні.

Існує можливість автоматичного створення фігур, використовуючи інструменти **Автофігури** Автофигуры ▾ на панелі малювання. Меню **Автофігури** містить декілька категорій фігур. Окрім ліній в ній містяться основні форми, фігурні стрілки, елементи блок-схем, зірки і прапори, а також винесення.

Для малювання автофігури натисніть кнопку **Автофігури**, а потім виберіть необхідну фігуру у відповідній категорії. Клацніть по документу для додавання фігури з розміром, встановленим за умовчанням, або розтягніть її до необхідного розміру.

Існує можливість зміни фігури. Для цього виділіть автофігуру, яку необхідно замінити, натисніть кнопку Дії на відповідній панелі, а потім виберіть необхідну автофігуру із списку **Змінити автофігуру**.

Контрольні питання:

1. Як можна створити прості формули?
2. Як запустити редактор формул Microsoft Equation?
3. Які шаблони вам необхідно задіювати для написання слід. формули: $\int x^m dx = \frac{x^{m+1}}{m+1} + c$?
4. Що таке об'єкт WordArt?
5. Які можливості надає панель Рисование?
6. Що таке згрупувати і розгрупувати об'єкти?
7. Що можна створити використовуючи команду Автофігури?

3.1.6. РОБОТА З ТАБЛИЦЯМИ.

Поняття таблиці і її складових. Способи побудови таблиці. Основні клавіші для роботи з таблицями. Вставка і видалення рядків і стовпців. Форматування таблиці. Побудова діаграми за даними таблиці. Написання простих формул для обчислень усередині таблиці.


Word має вбудований засіб створення і зміни таблиці. Основні елементи таблиці - **рядки, стовпці, ячейки**. **Ячейка** - це клітка, розташована на перетині рядка і стовпця. У кожен осередок можна занести дане - число, текст, до яких застосуємо загальний принцип форматування тексту (вирівнювання, розмір і стиль шрифту і тому подібне), дату, графіка, посилання на дані з інших документів. Для наочності таблиці можна задавати розмічальні лінії сітки, видимі або невидимі при друці. При створенні і редагуванні таблиць зручно використовувати панель інструментів Таблиці і межі, яка включається натисненням на кнопку  на панелі інструментів Стандартна.



Рис. 3.1.6.1. Панель інструментів Таблиці і Межі


3.1.6.1 Створення таблиці.

Таблиця в документі створюється наступними способами:

Спосіб 1

1. Помістити курсор в місці розміщення таблиці.
2. Виконати команду **Додати Таблицю** в меню **Таблиця**.
3. У полях **Число стовпців:** і **Число рядків:** задайте відповідні параметри таблиці. Натисніть ОК.

Спосіб 2

1. Помістити курсор в тому місці документа, куди потрібно вставити таблицю.
2. У Стандартній панелі інструментів вибрати значок *Додати таблицю* .
3. У діалоговому вікні, що з'явилося, покажчиком миші виділити необхідну кількість рядків і стовпців.
4. На робочому полі документа з'явиться сітка таблиці.

Спосіб 3

1. Набрати текст, відокремлюючи вміст майбутніх осередків заданим символом (крапка, двокрапка, табуляція і т.п.).
2. Виділити текст.
3. Вибрати команду *Перетворити, Перетворити в таблицю* в меню *Таблиця*.
4. У вікні, що з'явилося, вказати кількість стовпців майбутньої таблиці, вказати символ-обмежувач і натиснути кнопку ОК.

3.1.6.2 Основні комбінації клавіш для роботи з таблицями.

Переміщення по таблиці можна здійснювати наступними клавішами:

- **{Tab}** – до наступного осередку в поточному рядку
- **{Shift+Tab}** – до попереднього осередку в поточному рядку
- **{Alt+Home}** – до першого осередку поточного рядка
- **{Alt+PageUp}** – до першого осередку поточної стовпця
- **{Alt+End}** – до останнього осередку поточного рядка
- **{Alt+PageDown}** – до останнього осередку поточної стовпця
- **↑** – до попереднього рядка
- **↓** – до наступного рядка

Выделение элемента таблицы можно осуществлять следующим образом:

- Виділення ячейки - встановите покажчик на лівий край осередку і натисніть кнопку миші
- Виділення рядка - встановите покажчик зліва від рядка і натисніть кнопку миші
- Виділення стовпця - встановите покажчик на верхній лінії сітки стовпця і натисніть кнопку миші
- Виділення декількох ячеек, рядків або стовпців - утримуючи кнопку миші, переміщайте покажчик по ячейке, рядку або стовпцю або виділіть один осередок, рядок або стовець, а потім утримуючи клавішу **{Shift}**, виділіть наступну ячейку, рядок або стовець
- Виділення тексту в наступній ячейці - натисніть клавішу **{Tab}**.
- Виділення тексту в попередній ячейці - натисніть комбінацію клавіш **{Shift + Tab}**.

3.1.6.3 Зміна таблиці.

Зміна ширини і висоти рядків і стовпців в таблиці:

1. Підвести покажчик миші до межі змінного стовпця (рядки) і добитися того, щоб покажчик придбав вид спеціальної двонаправленої стрілки.
2. Натиснути на ліву кнопку миші і перетягнути межу на потрібне місце.

Вставку рядків можна здійснити наступними способами:

Спосіб 1

1. Вивести курсор за праву межу рядка, після якого потрібно вставити рядок.
2. Натиснути клавішу **{Enter}**.

Спосіб 2

1. Встановити курсор в останню ячейку останнього рядка.
2. Натиснути клавішу **{Tab}**.

Спосіб 3

1. Встановити покажчик на лівий край рядка в таблиці, перед якою потрібно вставити рядок і виділити його. При такому ж положенні покажчика натиснути на праву кнопку миші. При цьому з'явиться контекстно-залежне меню.

2. Вибрати команду **Вставити рядки**.

Щоб видалити рядки з таблиці необхідно виконати наступні дії:

1. Виділити рядки, які потрібно видалити.
2. Виконати команду **Видалити рядки** у меню **Таблиця**.

Щоб вставити стовпці необхідно виконати наступні дії:

1. Виділити стовпець, перед яким потрібно вставити новий стовпець.
2. Не переміщаючи курсор, натиснути праву кнопку миші.
3. У контекстно-залежному меню, що з'явилося, вибрати функцію **Вставити стовпці**.

При необхідності об'єднати групу суміжних осередків з однієї або декількох суміжних рядків і стовпців в один осередок з розміщенням в ній вмісту цих осередків здійснюється таким чином:

1. Виділите одночасно всі об'єднувані осередки рядка.
2. Задайте команду **Об'єднати осередки** в меню **Таблиця**.

Якщо необхідно розбити осередок на декілька осередків, то необхідно виконати наступні дії:

1. Встановите курсор в ячейку, який хочете розбити.
2. Задайте команду **Розбити ячейки** в меню **Таблиця**.

3.1.6.4 Форматування таблиці

Існує два види форматування таблиці: автоматичне і ручне. Перший спосіб передбачає вибір стандартного формату, пропонованого Word. При другому способі користувач сам встановлює параметри форматування.

Розглянемо автоматичне форматування таблиці. Його можна здійснити, виконавши наступні дії:

1. Помістите курсор в створену таблицю.
2. Виконаєте команду **Автоформат** меню **Таблиці**. В результаті відкриється діалогове вікно **Автоформат таблиці** (див. рис. 3.1.6.2).

3. У списку **Формати**: виберіть варіант, найбільш відповідний по характеру оформлення таблиці.

4. У полі **Використовувати** встановите потрібні прапорці (межі, заливку, шрифт і колір) для завдання відповідного оформлення елементів таблиці.

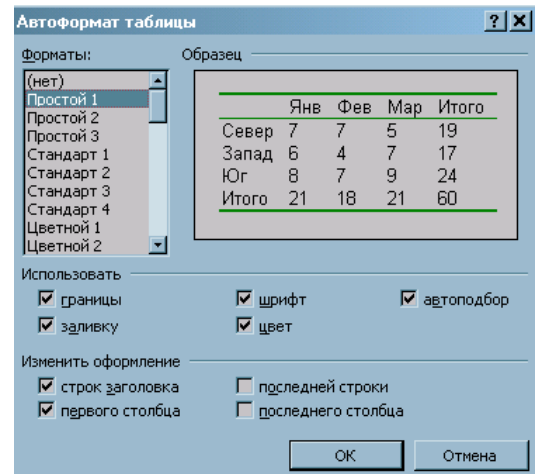



Рис. 3.1.6.2 Вікно Автоформат таблиці

5. Для того, щоб ширина стовпців таблиці автоматично підстроювалася під вміст елементів таблиці, встановимо прапорець **автопідбір**.

6. У полі **Змінити** оформлення встановимо прапорці рядків заголовка, першого стовпця, останнього рядка, відповідні тим елементам таблиці, для яких потрібно застосувати оформлення, що відрізняється.

7. Натисніть ОК.


Ручне форматування тексту можна здійснювати по наступних параметрах:

1. Обрамлення таблиці
 - Виділити таблицю.
 - Натиснути кнопку **Зовнішні межі**  на панелі **Форматування**.
 - Вибрати на ній кнопки **Зовнішні межі**, **Внутрішні межі** або **Всі межі**.
2. Вирівнювання вмісту ячеек
 - Виділити потрібні елементи таблиці.
 - Вибрати відповідну кнопку на панелі **Форматування**.



3. Зміна параметрів шрифту
4. Заповнення елементів таблиці (узори, колір і т.п.)
 - Виділіть потрібний рядок таблиці.
 - На панелі інструментів **Таблиці і межі** розкрийте список **Колір заливки** і зробіть відповідний вибір.
 - Виділіть частину таблиці з даними.
 - Виберіть команду **Формат, Межа і заливка** і встановите відповідні параметри заповнення і оформлення таблиці.

3.1.6.5. Написання формул і побудова діаграм.

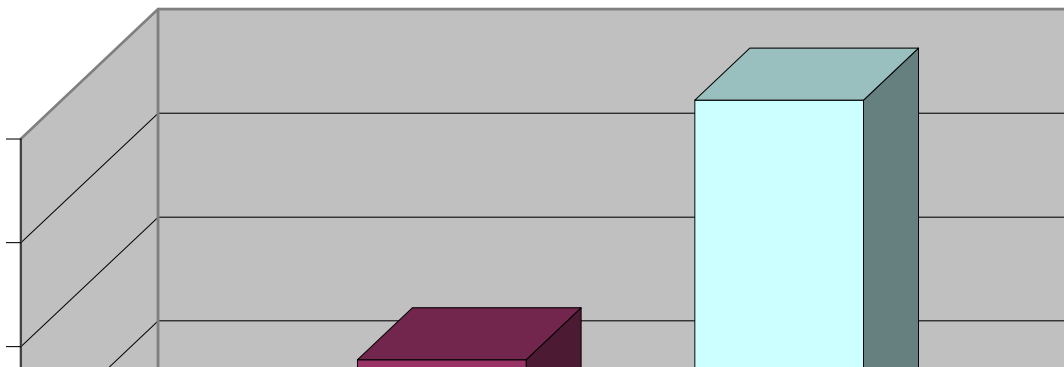
Дані таблиці можна відобразити у вигляді діаграми. Для цього виділяємо таблицю з назвами стовпців і з назвами відділів. Далі виконуємо команду **Вставка, Об'єкт, Діаграма Microsoft Graph 97** або викликаємо команду **Діаграма** в меню **Вставка**  **Рисунок**.

Наприклад, хай дана наступна таблиця і необхідно побудувати діаграму за її даними:

Таблиця об'єму продажів

Місяць	Об'єм
Січень	100
Лютий	150
Березень	95
Апрель	200
Май	125

Виділяємо область даних для нашої діаграми і вибираємо команду **Діаграма**. В результаті отримуємо діаграму наступного вигляду:



Зустрічаються такі випадки, коли в таблиці необхідно підрахувати деякий вираз і результат обчислення записати в осередок. Для виконання такої дії необхідно встановити

курсор в осередок, в який необхідно записати результат обчислення і скористатися командою Формула в меню Таблиця. В результаті з'явиться вікно Формула (див. рис. 3.1.6.3).

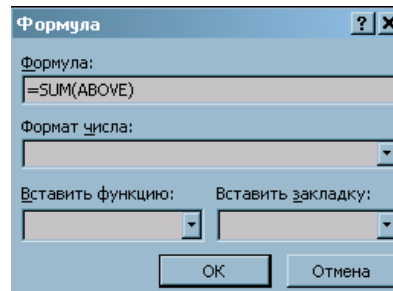


Рис. 3.1.6.3 Вікно Формула.

У полі **Вставити функцію** можна вибрати функцію для обчислення. У дужках цієї функції вказуються елементи таблиці над якими проводиться обчислення. Якщо для обчислення необхідні осередки, що стоять над поточною те в дужках можна вказати ABOVE, якщо зліва від поточної - LEFT, якщо справа - RIGHT, якщо під поточною - BELOW.

Якщо в списку наявних функцій потрібної функції немає, то в полі Формула можна самостійно написати свою формулу. Для цього необхідно пам'ятати, що стовпці таблиці нумеруються заголовними латинськими буквами: A, B, C, D, E, F і так далі, а строки нумеруються числами: 1, 2, 3, 4 і т.д. (див. рис. 3.1.6.4).

	A	B	C
1	A1	B1	C1
2	A2	B2	C2
3	A3	B3	C3

Рис. 3.1.6.4 Нумерація рядків і стовпців в таблиці.

Контрольні питання:

1. Які ви знаєте способи створення таблиць?
2. Як перетворити текст в таблицю?
3. Як перейти до першого осередку поточної стовпця?
4. Як виділити рядок і стовець?
5. Які ви знаєте способи вставки рядків?
6. Як об'єднати декілька ячеек?
7. Як розбити ячейку на декілька ячеек?
8. Що робить команда Автоформат?
9. Як побудувати діаграму за наявними даними в таблиці?

3.2 Текстовий редактор Open Office org Writer

3.2.1 Робота з документами

Головне меню текстового редактора Open Office org Writer (OOo 1.1.4) складається з 8 розділів: **Файл, Правка, Вид, Вставка, Формат, Сервис, Окно, Справка**, кожен з яких має свої функції і відкривається за допомогою миші або клавіші **Alt + підкреслена літера найменування відповідного розділу**.

Панель інструментів – це група піктограм – кнопок, за допомогою яких здійснюється більш швидкий та наочний вибір та виконання команд. Ввімкнути/вимкнути панель інструментів можна за допомогою команд **Вид → Панель инструментов → відповідна панель**. Як правило, на екран викликають дві панелі: **Стандартная** (панель функцій) та **Форматирование** (контекстна панель). Перша містить кнопки: створення документа, відкриття, зберігання, редагування документа, експорт в PDF, друк, дії з фрагментами тексту (при цьому фрагмент має бути виділеним): вирізання, копіювання, вставка, відміна останньої дії користувача, повторення останньої дії, додавання гіперпосилання і так далі. Панель форматування (контекстна) дозволяє: вибір стилю та типу написання (жирний, курсив, підкреслений, розмір, спосіб вирівнювання, вибір шрифтів (Times New Roman, Arial і так далі), колір шрифту, фон абзацу, маркери і так далі.

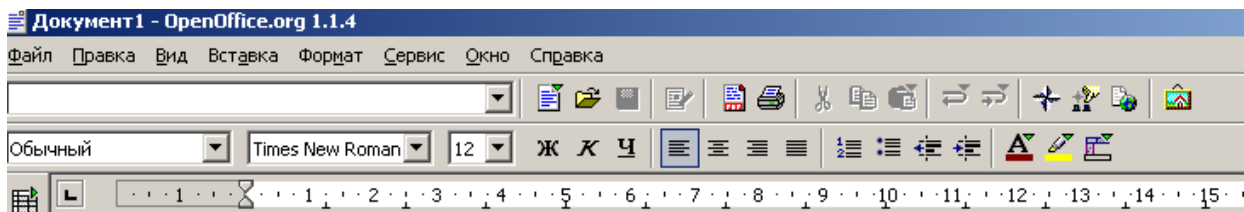



Рис. 3.2.1 Лінійка та піктограми стандартної панелі та панелі форматування

За допомогою команди **Вид → Линейка** встановлюється/вимикається лінійка, за допомогою якої можна швидко встановити відступи абзаців, розміри полів сторінки, розміри колонок на сторінках і в таблиці. В звичайному режимі виводиться тільки горизонтальна лінійка.

Робота з документами

Створення документа починається з команди **Файл → Создать → текстовый документ** або за допомогою відповідної піктограми стандартної панелі інструментів .

Під час створення документа можна використовувати **Файл → Мастер**, який надає можливість створювати листи, факси, порядок денний для зборів, презентацію і таке інше за допомогою встановлених шаблонів.

Зберігання поточного документа виконується командами **Файл → Сохранить** або **Файл → Сохранить как**. Перша команда зберігає файл з його початковим іменем (якщо воно є) або відкриває діалогове вікно для його введення, друга – під новим іменем. У залежності від того, який тип документа створювався, файл зберігається у відповідному форматі: текстовий документ .sxw, малюнок .sxd, презентація .sxi і так далі. Зберегти документ можна з паролем, можливість встановити який надається у діалоговому вікні зберігання документа, що запобігає несанкціонованому доступу до файлу. Якщо створений файл має велику цінність, то потрібно створити резервну копію (за допомогою архіватора) або зберегти під іншим ім'ям.

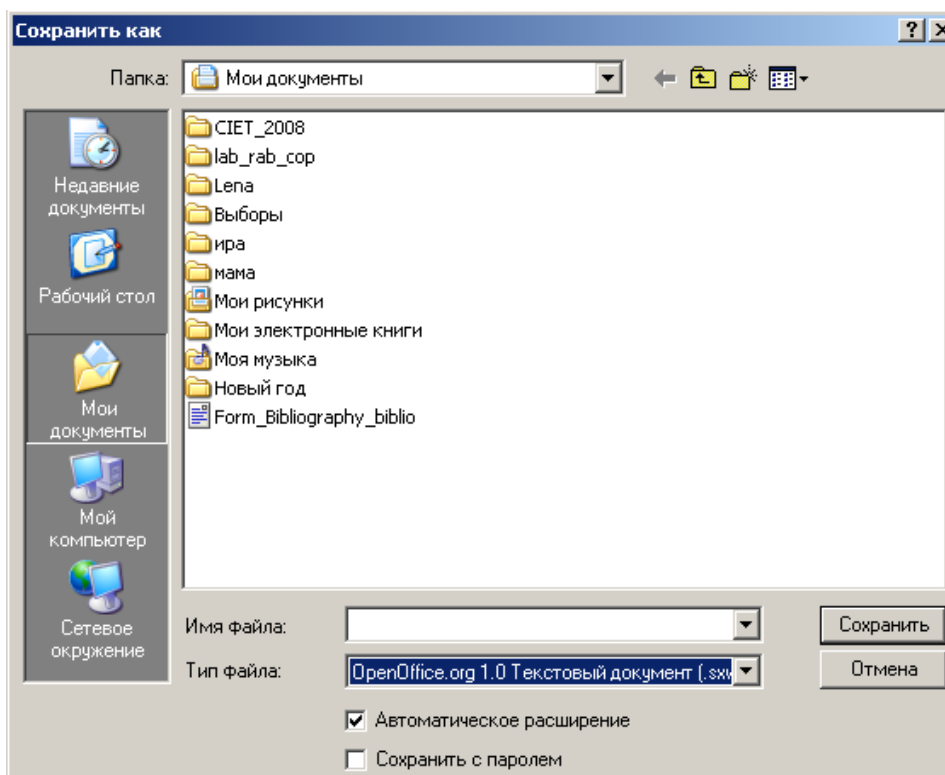



Рис. 3.2.2 Діалогове вікно збереження документа

Відкрити існуючий документ можна за допомогою команди **Файл → Открыть** або відповідної піктограми . З'являється діалогове вікно «Открыть», де потрібно вибрати потрібну папку та файл.

Місце введення тексту під час створення чи редагуванні документа визначається курсором. Вилучити символ можна за допомогою клавіші <Delete> (вилучається символ, що знаходиться зправа) або <Backspace> (вилучається символ, що знаходиться зліва). Після введення тексту може бути необхідним його форматування: корекція, видалення і переміщення слів і речень, абзаців і блоків, зміна параметрів шрифту (розмір, вирівнювання, написання і так далі).

Виділення тексту та його фрагментів.

Текст всього документа виділяється командою **Правка → Выделить все (Ctrl + A)**. Фрагмент тексту можна виділити різними способами:

1. курсор встановлюють на початок фрагменту, натиснувши ліву кнопку миші її вказівник перетягують на кінець фрагменту, потім клавішу відпускають;
2. курсор встановлюють на початок фрагменту, натискають і утримують клавішу <Shift>. За допомогою стрілок “→” “↓” курсор переміщують на кінець фрагменту. Всі клавіші відпускають.
3. курсор встановлюють на початок фрагменту, натискають F8 і за допомогою стрілок “→” “↓” обирають потрібний фрагмент тексту.

Для того, щоб виділити окреме слово потрібно двічі клацнути лівою клавішею миші на його середині, щоб виділити стрічку символів потрібно двічі клацнути мишею зліва від неї. Відмінити виділення – клацнути мишею на будь-якому вільному місці екрану.

Виділений фрагмент можна видалити, вирізати, вставити, перемістити, скопіювати за допомогою буфера обміну (Clipboard) , використовуючи кнопки **Вырезать**, **Копировать**, **Вставить** стандартної панелі інструментів , клавіші Ctrl + x, Ctrl + c, Ctrl + v або відповідні команди меню **Правка**.

У редактор Writer включено декілька таблиць символів, яких немає на клавіатурі. Вони складаються з математичних символів, знаків арифметичних та логічних операцій, дужок, стрілок, геометричних фігур, літер різних алфавітів і так далі.

Вставка символів виконується командою **Вставка** → **Спеціальний символ**. Відкривається вікно, у якому є вкладки **Шрифт** та **Набір символів**. Користуючись списками цих вкладок можна змінювати набір символів що відображаються в таблиці. Після вибору необхідного символу і натискання кнопки ОК, його буде вставлено в текст, або потрібно двічі клацнути на ньому лівою клавішею миші.

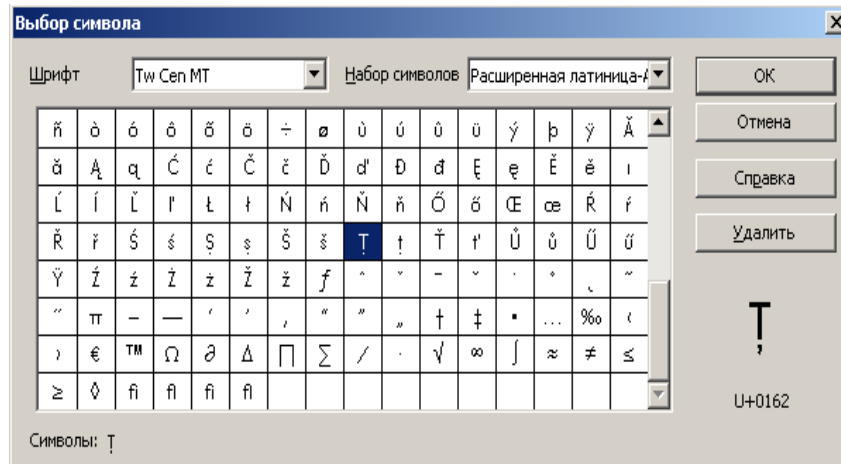


Рис. 3.2.3 Вікно вибору спеціальних символів

За допомогою математичних символів (потрібно встановити шрифт *Times New Roman*, а у вікні набору символів *Математические операторы*) можна вводити в текст нескладні математичні формули, а більш складні утворюються за допомогою **Вставка** → **Объект** → **Формула**.

Під час створення документу часто потрібно вводити в нього однакові фрагменти тексту. В цих випадках доцільно створити елемент автотексту за допомогою кнопки **Автотекст**

головної панелі інструментів, або команди **Правка** → **Автотекст**. У вікні, що з'являється, треба вказати власне авто текст та скорочення при наборі якого буде з'являтися автотекст.

У редакторі Writer можна створювати пошук слів і фрагментів тексту із наступною їх заміною. Ця процедура виконується командою **Правка** → **Найти и Заменить** яка активізує однойменне вікно. В поле **Найти** і в поле **Заменить на** вводяться відповідні слова, а потім при натисканні відповідних кнопок виконується пошук та (або) заміна. Вікно **Найти и Заменить** можна активувати також за допомогою кнопки **Поиск** головної панелі інструментів.

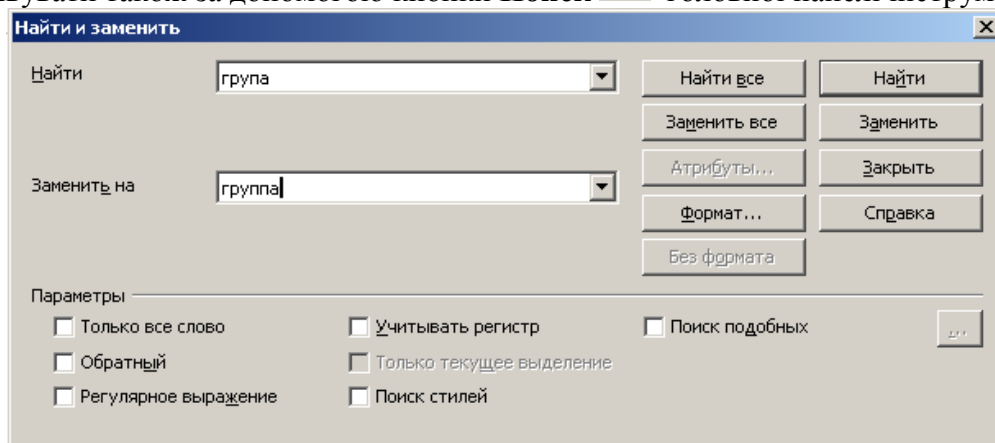



Рис. 3.2.4 Діалогове вікно для пошуку слів та фрагментів тексту

Переміщення у межах тексту виконується як за допомогою клавіші керування курсором, смуг прокручування, так і за допомогою кнопки **Навигатор**  (Команди **Правка** → **Навигатор** або клавіші **F5**). З'являється однойменне вікно, в якому можна вибрати об'єкт: заголовки, формули, таблиці, графічні об'єкти і так далі. Для того, щоб відбувся перехід на вибраний об'єкт, необхідно натиснути кнопку **Поиск вперед** або **Обратный поиск**.

КОМБІНАЦІЇ КЛАВІШ ДЛЯ ПЕРЕМІЩЕННЯ В МЕЖАХ ТЕКСТУ

→, ←	переміститися на позицію вправо, вліво
↑, ↓	вгору, вниз
Ctrl + ←	на одне слово вліво
Ctrl + →	на одне слово вправо
Ctrl + Home	на початок документа
Ctrl + End	на кінець документа
PgUp	на одну сторінку вгору
PgDn	на одну сторінку вниз
Home	на початок рядка
End	на кінець рядка

Питання для самоперевірки

1. Дайте поняття панелі інструментів як елемента вікна текстового редактора
2. За допомогою яких команд можна ввімкнути/вимкнути панель інструментів?
3. Які кнопки зазвичай має Стандартна панель інструментів ?
4. Які кнопки зазвичай має панель Форматування ?
5. Чим відрізняється команда «сохранить» від команди «сохранить как» ?
6. Чим відрізняється команда «Delete» від команди «Backspace» ?
7. Якою командою можна виділити увесь текст ?
8. Як можна виділити фрагмент тексту за допомогою миші ?
9. Як можна виділити фрагмент тексту за допомогою клавіш ?
10. Яку комбінацію клавіш використовують, щоб вирізати фрагмент тексту ?
11. Яку комбінацію клавіш використовують, щоб скопіювати фрагмент тексту у буфер обміну ?
12. Яку комбінацію клавіш використовують, щоб вставити у документ фрагмент тексту з буферу обміну ?
13. Які дії треба виконати, щоб виділити окреме слово ?
14. Які дії треба виконати, щоб виділити стрічку символів ?
15. Якими командами треба скористатися, щоб включити у документ символ якого немає на клавіатурі ?
16. За допомогою яких команд ведуть пошук заголовків, формул, таблиць, тощо ?
17. За допомогою яких команд ведуть пошук слів з метою їх заміни ?
18. Які треба виконати дії, щоб створити автотекст ?

3.2.2 Форматування тексту

Форматування символів включає: вибір типу, накреслення, розміру, кольору шрифту, встановлення інтервалів між символами, вибір засобу анімації (обрамлення, виділення) окремих слів і фрагментів тексту; введення в текст спеціальних символів і так далі.

Щоб відформатувати символи треба скористатися командою **Формат** → **Символы**, яка викликає вікно **Символы** із вкладками: **Шрифт**, де встановлюються тип шрифту (Times New Roman, Arial), накреслення (звичайне, курсив, напівжирне), розмір (висота символів), мова; **Эффекты шрифта**, де встановлюються такі ефекти, як написання літер (прописом, рядковим, маленькі, заглавні), рельєф (припіднятий, втоплений), анімація (контур, тінь, мерехтіння); **Положение**, де можна задати звичайне написання символа або написання у вигляді верхнього

індексу, нижнього індексу, задати обертання (0^0 , 90^0 , 270^0) та міжсимвольний інтервал (звичайний, розріджений, ущільнений) і т.д.; **Гиперссылка**, де можна задати посилання на об'єкти документа, що створюється або об'єкти іншого документа, для цього потрібно заповнити відповідні поля. На цій вкладці також можна створити макрос, що відповідає конкретній події; **Фон**, де встановлюється колір фону, на якому буде набрано текст. Більшість з описаних параметрів шрифту можна активізувати за допомогою кнопок панелі інструментів.

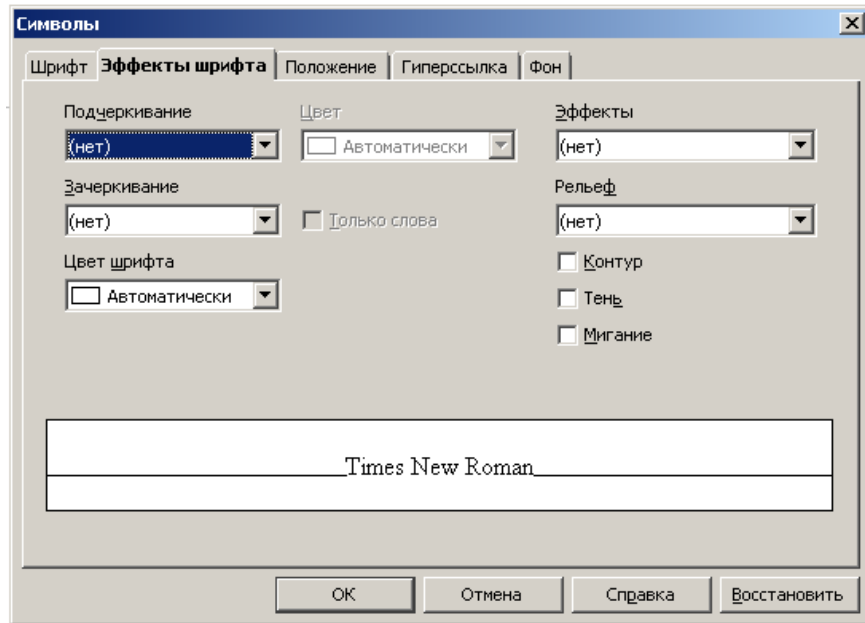


Рис. 3.2.5 Вікно форматування символів

Форматування абзацу дозволяє вирівнювати, задавати відступи, встановлювати інтервали між рядками та абзацами і таке інше. Дії з абзацами виконують за допомогою команди **Формат** → **Абзац**. З'являється вікно з іменем **Абзац** і з вкладками: **Отступы и Интервалы**, де можна задавати відступи зправа, зліва, відступ для першого рядка, міжрядковий інтервал (одинарний, полуторний, двійний і так далі), відступ перед та після абзацу; **Выравнивание**, де можна задавати вирівнювання по лівому краю, правому, центру, ширині (відповідні кнопки виведені на панель форматування); **Буквица**, де можна встановити символ (декілька символів) виділеного абзацу, а також висоту буквиці, стиль написання і так далі.

Під час заповнення сторінок абзаци, як правило, розриваються на частини. В разі необхідності такі розриви можливо вимкнути перемикачем **Не разрывать абзац** вкладки **Положение на странице**. За допомогою перемикача **Не отрывать от следующего** цієї ж вкладки два або декілька сусідніх абзаци обов'язково розміщуються на одній сторінці. Вкладка **Положение на странице** містить також перемикачі, що забороняють/дозволяють здійснювати перенесення слів. Вкладка **Нумерация** дозволяє вибрати стиль нумерації абзацив (1. або 1 і так далі), створити наскрізну нумерацію абзацив або почати нумерацію з конкретного абзацу і так далі. Для надання тексту більшої виразності його окремі абзаци і заголовки іноді вкладають у рамку з тінню та фоном. На вкладці **Обрамление** можна задавати межі обрамлення (||, |, □), стиль обрамлення, тобто товщину лінії (0,05pt, 1,20pt і так далі, де pt- кількість пунктів), відступи від обрамленого абзацу, колір ліній обрамлення, стиль і колір тіні. Фон абзацу задається на вкладці **Фон**.

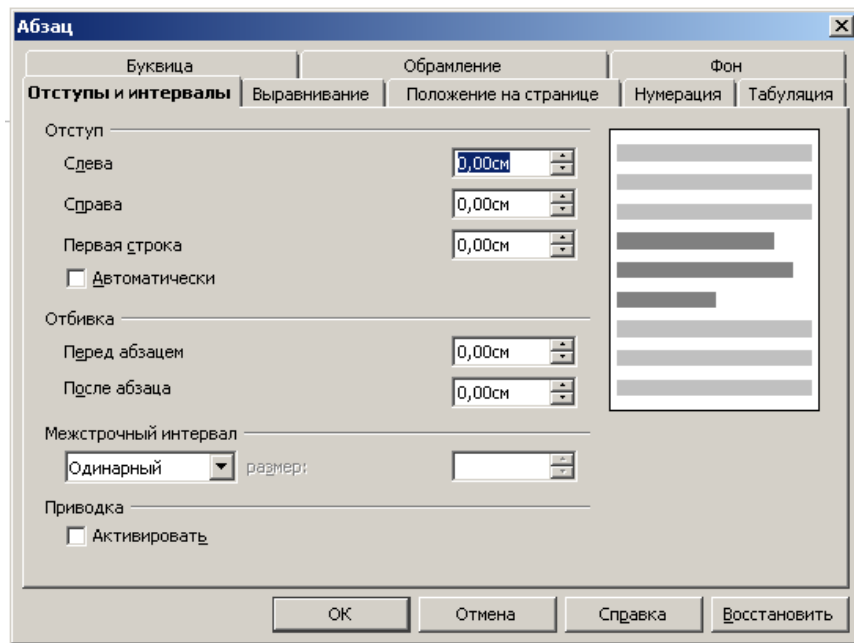


Рис. 3.2.6 Вікно форматування абзаців

Форматування сторінок документа

Процес форматування сторінок включає:

- встановлення розміру і орієнтації сторінки;
- встановлення параметрів полів;
- масштабування зображення сторінки;
- розбивку тексту документа на сторінки;
- нумерацію сторінок;
- створення верхнього та нижнього колонтитулів;
- розбивка сторінок на колонки і так далі.

Встановлення розміру та орієнтації сторінки виконується командою **Формат** → **Страница** → **Страница**, де також можна задати формат сторінки (A4, A5 і так далі) і встановити параметри полів (відступи тексту від країв), встановлення верхнього і нижнього колонтитулів можна виконати на однойменних вкладках. Масштабування зображення сторінки виконується командою **Вид** → **Масштаб** → обирається відповідна радіокнопка.

Розбивку тексту документа на сторінки можна виконати автоматично або вручну. В автоматичному режимі текст розбивається на стандартні сторінки з однаковою кількістю рядків. Наприклад, у випадку використання міжрядкового інтервалу «Одинарный», шрифту Times New Roman з розміром 14 і полями 2; 1.7; 2.5; 1.5 см кількість рядків буде складати 43-44.

Автоматична розбивка документа не дуже зручна, оскільки призводить до розривання ілюстрацій, появи вільних полів та інших дефектів, тому розбивку документів нерідко роблять вручну. Ручна розбивка починається із встановлення курсору на потрібне місце розриву і виконується командою **Вставка** → **Разрыв**, яка викликає вікно **Вставить разрыв**, за допомогою якого можна виділити новий рядок, стовпчик або сторінку.

За допомогою команди **Вставка** → **Поля** можна вставити в документ дату, час, відомості про автора, номери сторінок і так далі. *Поле* – це спеціальний код, який дозволяє вставляти інформацію із різних джерел (дата і час від годинника ПК і т.д.). За командою **Формат** → **Колонки** виводиться вікно **Столбцы**, за допомогою параметрів цього вікна можна розбити сторінку на стовпчики із вказуванням їх ширини, інтервалів між стовпчиками, роздільних ліній.

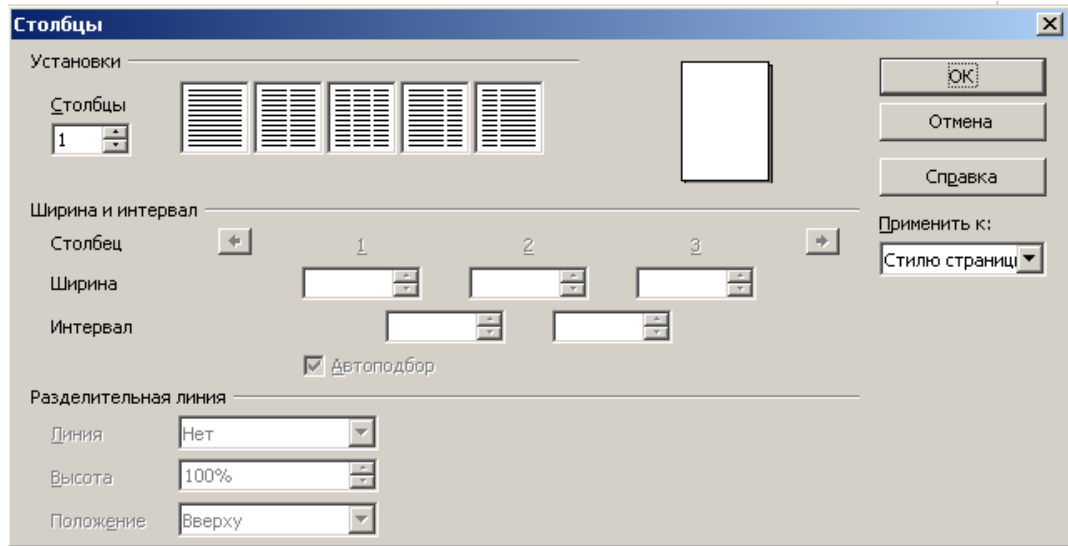


Рис. 3.2.7 Вікно створення стовпців

Питання для самоперевірки

1. За допомогою яких команд можна виконати форматування символів ?
2. За допомогою яких команд можна виконати форматування абзаців ?
3. Дайте поняття гіперпосилання.
4. За допомогою яких команд можна задати буквицю ?
5. Як зробити так, щоб абзаци не розривались під час форматування сторінки ?
6. За допомогою яких команд можна виконати форматування сторінки ?
7. З якою метою проводиться ручна розбивка документа ?
8. Яким чином можна зробити ручну розбивку документа ?
9. Дайте поняття поля, як об'єкта, що вставляється в документ.
10. Які треба виконати дії, щоб розбити сторінку на стовпці ?

3.2.3 Створення таблиць та діаграм

Таблицю можна створити за допомогою команди **Вставка → Таблица** (або кнопки **Вставить** головної панелі інструментів). З'являється вікно **Вставка таблицы**, за допомогою якого можна задати розмір таблиці (кількість рядків і стовпчиків), назву таблиці, оформлення. Після того, як таблиця буде відображена у документі, на екрані з'являється панель форматування таблиці, кнопки якої дозволяють вставляти/видаляти рядки і стовпчики, об'єднувати комірки, змінювати стиль оформлення, встановлювати колір фону.

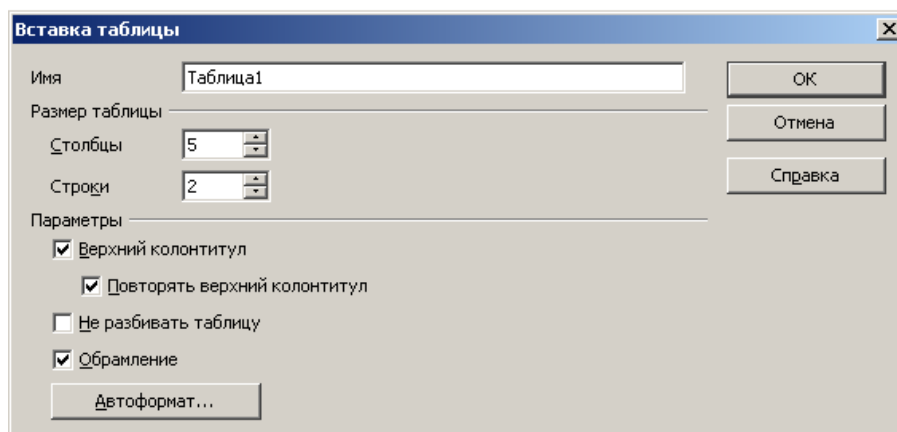


Рис. 3.2.8 Вікно створення таблиці



Рис. 3.2.9 Панель форматування таблиці

Ширину стовпчиків та рядків можна змінювати за допомогою миші, встановивши вказівник на межу двох стовпчиків (з'явиться маркер таблиці \leftrightarrow) і натиснувши ліву кнопку миші, перетягнути межу вліво/вправо; або викликати контекстне меню і використати команди **Столбец**, **Строка**; або командою **Формат** → **Строка**.

Розбити таблицю на декілька частин по горизонталі можна командою **Формат** → **Таблица**, на вкладці **На странице** ввімкнути перемикач **Разрыв**, те саме можна зробити за допомогою команд контекстного меню таблиці. Розбивка здійснюється, починаючи з позиції курсора. Для об'єднання таблиці використовується команда **Формат** → **Объединить таблицы**, видаливши усі абзаци між частинами таблиці, попередньо відобразивши недруковані символи.

Якщо таблиця має розмір більший ніж сторінка, то щоб повторювалися заголовки її стовпців на наступних сторінках, повинна бути ввімкнена кнопка **Повторить заголовок**, яка знаходиться на вкладці **На странице** вікна **Оформление таблицы**, яке можна викликати із розділа **Таблица** головного меню **Вставка** чи контекстного меню.

Як і в табличних процесорах сторінки і стовпці у OOo Writer позначаються цифрами 1, 2, ..., N, а стовпці буквами латинського алфавіту A, B, C і так далі. Наприклад, перша комірка має адресу A1, а діапазон комірок ідентифікується записами виду C2 : F4, B2 : D4 і так далі. Нумерація рядків і стовпців на екрані не відображується і приймається редактором за замовчуванням. Завдяки такому позначенню можна створювати формули у пустих клітинках таблиці, при цьому звертаючись до інших комірок за їхньою адресою. Створення формули починається зі знака =. Після того, як знак = введено до комірки, з'являється рядок формул з кнопкою **Формула**, у меню якої знаходяться елементарні математичні формули: сума, округлити, процент, корінь, ступінь; оператори: =, ≠, >, <; логічні оператори: і, або, ні; статистичні функції: середнє значення, мінімум, максимум; тригонометричні функції: синус, косинус, тангенс і обернені їм тригонометричні функції, за допомогою яких можна створювати досить складні формули. Наприклад, якщо комірка B2 містить число 200, а комірка C3 - число 210, то встановивши курсор у комірку D3 і набравши = (з'являється рядок формул) конструємо формулу: = <C3> - <B3>. Натискаємо кнопку Enter і у клітинці D3 з'являється число 10. Якщо в клітинці C4 знаходиться число 280 і треба одержати з нього корінь, а результат помістити у клітку D4, то у клітинці D4 набираємо =, потім із кнопки **Формула** вибираємо функцію **корінь**, з'являється функція sqrt, продовжуємо формулу, набравши <C4>, тобто в стрічці формул повинен бути вираз =sqrt (<C4>), натискаємо кнопку Enter і в клітинці D4 з'явиться число 16,73. Щоб адреса клітинки з'явилась в стрічці формул достатньо клацнути на клітинці лівою клавішею миші. Якщо треба обчислити суму чисел у стовпці, то треба встановити курсор у першу пусту клітку стовпця і натиснути кнопку Σ панелі інструментів.

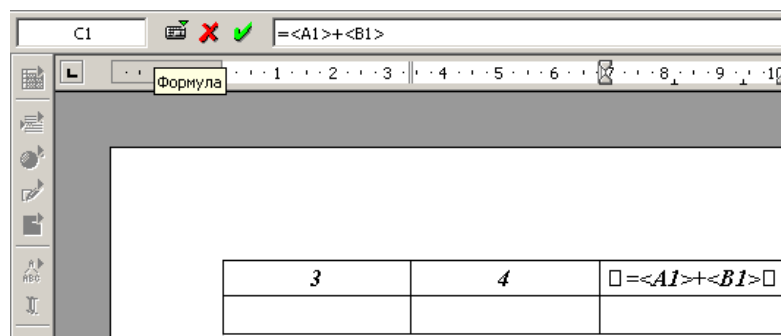


Рис. 3.2.10 Створення формул в клітинках таблиці

Для оформлення таблиці використовують вкладку **Фон** вікна **Оформление таблицы** або вкладку **Обрамление** цього ж вікна, де можна встановити тип лінії обрамлення, вид обрамлення. Лініям можна придати будь - який колір. Можна використовувати заливки комірок, рядків, всієї таблиці будь - якими кольорами. Якщо таблиця містить числову інформацію, то дані цієї таблиці можна оформити у вигляді діаграми. Щоб вставити відповідну таблиці діаграму, потрібно її виділити і виконати послідовність команд **Вставка** → **Объект** → **Диаграмма**. Буде запущено процес створення діаграми, який має декілька кроків, де потрібно вказати інформацію чи потрібно вводити перший рядок і стовпець у заголовок діаграми, тип діаграми і тому подібне. Одержану діаграму можна редагувати два рази клацнувши на її полі.

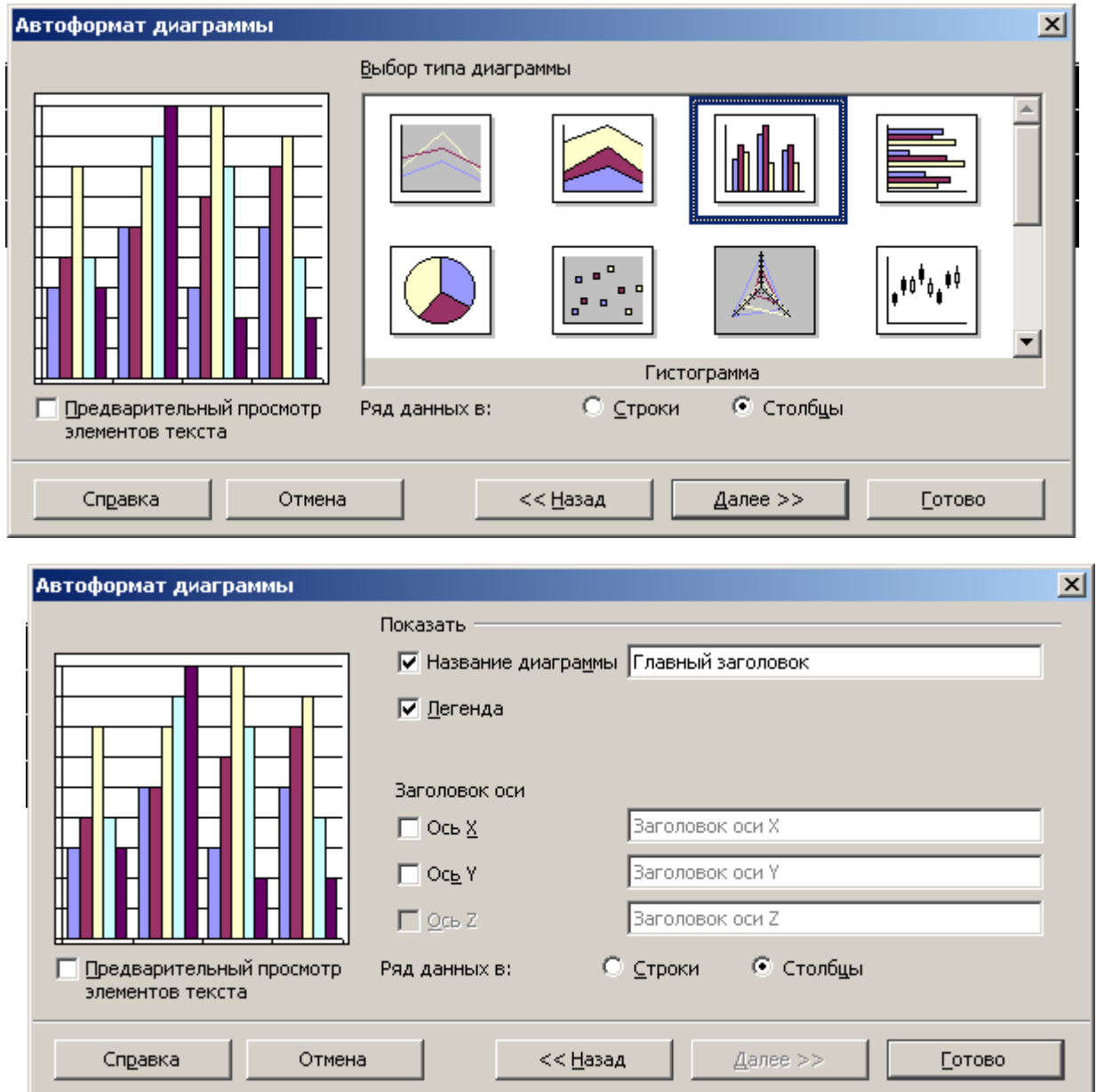


Рис. 3.2.11 Этапы створення діаграм

Питання для самоперевірки

1. Які є способи створення таблиці ?
2. Як створити формулу у клітинці таблиці ?
3. Як розбити таблицю на декілька частин ?
4. Як зробити так, щоб заголовки стовпців таблиці повторювалися на інших сторінках ?
5. Яку адресацію мають клітинки таблиці ?

6. Які стандартні функції можна використовувати при створенні формул у клітинках таблиці ? Де вони знаходяться ?
7. Як найпростіше вказати адресу клітинки при створенні формули ?
- 8 Які існують типи діаграм ?
9. Дайте поняття легенди. З якою метою вона створюється на діаграмі ?

3.2.4 Вставка об'єктів у документ

ООо Writer має вмонтовану галерею художніх зображень, які можна вставляти у документ. Зображення можна вставити по команді **Вставити**→ **Графический объект**→ **із файла** ... з'являється діалогове вікно зі списком зображень. Вставлене зображення можна редагувати: змінювати розмір, переміщати і т.д.

Елементарні фігури можна створювати за допомогою кнопки **Функции рисования** головної панелі інструментів.

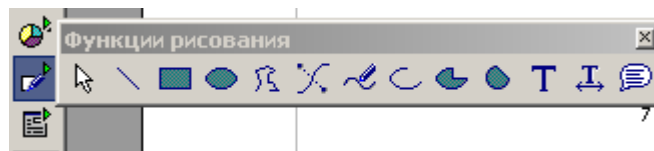



Рис. 3.2.12 Панель функцій малювання

При створенні навчальних методичних посібників, часто необхідно вставляти в документ діалогові вікна, піктограми. Для цього треба викликати на екран монітора потрібне діалогове вікно, наприклад, **Рассылка писем**. По команді **Alt + PrtScr** вікно заноситься у буфер обміну, потім потрібно активувати документ, і по команді **Вставити** (або **Вставить как ...**) з розділу **Правка** головного меню вікно буде встановлено у позицію курсора. Потім можна зробити вирівнювання об'єкта, прив'язку до абзацу чи сторінки, обтікання текстом і т.п.

Аналогічно у документ можна вставляти і об'єкти, сформовані безпосередньо у додатках **Open office** (діаграми, таблиці, фрагменти баз даних і т.п.)

Можна вставляти нескладні формули у документ командою: **Вставка**→**Объект**→**формула**. З'являється вікно редактора формул у якому конструємо потрібну формулу. На головній панелі інструментів знаходиться кнопка Σ , за допомогою якої можна викликати вікно з різними математичними символами. Після того, як формули створені, від редактора формул можна відмовитися кнопкою **Esc**. Знову викликати редактор формул можна двічі клацнувши лівою кнопкою миші по формулі.

Гіперпосилання

Може виникнути необхідність включити в документ інформацію, створену іншими додатками, не вставляючи його в головний документ. У цьому випадку використовується технологія зв'язування об'єкта. Гіперпосилання створюються так : **Вставка**→**Гиперссылка** або кнопкою , з'являється вікно **Гиперссылка**. Обираємо потрібну вкладку: інтернет, почта та новини чи документ. Вказуємо шлях до об'єкта, на який створюється посилання, і натискаємо кнопку **Применить**. У документ вставляється посилання з вказаним ім'ям. Щоб отримати доступ до об'єкта потрібно двічі клацнути на посилання.

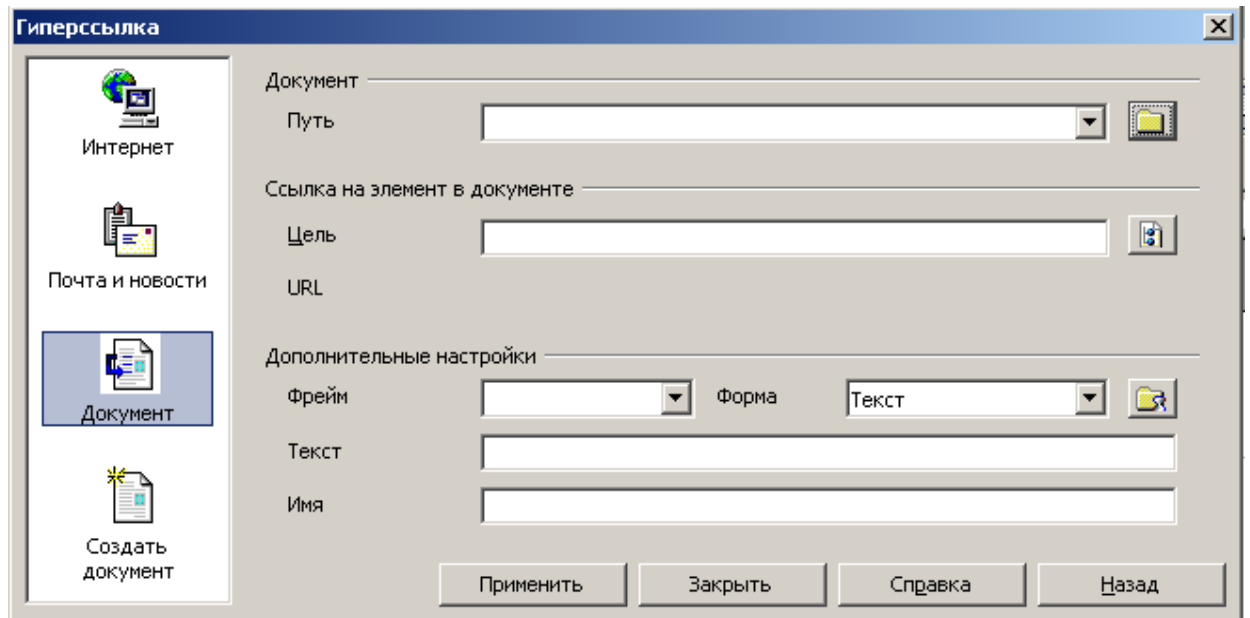


Рис. 3.2.13 Вікно створення гіперпосилання

Вставка макросів


Макрос – це послідовність команд, які можна виконати у будь - який момент. Наприклад, можна створити макрос для тих дій, які виконуються дуже часто. Щоб запустити режим створення макросу, потрібно виконати команди: **Сервис→Макросы→Записать макрос ...**, з'являється вікно **запись макроса/завершить запись**, яке показує нам, що почався запис макроса. Виконуємо дії, які хочемо оформити у вигляді макроса, наприклад, виділяємо фрагмент тексту і centruємо його, потім натискаємо клавішу **завершить запись**, з'являється діалогове вікно, де у полі **имя макроса** треба ввести назву макроса, вибрати режим, у якому будемо працювати з макросом – як з кнопкою, на панелі інструментів, як з пунктом меню чи як з комбінацією клавіш. Щоб назначити режим, треба скористуватися кнопкою **назначить**.

Питання для самоперевірки

1. Які треба виконати дії щоб діалогове вікно вставити в документ ?
2. Як створити формулу в документі ?
3. Які дії треба виконати, щоб відредагувати формулу, яка є в документі ?
4. Що таке гіперпосилання ?
5. З якою метою використовують гіперпосилання ?
6. Що таке макрос ?
7. Які дії треба виконати, щоб створити макрос ?

3.2.5 Призначення стилів і їх використання в документах

Стиль – це сукупність параметрів форматування різних об'єктів документа. Використання стилів дає змогу стандартизувати оформлення документів. Користувач може створювати, вилучати і модифікувати стилі, копіювати їх в шаблон і навпаки. Розрізняють встроєні стилі і стилі користувачів. Перші призначені для форматування стандартних складових документа, їх можна модифікувати, але вилучати не можна. Стили користувача створюються безпосередньо або модифікацією існуючих. Стили зберігаються у списку вікна **Каталог стилей**, які активізуються командою **Формат → Стили → Каталог...** У цьому вікні можна проглянути стилі абзацу, стилі символу, стилі врізки і т.п. Стильове оформлення документа

треба виконувати перед введенням тексту чи відразу після введення абзацу тексту. Під час введення чергового абзацу, йому присвоюється стиль попереднього абзацу. Новий стиль може створюватися за допомогою кнопок панелі форматування, кнопки  чи діалогового вікна, яке викликається командою **Формат** → **Стили** → **Каталог**. Обираємо категорію **стили користувача** і категорію стилів, наприклад, **стили абзаца** і натискаємо кнопку **Создать**. З'являється вікно **Стиль абзаца**, де можна задати відступи і інтервали, вирівнювання, шрифт, фон, обрамлення і т.п., вказуємо ім'я стилю. Після натискання кнопки ОК щойно створений стиль з'являється у списку стилів абзаців. Таким чином можна створити стилі символів, стилі сторінки і т.п. Будь-який стиль можна модифікувати, використовуючи кнопку **Изменить**, або вилучати за допомогою одноіменної кнопки.

Питання для самоперевірки

1. Дайте поняття стилю.
2. З якою метою використовують стилі ?
3. Які типи стилів Ви знаєте ? У чому їх відмінність ?
4. Де зберігаються стилі ?
5. Яким чином можна створити новий стиль ?

3.2.6 Засоби підготовки і представлення презентацій

Завдяки великому розповсюдженню мультимедійних комп'ютерів значно підвищилася зацікавленість до створення мультимедіа – додатків. Слово „мультимедіа” (мульти – багато, медіа – середовище, носій) означає сукупність середовищ чи носіїв, тобто сукупність різних способів подачі і збереження інформації (звука, графіки, анімації т.п.).

Мультимедіа – сучасна інформаційна технологія, яка забезпечує об'єднання графічних образів, відео, звука і других спеціальних ефектів за допомогою комп'ютерних засобів.

Для розробки мультимедійних додатків потрібен мультимедійний ПК на основі ОС Windows та відповідне програмне забезпечення. Розробкою і розповсюдженням стандартів і специфікацій на апаратне і програмне забезпечення мультимедійних комп'ютерів займається підрозділ Software Publishers Association Ради Multimedia PC Marketing Council (MPC). Пристрої та програми, які марковані торговою маркою MPC, відповідають вимогам для роботи по мультимедійним технологіям.

Авторські засоби мультимедіа – це засоби, які мають заделегіть підготовлені елементи для розробки інтерактивних мультимедіа програм. В основу різних класифікацій авторських засобів мультимедіа покладено авторську метафору – методологію, однак деякі авторські засоби мають рису декількох метафор. Саму повну їх класифікацію запропонував Джемі Сілгаром, по якій розрізняється 8 типів авторських засобів мультимедіа: мова сценаріїв; образотворче керування потоком даних; кадр; картка з мовою сценаріїв; тимчасова шкала; ієрархічні об'єкти; гіпермедіа – посилання; меркери.

Мова сценаріїв - це об'єктно - орієнтована мова програмування. У ній за допомогою спеціальних операторів визначається взаємодія елементів мультимедіа, розташування активних зон, кнопок і т.п. Однак редагування графічних зображень, відео і звуку у таких мовах відсутнє або дуже обмежене. До засобів, які базуються на мові сценаріїв, відносяться Ten Core Language (фірма Computer Teaching), Media View (фірма Microsoft).

Образотворче керування потоком даних - основою цього засобу є палітра значків (IconPalette), що включає різні функції взаємодії елементів засобу і напрямну лінію (FlowLine), яка показує зв'язки між значками. За допомогою значків забезпечується вивід на екран тексту, графічного зображення, їх переміщення, відтворення фільмів, звуків і т.п. Ці засоби мультимедіа достатньо дорогі, до них відносяться Authorware (фірма Macromedia), Icon Author (фірма Aim Tech), TIE (фірма Global Information Systems).

Кадр – авторські засоби цього типу подібні до типу образотворчого керування потоком даних. У них також включається палітра значків, але зв'язки між ними можуть відображати складні алгоритми, які розгалужуються. Ці засоби дуже швидкі, потребують автоматичної наладки, оскільки помилки візуально невловимі. До цих засобів належать Quest (фірма Allen Communication), CBT Express (фірма Aim Tech) і т.п.

Картка з мовою сценаріїв - дуже потужний мультимедійний засіб. Процес розробки картки нагадує процес роботи з книгою. Створюють сторінку за сторінкою, а потім використовують гіпертекстові можливості для навігації між сторінками і об'єктами в рамках сторінок. Авторські системи постачаються з великою кількістю шаблонів, прикладів і графічних елементів. Основним недоліком засобів цього типу є неможливість забезпечити точне керування синхронізацією і виконання паралельних процесів. До засобів типу картка відносяться Multimedia Tool Book (фірма Asymetrix), Power Point (фірма Microsoft).

Тимчасова шкала – засоби цього типу подібні за структурою інтерфейсу користувача до звукового редактора для багатоканального запису. Основними елементами є трупа (база даних об'єктів) і партитура (покадровий графік подій, які відбуваються з цими об'єктами). Кожна поява об'єкта із трупи в одному з каналів партитури називається спрайтом. Для керування спрайтами у залежності від дій користувача використовується об'єктно - подійна мова сценаріїв. Авторські засоби на основі тимчасової шкали використовуються для розробки складних комерційних додатків і комп'ютерних ігор з інтенсивною мультиплікацією. До авторських засобів цього типу належать Director (фірма Macromedia), Power Media (фірма RAD Technologies).

Ієрархічні об'єкти – у цьому засобі використовується метафора об'єкта, як і в об'єктно-орієнтованому програмуванні. Є можливість створювати складні конструкції з розвинутим сюжетом. Використовуються в основному професійними розробниками. До засобів цього типу належать New Media Studio (фірма Sybase).

Гіпермедіа – посилання - авторські засоби, що ґрунтуються на гіпермедіа – посиланнях подібні до засобів типу кадр. З їх допомогою можна розробляти гіпертекстові додатки з елементами мультимедіа. До засобів цього типу належать Hyper-Method (фірма Prog Systems AI Lab), Formula Graphic (фірма Harrow Media).

Маркери – в авторських засобах цього типу використовуються спеціальні команди – теги, які забезпечують взаємодію та об'єднання елементів мультимедіа. Їх в основному використовують для підготовки діалогових довідкових матеріалів, сторінок для вузлів Internet. Прикладами засобів типу маркер є Hot Dog (фірма Sausage Software), Front Page (фірма Vermeer).

Види і типи презентації

У залежності від способу презентації на ПК розрізняють такі їх види:

- із сценарієм;
- інтерактивна;
- автоматична;

Презентація із сценарієм – традиційна презентація із слайдами, додатковими засобами кольорової графіки і анімації з виводом відеоматеріалу на великий екран чи монітор. Є можливість вносити зміни під час демонстрування. Можна використовувати анімацію, графіки, ілюстрацію. Озвучування показу, як правило, веде сам ведучий.

Інтерактивна презентація – це діалог користувача с ПК. Користувач приймає рішення, який матеріал для нього важливий, і робить вибір потрібного об'єкту на екрані. Інтерективна презентація дає можливість робити пошук потрібної інформації, заглиблюючись у неї настільки, наскільки це було передбачено розробником презентації.

Автоматична презентація – це закінчений інформаційний продукт. Таку презентацію можливо перенести на відеоплівку, дискету, компакт-диск і розіслати потенційним

користувачам. У залежності від сфери застосування розрізняють торгові, маркетингові, навчальні і корпоративні типи презентацій.

Створення презентацій

Презентація являється одним з додатків, який можна створити у середовищі OpenOffice.org. Після вибору пункту **Презентация** программы **OpenOffice.org** на екрані з'являється діалогове вікно **Мастер презентаций**, у якому пропонується вибір одного з трьох режимів: порожня презентація, із шаблону, відкрити існуючу презентацію.

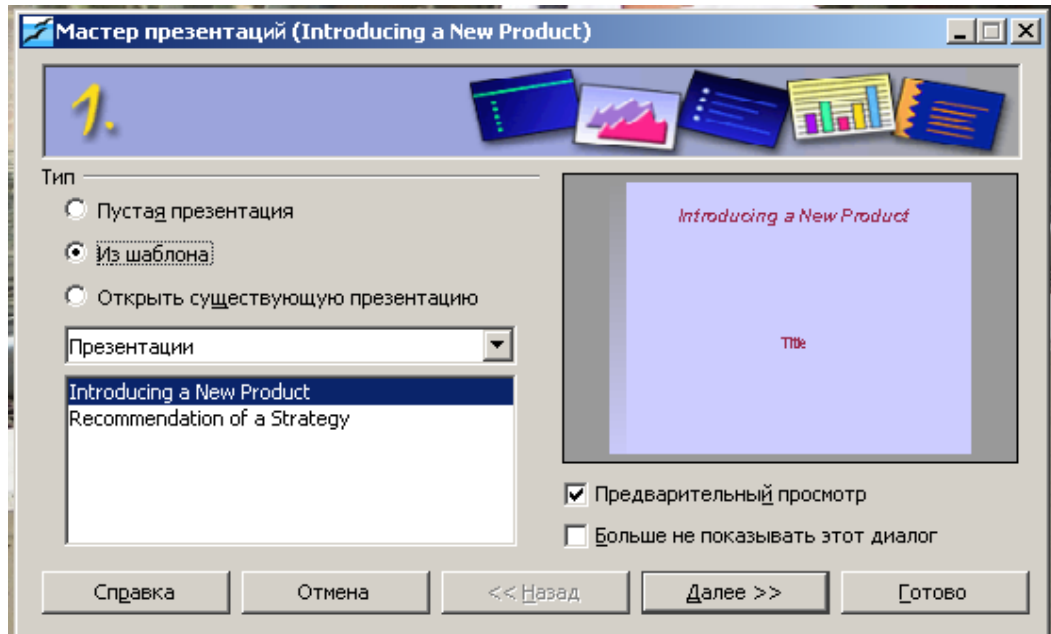


Рис. 3.2.14 Вікно вибору типу презентацій

Розглянемо створення презентацій в режимі **Из шаблона**. Після вибору режиму **Из шаблона** пропонується вибір виду презентації із пропонованого списку (службові, проекти, ділові і т.п.). На другому етапі можна обрати стиль презентації і способи її відображення: на екрані, плівці, папері, слайд і т.п.

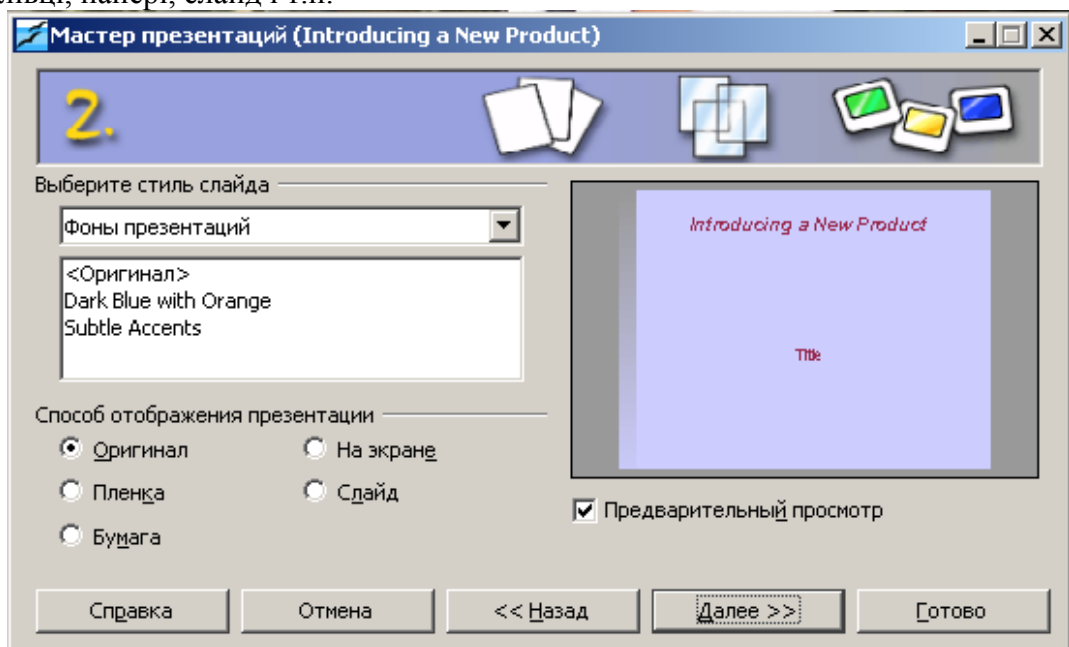


Рис. 3.2.15 Другий етап створення презентацій

Після вибору кнопки **Далее** на третьому етапі можна задати ефекти при переході від одного слайду до іншого під час демонстрації: зникнення слайду всередину, спливання (знизу, зверху) і т.п.; швидкість появи презентації; якщо обрана кнопка **автоматически**, то можна вказати час затримки сторінок під час перегляду.

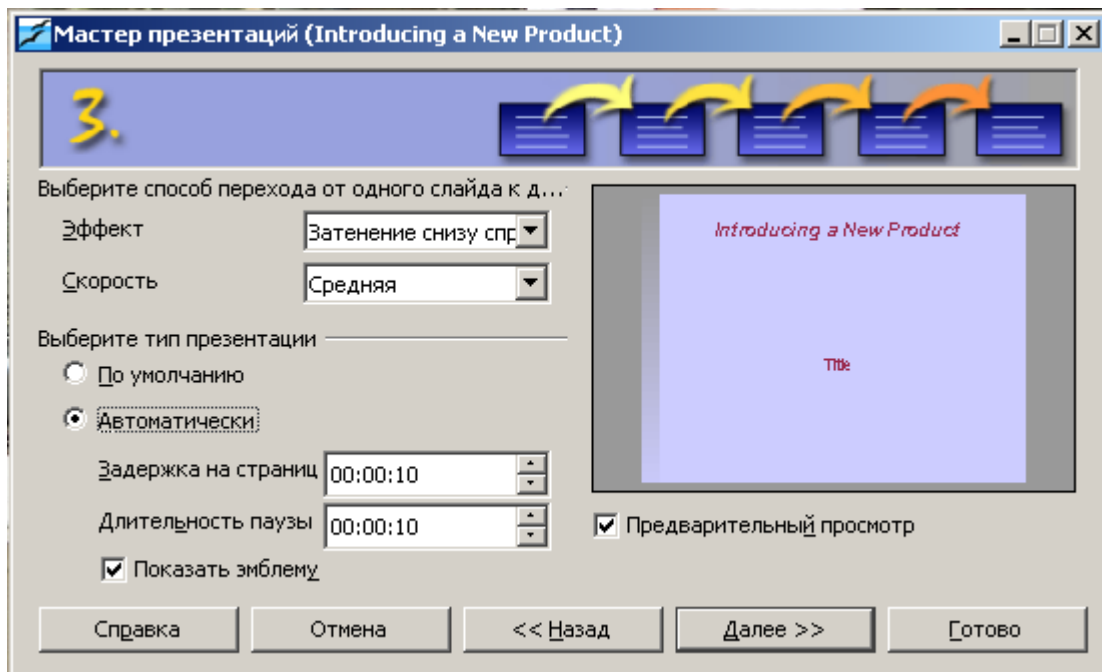


Рис. 3.2.16 Третій етап створення презентації

На четвертому етапі можна задати назву організації, тему презентації, цілі і т.п. (формується заголовна сторінка (слайд)). На п'ятому етапі пропонується вибрати сторінки, які будуть включені у презентацію. Наприклад, якщо була вибрана презентація типу **Представление новой продукции**, то можуть бути запропоновані сторінки: запит покупців (пропонується описати пропозиції покупців і їх вимоги), аналіз цін (пропонується описати фінансову вигідність покупки для покупців, порівняння якості продукції і цін у відношенні до інших фірм, які виробляють подібну продукцію) і т.п.

У іншому засобі створення презентації, такому як, наприклад, Power Point Microsoft Office, після вибору виду презентації, пропонується вибір її підвиду. Наприклад, у виді **Служебные** є підвиди **Общее собрание**, **Финансовый отчет** і т.п., де також можна обрати сторінки, які потрібно включити у презентацію.

При виборі режиму **Пустая презентация** четвертий і п'ятий етапи випускаються (немає шаблону) – відкривається редактор презентації. У середовищі редактора можна корегувати уже існуючі сторінки, їх вилучати або додавати (вставляти нові). Для сторінки можна обрати чи змінити розмітку сторінки тобто склад сторінкозаповнювачів(контейнерів), наприклад, сторінка може представляти собою пустий слайд чи два сторінкозаповнювача - для заголовку та для тексту, або три контейнера для заголовка, для діаграми та тексту.

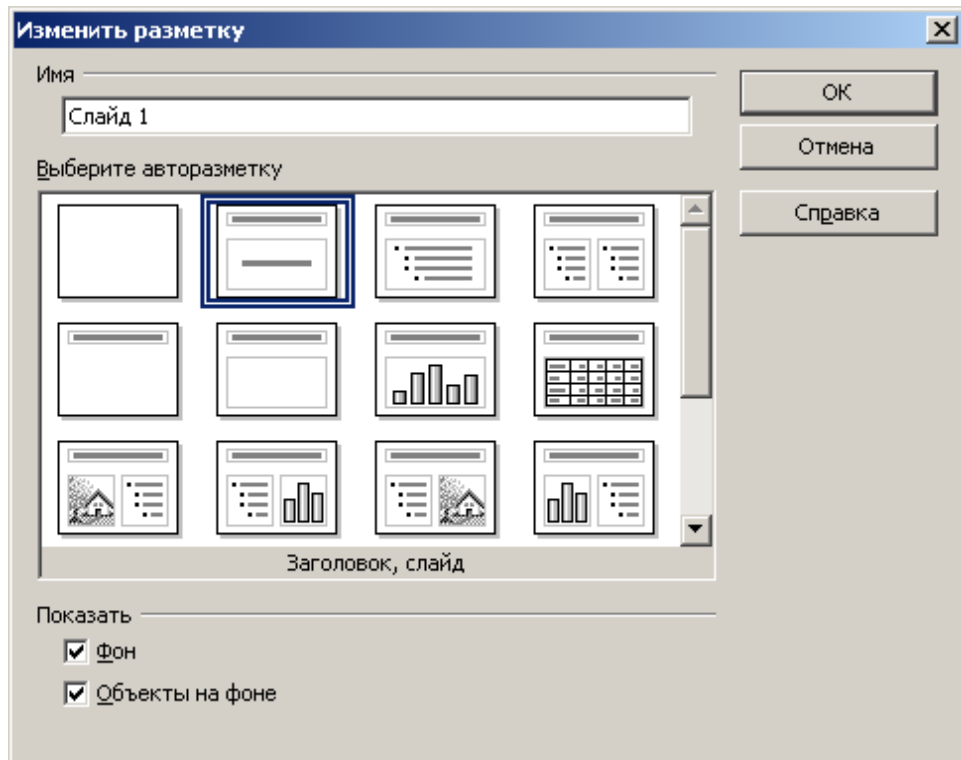


Рис. 3.2.17 Вікно для вибору розмітки сторінки

Якщо, наприклад, обрана розмітка з контейнером для зображення, то подвійним клацанням лівою кнопкою миші на контейнері для зображення ми одержуємо доступ до галереї зображень. Обране зображення вставляється в контейнер. За допомогою контекстного меню об'єкта, його можна редагувати. Текст можна вводити вручну або використовувати текст із документу, використовуючи буфер обміну, при цьому контейнер тексту повинен бути виділеним. Потім текст може бути форматований з приданням йому будь - яких ефектів(подібно до методів роботи у текстовому редакторі).



Рис. 3.2.18 Приклад розмітки сторінки

Використання гіперпосилань

Створені слайди можна продивлятися під час демонстрації один за одним (на екрані присутня панель з створеними слайдами) чи за бажанням, тобто під час демонстрації треба звертатися до конкретного слайду, який не знаходиться за поточним і не передує йому. Для цього потрібно організувати розгалужений і циклічний способи перегляду презентацій. Це можна зробити за допомогою гіперпосилання. На слайді гіперпосилання можна представити текстом чи кнопкою. Клацання мишею на гіперпосиланні приводить до переходу до того слайду, який знаходиться за адресою, яка вказана у гіперпосиланні. Для створення гіперпосилання треба вибрати потрібний контейнер і позицію курсора, потім використати команду **Вставка→Гиперссылка**, з'являється одноіменне вікно, обираємо кнопку **Ссылка на элемент в документе**, яка знаходиться біля одноіменного поля – з'являється список усіх створених слайдів у презентації – вибираємо потрібний. Назва вибраного слайду відображається у полі **Цель**. Потім вибираємо представлення гіперпосилання: кнопка чи текст, задаємо ім'я об'єкту і текст, який повинен представляти гіперпосилання. Натискаємо кнопку **Применить**.

Підготовка до демонстрації

Підготовку до демонстрації можна виконати за допомогою пункту **Демонстрация** головного меню, а саме за допомогою режимів **Настройки демонстрации** чи **Обычная демонстрация**. Якщо обираємо режим **Обычная демонстрация**, тоді в одноіменному вікні обираємо кнопку **Создать** – з'являється список створених слайдів, з яких можна обрати потрібні для демонстрації і запускаємо демонстрацію командою **Запустить**. Якщо ж обраний режим **Настройка демонстрации**, тоді у одноіменному вікні, що з'явилося, можна вказати, з якого слайда розпочати. Можна задати автоматичну демонстрацію слайдів і тоді треба вказати час – проміжок між показами двох слайдів чи вручну, показувати чи ні показник миші і у якому вигляді (наприклад у вигляді олівця) і т.п.

Щоб розпочати демонстрацію слайдів, можна обрати один із способів:

1. Запуск із редактора презентацій – клавіша F9. Підказку клавіші можна подивитися по команді **Демонстрация → Демонстрация**.
2. З робочого столу – для цього потрібно створити ярлик презентації.
3. Запуск файлу демонстрації слайдів – для цього потрібно створити такий файл командою **Файл → Сохранить как...** У списку **Тип файла** обрати Презентація і клацнути на кнопці **Сохранить**, перед цим вказавши ім'я файлу.

Під час демонстрації у автоматичному режимі досить клацнути мишею на слайді для переходу до наступного. Якщо на слайді є гіперпосилання, то досить клацнути на такій кнопці, щоб перейти до заданого слайду. Якщо курсор миші був заданий у вигляді олівця (пера), то під час демонстрації можна підкреслювати важливі пункти тексту.

Питання для самоперевірки

1. Які типи авторських засобів мультимедіа Ви знаєте ?
2. Які існують види презентацій за способом їх подання ?
3. В чому особливість створення презентацій за допомогою шаблона ?
4. Дайте поняття розмітки слайда.
5. Яке стильове оформлення можна застосувати до слайда ?
6. З якою метою використовують гіперпосилання при створенні презентації ?
7. Які режими демонстрації презентацій ви знаєте ? Дайте характеристику кожному режиму.

4. ЕЛЕКТРОННІ ТАБЛИЦІ (MS Excel та OO Calc)

Електронна таблиця (ЕТ) — це програма, призначена для введення та опрацювання даних, наведених у вигляді таблиці. З її допомогою можна виконувати складні обчислення з великими масивами чисел, будувати графіки та діаграми та використовувати для автоматизації математичних, інженерних, економічних чи статистичних обчислень.

Існують різновиди програм ЕТ: самостійні та програми, які входять до складу прикладного програмного забезпечення, таких як *Microsoft Office*, *Open Office.org* (програми *Excel*, *Calc*). Розглянемо роботу з ЕТ на прикладі офісних програм. Програми *Excel* і *Calc* мало чим відрізняються одна від одної за структурою та діями, які виконуються над даними, також мають практично однаковий зовнішній вигляд, однаковий набір команд. Але є деякі різниці: так у *Calc* інструмент **Анализ данных** має обмежені можливості, декілько відрізняються вікна **Мастеров функций** и **диаграмм**. Розглянемо роботу у програмі *Excel* і наведемо особливості у роботі з програмою *Calc*.

4.1. Загальні положення

Завантаження програми ЕТ. Стандартний запуск здійснюється з головного меню

Пуск → Программы → Microsoft Office → MS Excel (*OpenOffice.org → OpenOffice.org Calc*), або з допомогою кнопок відповідної офісної панелі або ярликів на **Рабочем столе**.

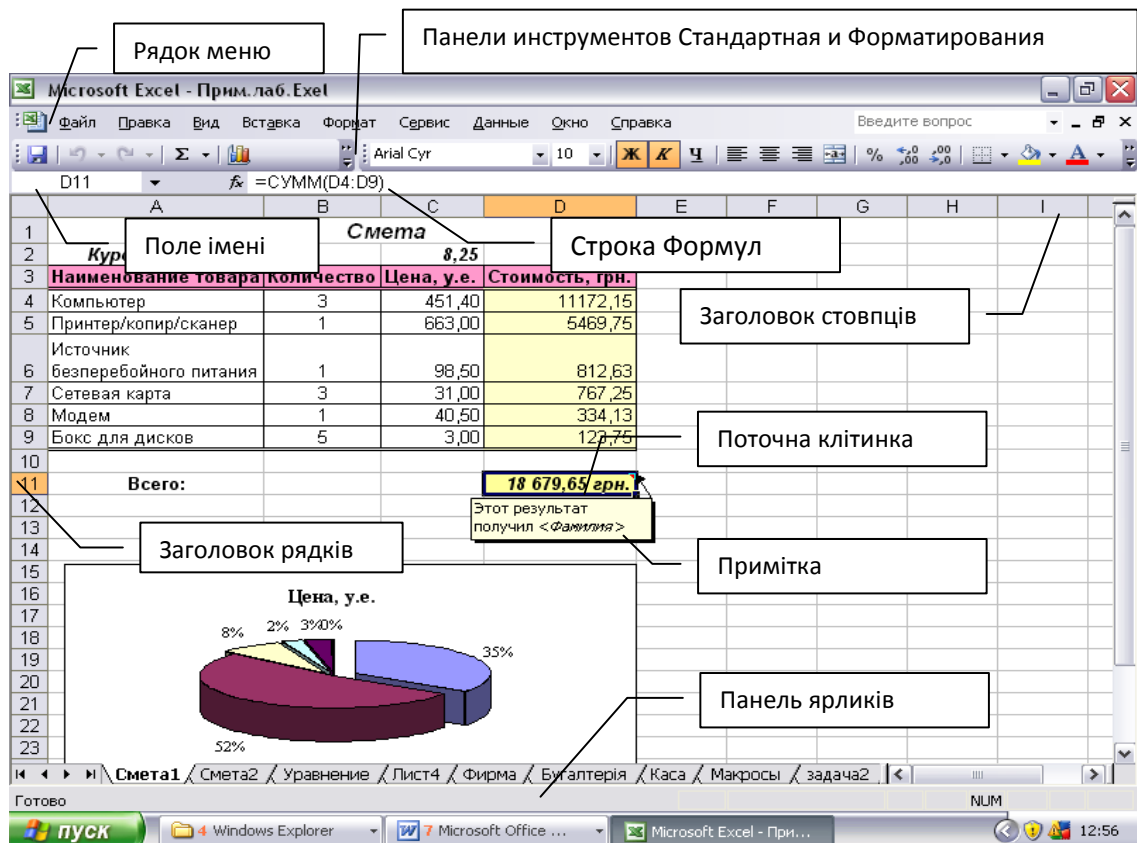
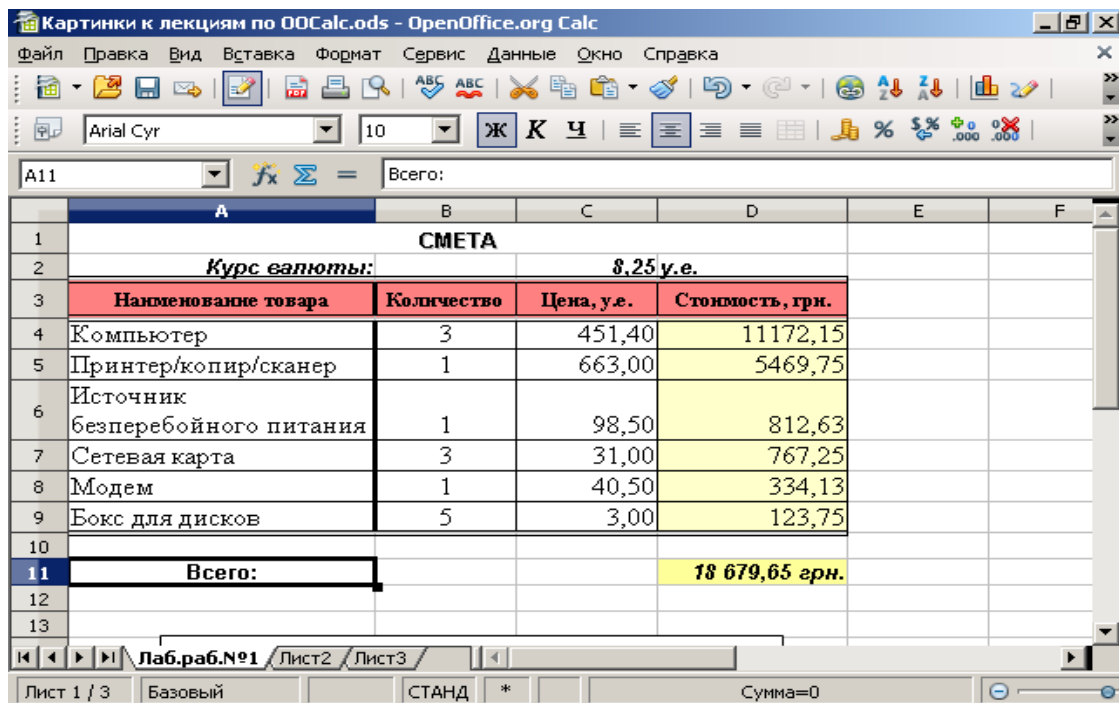


Рис.4.1. Вікно програми Ms Excel.

Структура ЕТ. ЕТ складається з клітинок, що утворюють *рядки* і *стовпці* (Рис.4.1, 4.2).

Стовпці таблиці позначені буквами (А, В, С, ..., Z, AA, AB, ..., AZ, BA, ...), а рядки — цифрами (1, 2, ...). Кожна клітинка має ім'я, яке складається з літери стовпця і номера рядка, на перетині яких розташована клітинка, наприклад А1. Стовпців може бути до 256, а рядків — до 65536 (Calc - 32000).

Клітинки утворюють таблицю, яка міститься на сторінці і має назву за умовчанням *Лист 1*, *Лист 2* тощо, назва якою відображена на бірці внизу екрана. За умовчанням їх має бути три або може бути збільшено до 255. Сторінки можуть бути декількох типів: саме таблиць, діаграмм, програмного коду VBA (*Visual Basic for Application*) або окно діалога. Ім'я листа можна змінювати, але воно не може містити символів "\", "/", "*", ":", "?", "!", "[", "]" і його довжина не повинна перевищувати 31 символа.



	А	В	С	Д	Е	F
1	СМЕТА					
2	Курс валюты:		\$,25 у.е.			
3	Наименование товара	Количество	Цена, у.е.	Стоимость, грн.		
4	Компьютер	3	451,40	11172,15		
5	Принтер/копир/сканер	1	663,00	5469,75		
6	Источник бесперебойного питания	1	98,50	812,63		
7	Сетевая карта	3	31,00	767,25		
8	Модем	1	40,50	334,13		
9	Бокс для дисков	5	3,00	123,75		
10						
11	Всего:			18 679,65 грн.		
12						
13						

Рис.4.2. Вікно програми Ooo Calc.

Декілька сторінок утворюють *книгу*, яка має назву *Книга 1* (Calc - Без имени 1). Книга зберігається у файлі з розширенням *.xls* (Calc - *.sxc*).

Адресація. В ЕТ у формулах та функціях для посилання на клітинки і блоки клітинок використовують імена клітинок як *адреси*. Існує три види адресів: *відносні*, *абсолютні* та *змішені*. *Відносні* адреси змінюються при копіюванні формул і мають вигляд просто адреси - *A1*, якщо посилання на клітинку не повинно змінюватися при копіюванні, то використовують *абсолютну адресу* клітинки, наприклад, *\$A\$1*, тобто додається символу "\$" перед позначенням стовпця і номером рядка. Іноді використовують *змішені адреси*, коли постійною є одна частина адреси, наприклад *A\$1* або *\$A1*. При редагуванні адреси у формулі циклічну зміну вигляду посилання здійснюють за допомогою клавіші **F4**.

Адреса прямокутного блоку (діапазону) клітинок, сумісного, складається з адреси лівої верхньої клітинки блоку, двокрапки (оператора діапазону) і адреси правої нижньої клітинки, наприклад *A1:E5*. Якщо треба описати не сумісний діапазон клітинок, то використовують «;», наприклад, *A1;B3;E5*.

Адреси використовують також *повні імена (адреси) клітинок*. Для зазначення повного імені клітинки до її адреси дописують ім'я листа, де вона розташована, і назву книги (файлу): **[Книга 1]Лист1!A1**.

Питання для самоперевірки

1. Що таке ЕТ і яке її призначення?
2. Яка структура ЕТ?
3. Які розширення мають файли ЕТ?
4. Як позначаються стовпці та рядки у ЕТ?
5. Як записують адресу клітинок?
6. Як записують адресу діапазону клітинок?
7. Які типи посилань використовують в ЕТ?
8. Коли використовують абсолютну адресацію клітинок?
9. Для чого використовують імена клітинок?
10. Як записують повну адресу клітинки?

4.2. Типи даних, введення та форматування даних у ЕТ

Типи даних. У клітинки користувач вводить дані трьох основних типів: *числа*, *тексти*, а також *формули* для виконання дій з даними.

Текстові дані використовують, зокрема, для оформлення назв таблиць і назв рядків, а також стовпців даних, вони можуть містити будь-які символи.

Числові дані використовуються для введення числових значень у різному відображенні, для цього використовують цифри від 0 до 9 та спеціальні символи "+", "-", "E", "e", "(", ")", ".", ",", "%", "/" і позначення грошових одиниць.

Формули призначені для виконання дій над вмістом клітинок (над даними) згідно з умовою конкретної задачі. Усі формули починаються символом "=". У формулах можна використовувати такі операції: додавання "+", віднімання "-", множення "*", ділення "/", піднесення до степеня "^". Операндами у формулі можуть бути числа, адреси клітинок, функції або заголовки стовпців (рядків) таблиці. Для зміни стандартного порядку виконання операції застосовують круглі дужки.

За замовчуванням після введення формули у клітинці відображається результат обчислень, а формулу можна побачити лише у **Строке Формул**.

Щоб побачити всі формули у таблиці, треба задати режим відображення формул у клітинках **Сервіс → Параметри... → вкладка Вид → опція Параметри окна → Формули (→ OpenOffice.org Calc → Вид → Формули)**.

Щоб знову побачити результати обчислень, потрібно вимкнути режим відображення формул (Рис.4.3).

Введення даних. Щоб виконати якусь дію над клітинкою чи її даним, клітинку потрібно виокремити (вибрати, активізувати). Це роблять за допомогою клавіш зі стрілками або миші. Активна (виокремлена) клітинка має рамку з *маркером*, який є у правому нижньому куті. З нею можна виконувати дії, визначені в головному чи контекстному меню: ввести чи вилучити дане, скопіювати чи перемістити дане в буфер обміну, очистити клітинку, відформатувати дане чи клітинку, вставити примітку тощо. Виокремлювати можна не лише одну, але й декілька клітинок (рядків чи стовпців).

Щоб увести в клітинку дане, її виокремлюють, набирають дане на клавіатурі та натискають на клавішу вводу **Enter** або на клавішу клавішу **Tab** або кнопку **Ввод** у **Строка формул**.

Дані активної клітинки та її адреса відображаються у **Строка формул**. Під час введення дане можна редагувати. Уведений у клітинку текст (до 255 символів) автоматично вирівнюється до лівого краю, а числа — до правого. Якщо почати вводити нове дане у клітинку, то старе пропадає.

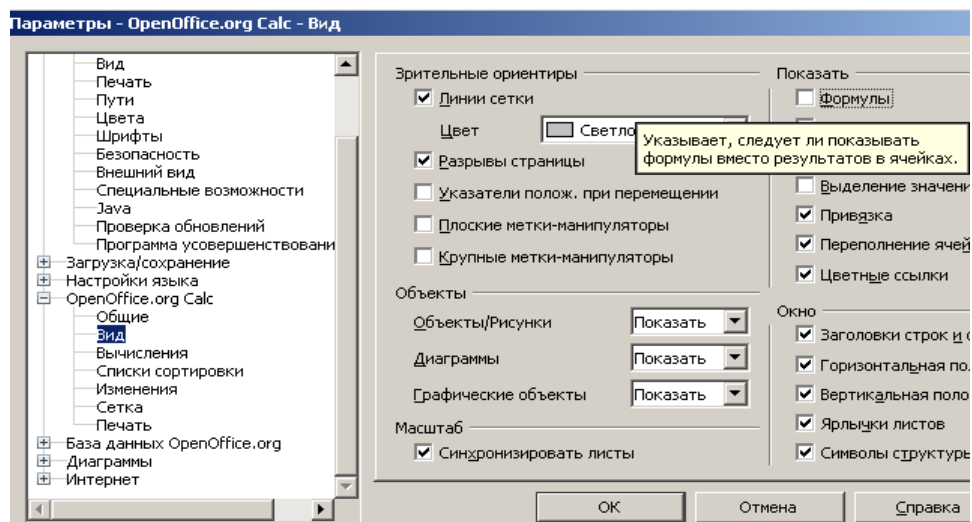
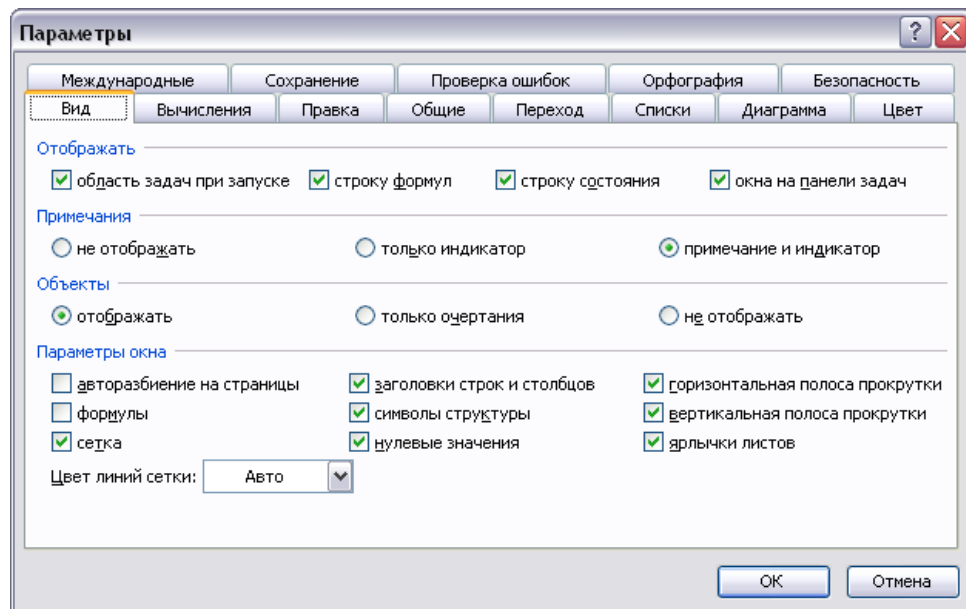


Рис.4.3. Вікно команди Параметри програм Excel та Calc.

Якщо розмір тексту перевищує розмір клітинки, то для його відображення використовують клітинки, розміщені праворуч. Зберігається текст тільки в одній клітинці. Його відображення в сусідніх клітинках зникає, якщо в ці клітинки вводяться дані. Для відображення в одній клітинці кількох рядків тексту виконайте команду **Формат → Ячейки... → вкладка Выравнивание → переносит по словам**. Висота рядка збільшиться і текст розміститься на додаткових рядках всередині клітинки.

Якщо в клітинці вже є дані і його треба відредагувати, то клітинку вибирають і користуються одним із трьох способів:

- двічі клацають мишею;
- натискають на клавішу **F2**;
- застосовують рядок формул.

Вилучити з клітинки дані, примітку, формат даного можна **Правка → Удалить содержимое... → Удалить все**.

Діапазон клітинок можна заповнювати рядками, натискаючи клавішу *Tab* для переходу в клітинку праворуч, або стовпцями, натискаючи клавішу *Enter* для переходу в клітинку, що розміщується нижче заповненої.

Форматування даних. Числа в клітинку вводять звичайним способом, але вони можуть бути відображені як заокруглені, із символом грошової одиниці (\$, грн), з комами чи пропусками, які відокремлюють тріади цифр тощо. Відображення даного залежить від формату його зображення (Рис.4.4).

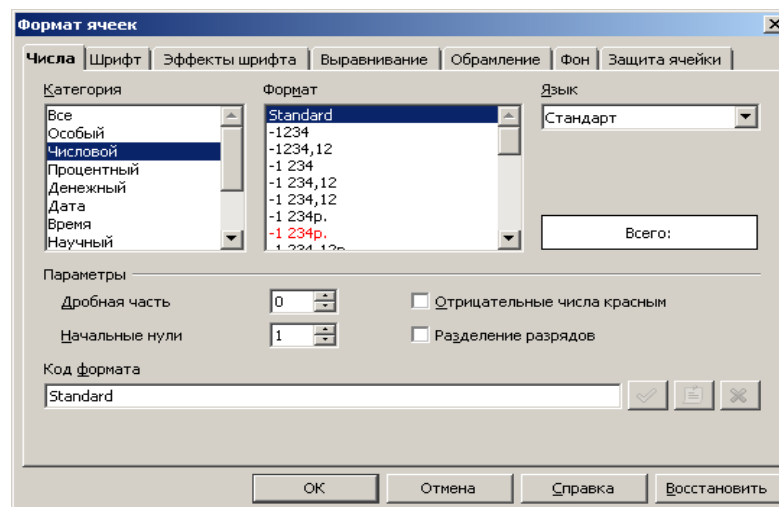
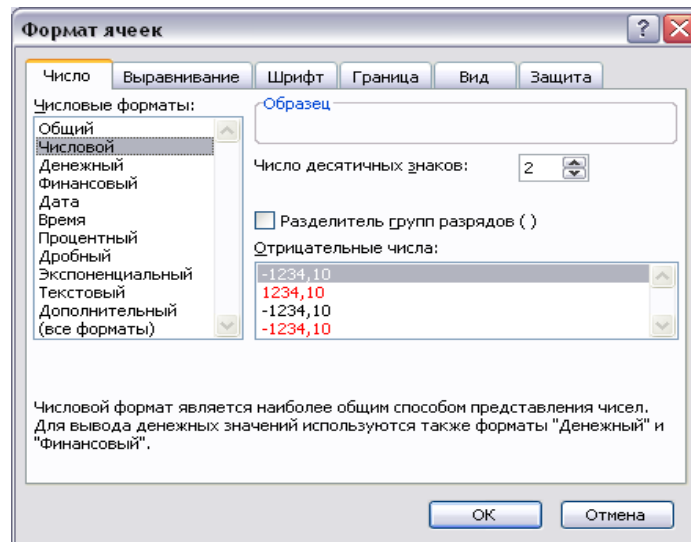


Рис.4.4. Визначення формату даних у програмах Excel та Calc.

Формати чисел у вибраних клітинках задають командою **Формат→Ячейки...→** вкладка **Числа → Числовые форматы** вибрати потрібний формат. Якщо формат має додаткові параметри, то після його вибору на вкладці з'являються поля, в яких можна зазначити ці параметри.

Роздільником цілої та дробової частини в числах може бути крапка або **кома** залежно від налаштування операційної системи.

Общий (Calc - Все) формат відображає дані в такому вигляді, як вони вводяться.

Числовой формат дає змогу встановлювати кількість відображуваних десяткових знаків, змінювати вигляд від'ємних чисел (наприклад, виділяти їх червоним кольором).

Формат *Денежный* має всі можливості форматування формату *Числовой*, а також дає змогу вибирати грошову одиницю.

Финансовый формат подібний до грошового, але має додаткові можливості щодо вирівнювання грошових одиниць відносно роздільника цілої і дробової частин.

Процентный формат при застосуванні до числа формату десяткова кома зміщується в числі на два знаки вправо, а в кінці числа виводиться знак відсотків "%".

Дробный формат дає змогу виводити числові значення у вигляді звичайних дробів. Вигляд дробу формату *Дробный* обирають зі списку *Тип*.

Экспоненциальный (Calc -Научный) формат. У форматі числа подаються у вигляді $aE \pm n$, де a — ціле число або десятковий дріб (мантиса); n — ціле число (ступінь, порядок). Зображення $aE \pm n$ означає, що в клітинці розміщується число $a \cdot 10^{\pm n}$.

Текстовый формат. Застосування до клітинки формату означає, що записані в клітинці число або формула вважатимуться текстом.

Дата та Время. Формати відображення дати або часу, Дату можна ввести в одному з форматів Д.М.ГГ, а час — в форматі Ч.ММ.СС.

Дополнительный формат (Calc -Пользовательский). Створюється користувачем, для таких даних, як номер телефону, індекс тощо.

Також до форматування відносяться такі дії, як: вирівнювання вмісту клітинок, варіанти розміщення даних у клітинці, все це задається у вікні **Формат→Ячейки...→** вкладка **Выравнивание** (рис. 4.5).

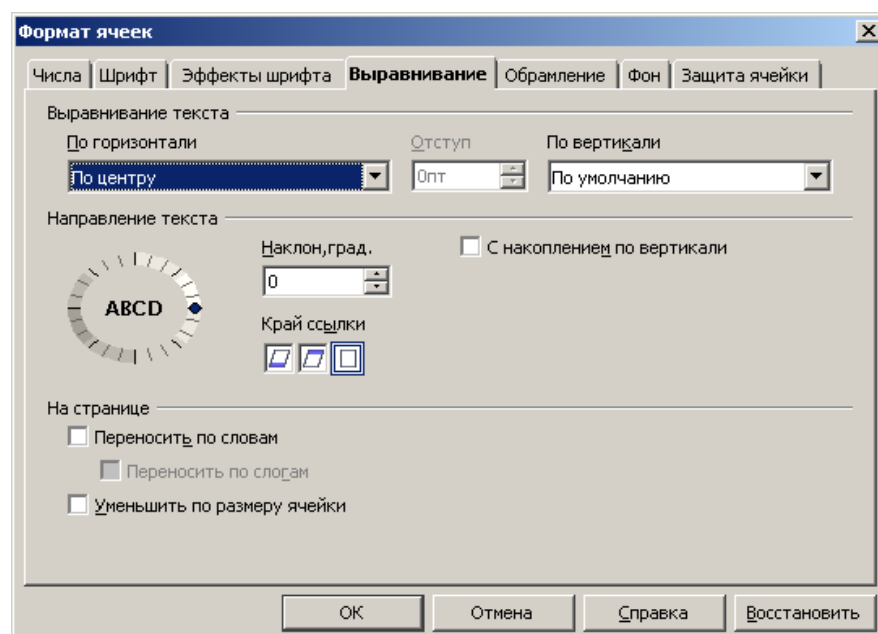
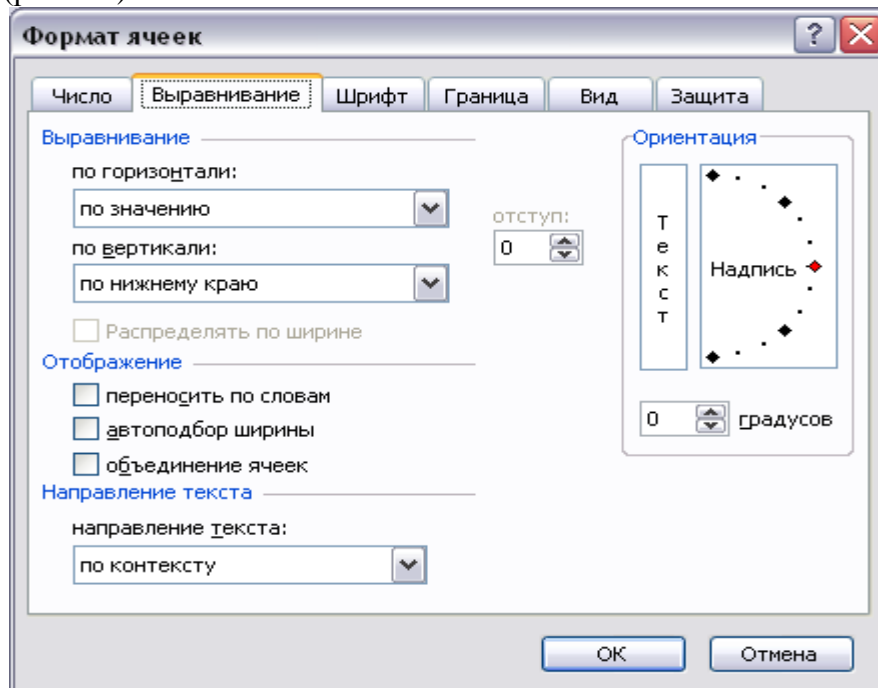


Рис.4.5. Можливості вирівнювання даних у програмах.

За замовчуванням для горизонтального вирівнювання застосовується варіант *по значенню*, згідно з яким числові значення вирівнюються відносно правого краю клітинки, а текстові — відносно лівого. Серед решти шести варіантів відокремимо вирівнювання *по центру виділення*, згідно з яким текст центрується відносно всіх виділених праворуч незайнятих клітинок. Список по вертикалі містить чотири варіанти *вертикального* вирівнювання тексту: *по верхньому краю, по центру, по нижньому краю, по висоті*.

Орієнтація тексту дає змогу повернути текст у клітинці на будь-який кут відносно горизонталі, для цього потрібно скористатися стрілкою або полем-лічильником градусів.

Якщо ширини стовпця недостатньо для відображення вмісту клітинок, її можна змінити одним з таких способів:

- помістіть курсор миші в рядку заголовків стовпців на лінію, що відокремлює стовпець від сусіднього стовпця праворуч; коли курсор набере вигляду перехрестя з двонапрямною стрілкою клацніть лівою кнопкою миші;

- виділіть будь-яку клітинку потрібного стовпця або весь стовпець, клацнувши на його заголовку, і виконайте команду **Формат → Столбец → Автоподбор ширины**;

- **Формат → Ячейки... → вкладка Выравнивание → Отображение → автоподбор ширины (Calc – На странице → Уменьшить по размеру ячейки)**.

В усіх випадках ширину стовпця буде встановлено для повного відображення найдовшого в ньому значення.

Щоб змінити висоту рядка, помістіть курсор миші в області заголовка рядка на лінію під номером рядка. Коли курсор набере вигляду перехрестя з двонапрямною стрілкою, натисніть кнопку миші й перетягніть лінію, яка розділяє рядки, у нове місце.

Обрати параметр **Отображение → переносить по словам**, якщо треба переносити текст у клітинці.

Для **об'єднання клітинок** треба виділити клітинки, які об'єднуються **→ Формат ячеек... → вкладка Выравнивание** діалогового вікна **Формат ячеек → опцію Объединение ячеек**.

В об'єднанні клітинки зберігаються тільки дані верхньої лівої клітинки. Щоб зберегти дані інших клітинок, їх потрібно перед об'єднанням перенести до верхньої лівої клітинки.

Щоб застосувати обрамлення до окремих клітинок, групи клітинок або всієї таблиці, виконайте такі дії: виділіть потрібний діапазон клітинок **→ Формат → Ячейки... → вкладка Граница (Calc – Оформление)** виберіть тип і колір лінії обрамлення, зазначте вигляд обрамлення.

Щоб вибрати колір для виділених клітинок, на вкладці **Вид (Calc – Фон)** зазначте колір і/або візерунок тла.

Умове форматування.

На відміну від звичайного формату клітинки або діапазону, умове форматування спрацьовує зміною формату на новий, попередньо зазначений користувачем формат, лише при виконанні певних умов. Типово цими умовами є значення в цій самій клітинці. Таких спеціальних умов застосування зміни формату може бути до трьох.

Для застосування до клітинки/діапазону умовного форматування треба виділити клітинку/діапазон **→ Формат → Условное форматирование → вікно Условное форматирование**. У вікні групою керуючих елементів пропонується задати першу умову застосування форматування. Для кожної умови (від однієї до трьох) може бути свій варіант форматування.

Є два типи умов: порівняння із значенням клітинки або умови, що задаються логічними функціями. Для значення пропонуються всі основні варіанти відношення ($=$, $<$, $>$, $<=$, $>=$, *вне*) у списку вибору відношення. Залежно від вибраного відношення справа від списку відношень з'являється необхідна кількість полів для значень чи посилань на клітинки із значеннями. Формулу як критерій застосування умовного форматування використовують для звернення до будь-яких функцій та виразів, які повертають значення **ИСТИНА** або **ЛОЖЬ**.

Задавши умову форматування, натисніть кнопку **Формат** → вікно **Формат ячеек** → задайте формати шрифту, меж і колір тла клітинки. Якщо потрібна ще одна умова застосування для клітинки/діапазону умовного форматування (тепер вже друга чи третя), натисніть кнопку **А також**, якщо всі чи кілька умов (загалом їх три) потрібно анулювати, натисніть кнопку **Удалить**. Відкривається діалогова форма, де зазначається умова, що видаляється.

Питання для самоперевірки

1. Які типи даних опрацюють ЕТ?
2. Які числові формати застосовують в ЕТ?
3. Як здійснити вертикальне вирівнювання чисел у грошовому форматі в стовпці таблиці?
4. Як змінити позначення грошової одиниці в комірці?
5. Як відмінити об'єднання комірок?
6. Що означає вирівнювання по центру виделения?
7. Як вибрати товщину і колір лінії для обрамлення діапазону комірок?
8. Які числові формати дають змогу встановити кількість десяткових знаків у зображенні чисел? Як це здійснити?
9. Як задати для таблиці альбомну орієнтацію сторінки?
10. Як розмістити таблицю в центрі сторінки?

4.3. Редагування даних в ЕТ

Редагувати у ЕТ можна клітинки, стовпці, рядки, сторінки (Листы), формули, тобто можна видаляти, переміщувати, копіювати табличні дані, автоматизовано заповнювати даними, додавати або видаляти клітинки, стовпці, рядки та сторінки, переіменовувати клітинки, стовпці, сторінки.

Видаляти дані з виокремленої клітинки або діапазону клітинок можна командою **Правка** → **Удалить...** або командою з контекстного меню або з клавіатури.

Також можна **очистити** клітинки, не тільки їх зміст, а також формат, примітки тощо, **Правка** → **Очистить** далі обрати потрібне.

Для **переміщення та копіювання** даних треба виокремити дані, далі обрати команду **Правка** → **Копировать (Вырезать)**, а потім **Правка** → **Вставить** або командами з контекстного меню або кнопками панелі інструментів.

У ЕТ існує команда спеціальної вставки, яка дає можливість перемістити або скопіювати не тільки значення у клітинках, а також їх формат, формули тощо, **Правка** → **Специальная вставка...**

Для того щоб **додати або виділити** стовець або рядок виокремити заголовок або діапазон клітинок **Вставка** → **Строка (Столбец)** або **Правка** → **Удалить**.

Переіменувати клітинку можна командою **Вставка** → **Имя** → **Присвоить...** (**Calc** **Вставка** → **Названия** → **Определить...**). Нове ім'я повинно починатися літерою та не мати пропусків.

У формулах для зручності використання аргументів, можна переіменувати стовпці або рядки назвами, як у формулі, для цього треба відкрити режим «**Автоматически искать названия столбцов и строк**» **Сервис** → **Параметры** → вкладка **Вычисления**.

Заповнення клітинок даними. Можна вводити послідовно дані, копіювати та переміщувати, а можна скористатися командою **Правка** → **Заполнить** → **Влево(Вправо Вверх Вниз)** перед цим заповнити клітинку та виокремити діапазон заповнення.

Для заповнення клітинок послідовностями типу арифметичною чи геометричною прогресіями, датами, послідовністю чисел, а також для копіювання однотипних формул, можна скористатися можливістю **автозаповнення**. Основними засобами автоматизації обчислень в ЕТ є **автоматичне переобчислення** всієї таблиці в разі зміни будь-яких вхідних даних і можливість копіювання однотипних формул.

Копіювання формул і автоматичне переобчислення у таблиці — це два головні засоби автоматизації обчислень в ЕТ.

Під час копіювання формули відбуваються такі дії:

- формула вводиться в інші клітинки автоматично;
- формула автоматично модифікується — змінюються відносні адреси, на які є посилання у формулі.

Наприклад, під час копіювання формули $=B3*C3$ з третього рядка у четвертий формула в четвертому рядку набуде вигляду $=B4*C4$.

Перший спосіб, заповнити одну чи дві клітинки та виокремити їх, потім протягнути за *маркер заповнення* (чорний прямокутник у правому нижньому куті рамки поточної клітинки).

Другий спосіб, заповнити першу клітинку початковим значенням → **Правка** → **Заполнить** → **Прогрессия...** (*Calc* **Ряды**) (Рис. 4.6)

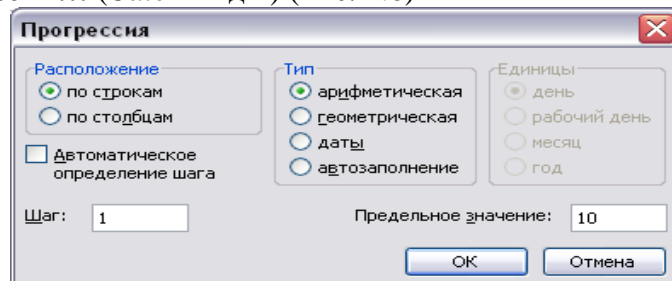


Рис.4.6. Діалогове вікно для заповнення клітинок даними (Ms Excel)

Можно створити свою власну послідовність будь яких значень (накштал список назв днів тижня, назв місяців тощо) *список*. **Сервис** → **Параметры...** → вкладка **Списки** та ввести свою послідовність (Рис. 4.7).

Редагування сторінок. З Листами можна виконувати слідуєчі дії: додавати, видаляти, копіювати та переміщувати, переіменувувати. Усі дії зручно виконувати з допомогою контекстного меню, для чого треба обрати потрібний Лист на Панелі ярликів і натиснути праву кнопку миши. А також можна виконувати ці дії командами **Правка**, **Вставка** та **Формат**.

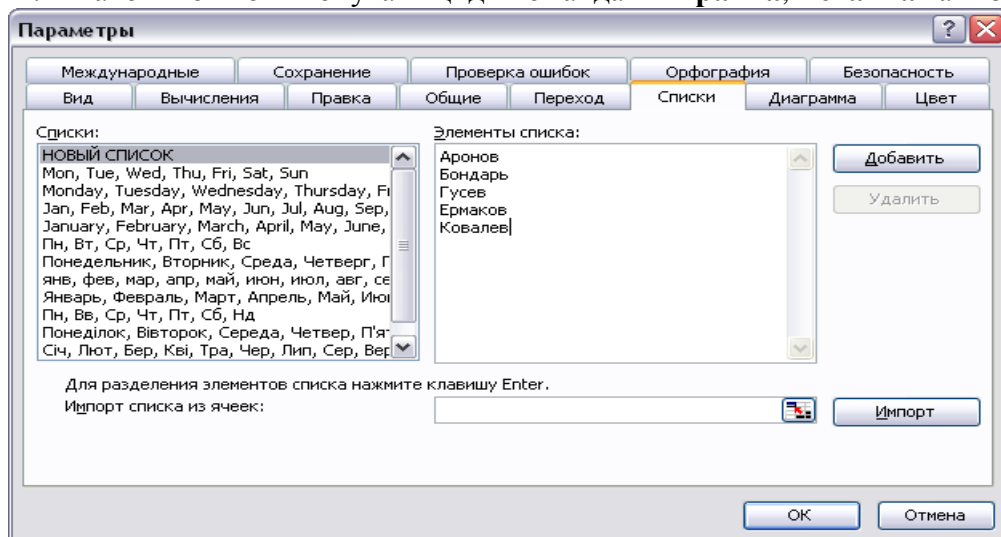


Рис.4.7. Створення власного списку (Ms Excel)

Типові *помилки*, які зустрічаються при введенні та роботі з даними. Якщо замість результатів у клітинці з'являється #####, то це означає, що велике число в клітинці не поміщається, отже, стовпець треба зробити ширшим, перетягнувши межу в заголовку стовпця.

Помилка ### - з'являється, якщо стовпець недостатньо широкий або дата чи час - є від'ємними числами.

Помилка #DIV/0! - з'являється при діленні числа на 0 (нуль).

Помилка #ИМЯ? - з'являється, якщо не можна розпізнати ім'я у формулі.

Помилка #ССЫЛ! - з'являється, якщо ссылка на клітинку вказана невірно.

Для виправлення будь якої з помилок **Сервис → Зависимости → Источник ошибки**.

Питання для самоперевірки

1. Як відредагувати дане у клітинці?
2. Як вставити стовпець у таблицю?
3. Як заповнити стовпець арифметичною прогресією?
4. Як вилучити рядок з таблиці?
5. Як вставити сторінку у книгу?
6. Як очистити весь стовпець?
7. Що таке список, як створити свій власний?
8. Як скопіювати формулу?
9. Які є способи заповнення рядка числовою послідовністю?
10. Де можна знайти, як виправити помилки?

4.4. Функції та обчислення у ЕТ

Обчислення в ЕТ. Обчислення в таблиці ведуться автоматично.

Це означає, що зміна будь-якого вхідного даного одразу ж веде до переобчислень всієї таблиці (**ОО Сервис → Содержимое ячейки → Пересчитать**).

Режим Автоматично можна вимкнути і скористатися ручним переобчисленням за допомогою клавіші F9.

Використання стандартних функцій. Функції у ЕТ — це скорочений варіант формул, які часто застосовують, тобто це програми, що за готовими формулами реалізують обчислення над аргументами, які задаються користувачем згідно з правилами запису (синтаксису) функцій. Кожна функція має ім'я (латинські або кирилицею у Excel, тільки латинські у Calc) та аргументи, які записують у круглих дужках після імені функції через крапку з комою. Аргументами функцій можуть бути числа, посилання на клітинки та діапазони, імена, текст, логічні значення, значення помилок, вкладені функції тощо. Доступ до функції здійснюється за її іменем.

Програми ЕТ мають декілько сотен стандартних функцій, а також є можливість створити власну функцію. Усі функції умовно поділяються на категорії:

Excel	Calc
Финансовые	Работа с базами данных
Дата и время	Дата и время
Математические	Финансовые
Статистические	Информационные
Ссылки и массивы	Математические
Работа с базой данных	Массив
Текстовые	Статистические
Логические	Электронная таблицы
Проверка свойств и значений	Текстовые
Инженерные	Подключаемые модули
Информационные	Логические
Мат. и тригонометрические	Дополнительные

Введення функцій. Можна вводити функції вручну безпосередньо з клавіатури у клітинці або у **Строке Формул** після знака «=», якщо синтаксис функції відомий, або за допомогою **Мастера функций**, який здійснює введення функцій у діалоговому режимі за декілько кроків.

Мастер функций побудовано за тематичним принципом (Рис. 4.8). У списку **Категория** вибрати потрібну, клацнувши лівою кнопкою миші, поряд зі списком можна отримати повний список імен функцій активної категорії. Обрав функцію, нижче з'явиться повний її запис та призначення. Функцію викликають клацанням лівою кнопкою миші на її імені. У вікні **Аргументы функции** вводять у відповідне поле (чи поля) аргументи функції числа або адреси клітинок, що містить число. Для зручності введення адрес клітинок можна скористатись

кнопкою справа від поля, клацання на якій згортає вікно і відкриває активний лист, що дає змогу мишею визначити потрібну клітинку чи діапазон клітинок.

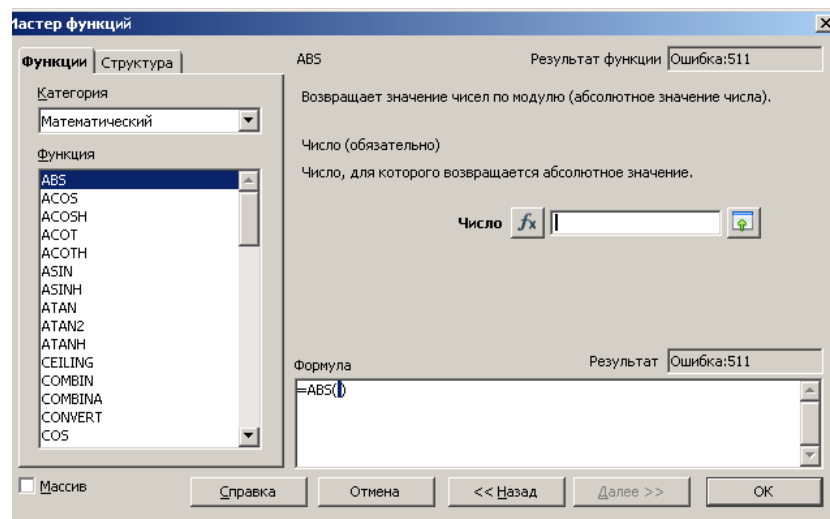
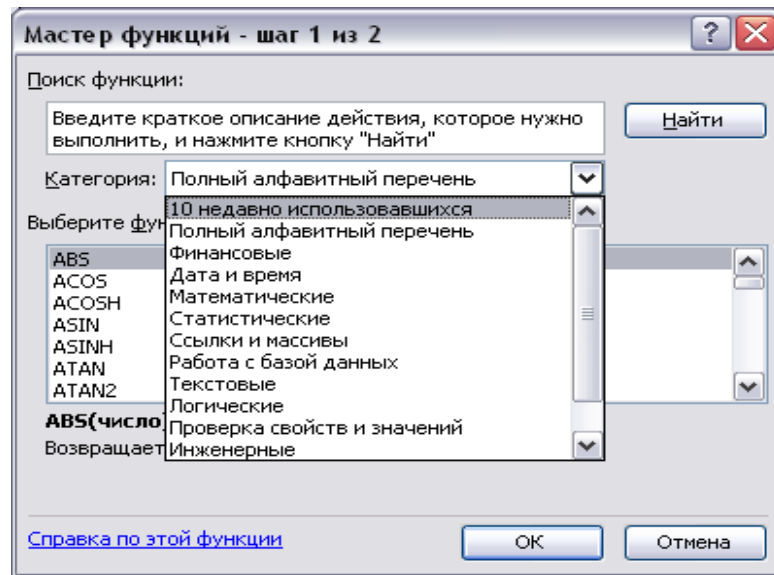




Рис.4.8. Вікна Мастера функцій

Для виклику довідки про використання функції і пояснення щодо її аргументів у вікні функцій у лівому нижньому куті вікна натисніть посилання **Справка по этой функции**.

Викликати **Мастер функций** можна декількома способами, спочатку виділити клітинку

- **Вставка** → **Функция**;
- (не має у *Calc*) Панель інструментів **Стандартная** → кнопку  стрілка униз →
- **Другие функции...**;
- **Строка Формул** → кнопку  (**Вставка функции**);
- ввести знак «**=**» → у **Поле Имени** → **Другие функции...**;
- (тільки у *Calc*) **Вставка** → **Список функций** (рис. 4.9).

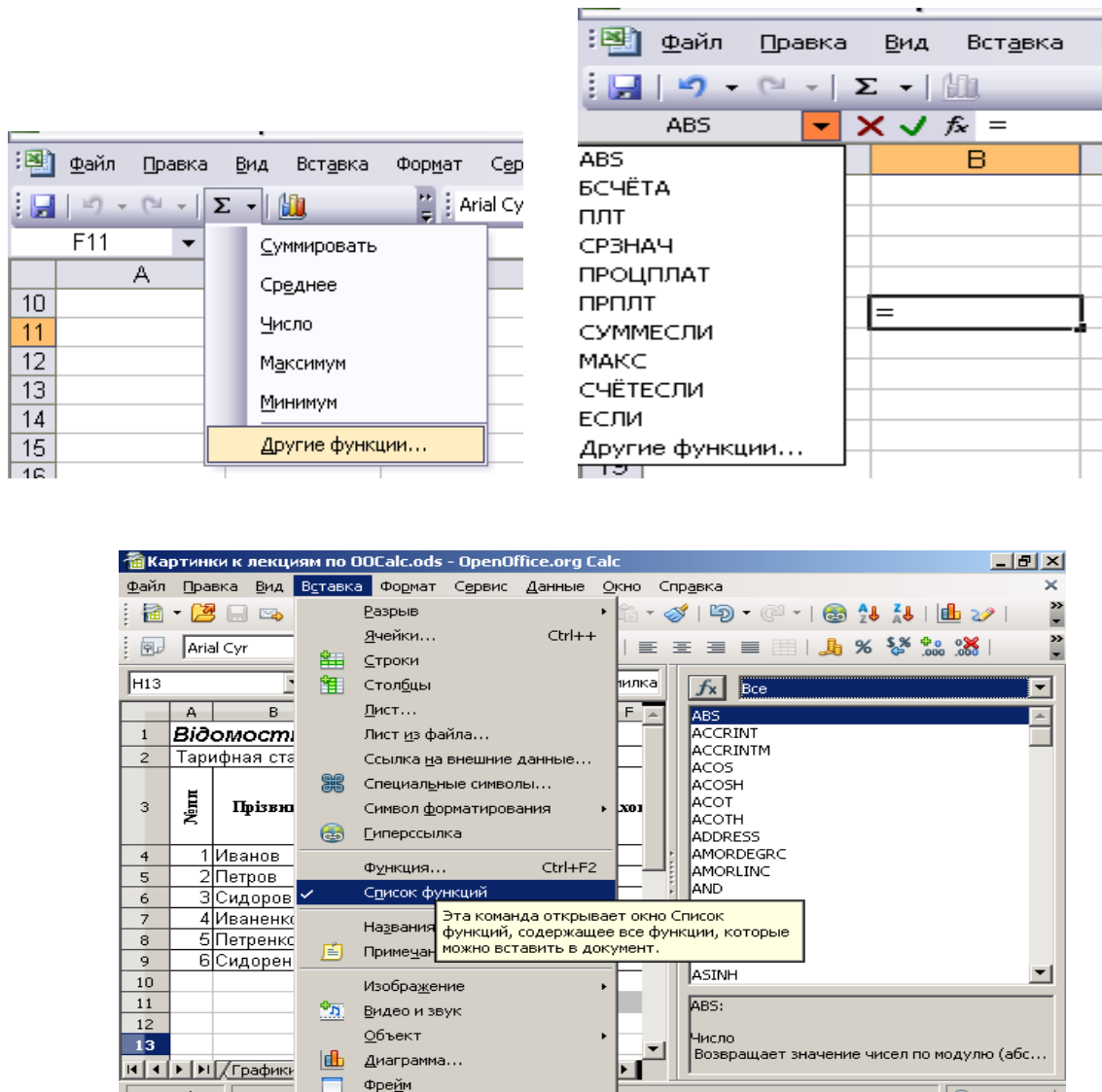


Рис.4.9. Виклик Мастера функций кнопкою ПИ та в Поле Имени (Excel)
та виклик Мастера функций (Calc)

Також можна викликати функції, що використовувались в останнє, у **Поле Имени**. Окремо виділено у ЕТ дію, що підсумовує значення, крім функції **СУММ()**, що підсумовує значення діапазону клітинок, застосовують **Автосуму**. Підсумувати значення рядка (стовпчика) суміжних клітинок можна, виділивши клітинку, що замикає рядок справа (стовпчик — знизу), і клацнувши кнопку **Автосумма** Σ Ппанели инструментов **Стандартная (Calc- Ппанель инструментов Строки Формул)**. Якщо виділити діапазон клітинок і натиснути кнопку **Автосумма**, функцію **СУММ** буде застосовано автоматично до всіх стовпців діапазону, а результати розташовано в наступних клітинках.

Крім того, окремі статистичні функції, наприклад **СУММ**, **МАКС**, **МИН**, можна швидко обчислити без використання формул, виділивши потрібний діапазон клітинок і ввімкнувши **режим Автообчислення** клацанням правої кнопки миші в рядку **Строка Состояния** на полі **Автовывчисление**. Результат обчислень буде зображений у цьому ж полі, наприклад, **Сумма=58**. У подальшому значення останньої вибраної функції обчислюватиметься автоматично.

Функції ЕТ. Розглянемо деякі функції.

Функції часу та дати. Функції дозволяють аналізувати та працювати зі значеннями дати та часу.

ГОД(дата в числовом форматі) – повертає рік відповідно дати (*Calc- YEAR*).

ДАТА(год;місяц;день) – повертає дату у встановленому форматі (*Calc- DATA*).

СЕГОДНЯ() – повертає поточну дату у встановленому форматі (*Calc- TODAY*).

ВРЕМЯ(часі;хвилини;секунди) – повертає час у встановленому форматі (*Calc- TIME*).

Математичні функції, за допомогою яких можна виконати загальні обчислення. Аргументами цих функцій є числові дані. Після обчислення функція повертає числовий результат. Математичні функції поділяються на чотири типи: *арифметичні, логарифмічні, тригонометричні та матричні, функції перетворення чисел або результатів обчислень, особливе місце займає функція додавання.*

Арифметичні функції

СУММ(число1; число2;...) - повертає суму аргументів (*Calc- SUM*).

ПРОИЗВЕД(число1; число2;...) - повертає добуток своїх аргументів (*Calc- PRODUCT*).

ABS(число) - повертає абсолютне значення числа.

КОРЕНЬ(число) - повертає додатне значення квадратного кореня числа (*Calc- SQRT*). Якщо число від'ємне, з'являється повідомлення про помилку #ЧИСЛО!.

СТЕПЕНЬ(число; показник степеня) - повертає результат піднесення до степеня (*Calc- POWER*). Цей вираз є еквівалентним до виразу $\text{число}^{\text{показник степеня}}$.

ОКРУГЛ(число; кількість розрядів) - повертає результат округлення до зазначеної кількості десяткових розрядів (*Calc- ROUND*).

ОКРВВЕРХ(х; точність) та **ОКРВНИЗ(х; точність)** - повертає результат округлення з надлишком (з нестачею) до найближчого числа, що кратне точності (*Calc- ROUNDDOWN* та *ROUNDUP*).

ОТБР(число; кількість розрядів) - залишає зазначену кількість розрядів числа після коми, відкидаючи всі наступні (не виконуючи округлення) (*Calc – TRUNC*).

СЧЁТЕСЛИ(Діапазон;Критерий) - повертає кількість тільки тих аргументів, які відповідають певним умовам (*Calc – COUNTIF*).

Діапазон — діапазон клітинок, для яких перевіряється критерій; Критерий — умова відбору у формі числа, виразу або тексту.

Розривні діапазони не підтримуються. Значеннями в клітинках можуть бути числові дані в усіх різновидах, логічні значення, слова і фрази (клітинки можуть бути також порожні). Допустимі значення можна задавати як константою, так і формулою.

СУММЕСЛИ (Діапазон; Критерий;Діапазон_Суммирования) - повертає суму клітинок діапазону, які задовольняють заданий критерій (*Calc - SUMIF*).

Діапазон_Суммирования визначає діапазон клітинок, в якому, власне, виконується додавання значень клітинок, для яких відповідні клітинки у параметрі Діапазон задовольняють Критерий. Параметр Діапазон_Суммирования можна не зазначати — необов'язковий параметр, що визначає клітинки, значення яких долучаються до підсумку в разі виконання умови. Діапазони Інтервал та Сумм_інтервал повинні мати однакову кількість клітинок. Якщо аргумент Сумм_інтервал пропущений, то розраховується сума з клітинок діапазону Інтервал, для яких справджується критерій.

Логарифмічні функції

LN(число) повертає натуральний логарифм числа.

LOG10(число) повертає десятковий логарифм числа.

Тригонометричні функції

ПИ() повертає число π з точністю до 15 цифр (*Calc*- **PI**). Наявність відкриваючих та закриваючих дужок обов'язкова.

РАДИАНН(кут в градусах) - обчислює значення кута в радіанах (*Calc*- **RADIANS** та оборотня - **DEGREES**).

SIN(кут) - повертає синус кута, заданого в радіанах. Часто значення кута задане у градусах. У такому разі необхідно використати попередню функцію.

COS(число), TAN(число) - повертає косинус і тангенс заданого числа відповідно.

ASIN(число) - повертає арксинус числа, тобто кут, синус якого дорівнює числу.

Матричні функції

Як правило, матричні функції застосовують для розв'язування систем лінійних рівнянь. Наведемо деякі матричні функції. У якості аргументів у *Excel* використовується значення масив, у *Calc* - матриця.

МУМНОЖ(масив1;масив2) – повертає добуток матриць (*Calc* – **MMULT**).

МОПРЕД(масив) - повертає визначник матриці (*Calc* - **MDETER**).

МОБР(масив) - повертає обернену матрицю (*Calc*- **MINVERSE**).

ТРАНСП(масив) – повертає транспоновану матрицю (*Calc*- **TRANSPOSE**).

Статистичні функції

Програма ЕТ містить комплекс засобів статистичного аналізу, до якого входять стандартні статистичні функції. Аргументом цих функцій найчастіше є діапазон клітинок або список чисел. Розглянемо стандартні статистичні функції, які використовують найчастіше.

МАКС(число1;число2;...) та **МИН(число1;число2;...)** - повертають відповідно найбільше та найменше значення з набору аргументів (*Calc* - **MAX** та **MIN**).

СЧЁТ(значение1;значение2;...) - визначає кількість клітинок у заданому діапазоні, які містять числа (у тому числі дати й формули, що повертають числа) (*Calc* – **COUNT**).

СРЗНАЧ(Число1 ;Число2;...) - обчислює середнє арифметичне своїх аргументів (*Calc* - **AVERAGE**).

Логические функції

У ЕТ логіческі функції, це функції з допомогою яких реалізуються базові логічні операції: множення -**AND**; складання - **OR** ; заперечення -**NOT**.

ЕСЛИ(условие; Значение_для_Истина; Значение_для_НЕ_Истина) – повертає одне значення аргумента условие ИСТИНА або ЛОЖЬ (*Calc* - **IF**).

Значение_для_Истина — це значення чи вираз, який поверне функція ЕСЛИ() як істинне значення параметра Условие. Якщо параметр не зазначити, то як результат ЕСЛИ() повернеться ИСТИНА. Виразом у параметрі, зокрема, може бути будь-яка інша функція, зокрема ЕСЛИ().

Значение_для_НЕ_Истина — це значення чи вираз, який поверне функція ЕСЛИ() як хибне значення параметра Условие. Якщо параметр не зазначити, то як результат ЕСЛИ() повернеться ЛОЖЬ. Виразом у параметрі, зокрема, може бути будь-яка інша функція, зокрема ЕСЛИ()

Зверніть увагу:

– якщо у функції ЕСЛИ() пропускається другий параметр, то символ ";" перед третім параметром має бути обов'язково, ЕСЛИ(Условие; ;Значение_для_НЕ_Истина);

– допускається до семи вкладень функцій ЕСЛИ() одна в одну.

— для складніших умовних виразів, коли задається багато значень, потрібно використовувати функції **И**(Pr;Pr;...) та **ИЛИ**(Pr;Pr;...) (**AND** або **OR**), де Pr — умовний вираз. Ці функції мають бути вкладені у функцію ЕСЛИ() на місці Условие.

Фінансові функції.

Фінансові функції використовують для розв'язування задач планування фінансової діяльності, визначення прибутків, аналізу вигідності капіталовкладень, кредитно-інвестиційної політики тощо.

Інвестицією називається вкладання грошей у деякий бізнес на певних умовах. Позика у банку називається *кредитом*, а внесок на рахунок в банк — *депозитом*, надходження грошей від деякого бізнесу — *рентою*.

Розглянемо головні *параметри фінансових функцій* та їхні скорочені назви:

процентна ставка за період **Ставка** (англ.: **Rate**) виражається у відсотках і може бути добовою, місячною, річною тощо;

кількість періодів **Кпер** (англ.: **NPER**), кожний тривалістю доба, місяць, рік тощо;

періодична виплата **Плт** (англ.: **PMT**) — сума, яку виплачує клієнт за кожний період (це від'ємне число), або сума, яку отримує клієнт за кожний період (це додатне число);

сума внеску **Пс** (англ.: **PV**) — сума інвестиції, капіталовкладення, початкового внеску (від'ємне число або нуль);

тип операції **Тип** (англ.: **Type** або **F**) — число 0, якщо виплата дійснюється наприкінці кожного періоду, і число 1, якщо на початку.

Вирізняють *кредитну і депозитну* процентні ставки. *Кредитна* ставка є вищою за *депозитну*. Процентна ставка має бути узгодженою з тривалістю періоду, наприклад, річна ставка 24% рівносильна місячній ставці 2%.

Функція *для визначення майбутньої вартості теперішніх інвестицій* на основі періодичних постійних (за значенням сум) платежів та постійної процентної ставки

БС(Ставка; Кпер; Плт; Пс; Тип) **(FV)**

Функція для визначення майбутньої вартості інвестиційного капіталу на умовах нарахування різних процентів за певну кількість (до 30) періодів

БЗРАСПИС (капітал; масив процентів) **(англ.: FVSCHEDULE)**

Функції для визначення виплат для погашення позики.

Введемо нові терміни та їхні скорочені назви:

номер періоду **Период** *(англ.: P), Период = 1 на початку періоду;*

сума позики **Пс** *(англ.: PV);*

кінцеве значення позики **Бс** (англ.: **FV**).

Функція *для визначення суми періодичних виплат для погашення боргу* на основі періодичних постійних (за значенням сум) платежів та постійної процентної ставки

ПЛТ(Ставка; Кпер; Пс; Бс; Тип) (англ.: **PMT**)

Така виплата складається з двох частин, які обчислюють за допомогою двох функцій:

а) *виплата за процентами* **ПРПЛТ(Ставка; Період; Кпер; Пс; Бс; Тип);** (англ.: **IPMT**)

б) *основна виплата* **ОСПЛТ(Ставка; Період; Кпер; Пс; Бс; Тип).** (англ.: **PPMT**)

Виплата за процентами кожний період зменшується, а основна виплата кожний період зростає, їхня сума постійна і дорівнює **ПЛТ**.

Функція *яка обчислює кількість періодів, потрібних для погашення суми позики*, на наданої під деяку процентну ставку за умови заздалегідь заданої суми періодичних виплат

КПЕР(Ставка; Плт; Пс; Бс; Тип) (англ.: **NPER**)

Функція, що визначає вигідність надання позики, тобто реальну процентну ставку в 0 надання позики на певну суму за умови фіксованих періодичних виплат протягом деякої кількості періодів

СТАВКА(Кпер; Плт; Пс; Бс; Тип; початкове наближення) (англ.: **RATE**)

Тут потрібно задати деяке початкове наближення до шуканої процентної ставки, наприклад 0,1 (10%)..

Функції для визначення теперішньої вартості майбутніх інвестицій, тобто функції *для визначення вигідності інвестицій (капіталовкладень) у деякий бізнес*.

Функція обчислює сьогоднішню вартість низки майбутніх надходжень (ренти) від бізнесу

ПС(ДПС; Кпер; Плт; Бс; Тип) (англ.: **PV**)

де **ДПС** — депозитна процентна ставка, тобто ставка, яку банк виплачує за вклади клієнтів.

Для обчислення сьогоднішньої вартості майбутньої ренти використовується *принцип дисконтування* — зведення суми ренти за деякий термін до її вартості у певний момент часу. Суттєво враховується депозитна процентна ставка, оскільки вважається, що вже перше надходження стає депозитом у банку. Дисконтування дає відповідь на запитання, чи варто вкладати гроші в певний бізнес, чи краще їх поставити у банк під відсотки.

Функція що обчислює сьогоднішню вартість різних рент, які надходять наприкінці рівномірних періодів

ЧПС(ДПС; рента1; рента2;...) (англ.: **NPV**)

Функції користувача. Користувач може для власних розрахунків створити свою функцію з певними діями та користуватися їй як стандартними. Для цього треба скористатися

вбудованим редактором Visual Basic Сервис → Макрос → Редактор Visual Basic. У вікні редактора необхідно записати всі дії майбутньої функції та дати їй ім'я.

Питання для самоперевірки

1. Як здійснити обчислення в таблиці?
2. Які елементи може містити формула?
3. Що таке копіювання формул?
4. В чому суть автоматичного переобчислення у ЕТ?
5. На які категорії поділяють стандартні функції ЕТ?
6. Як ввести у формулу функцію?
7. Як реалізується розгалуження в ЕТ?
8. Який розділювач можна використовувати у списках аргументів функції?
9. Які способи існують у ЕТ для обчислення суми чисел?
10. Як створити функцію користувача?

4.5. Організація графічної інформації

Типи діаграм. Як правило, для аналізу даних, записаних у вигляді таблиці потрібно багато часу. Графічне зображення табличних даних дає змогу суттєво пришвидшити цей процес. Інструментом такого зображення в ЕТ є діаграми. За допомогою діаграм можна унаочнити основні властивості та співвідношення даних конкретної таблиці.

Існує декілька типів діаграм. *Гістограми* порівнюють значення кількох рядів споріднених даних. Окремі значення зображуються як вертикальні стовпці існують три основних види гістограм, які різняться методами порівняння даних:

- гістограми, які порівнюють абсолютні значення даних;
- гістограми з накопиченням, що показують як співвідношення окремих рядів даних, так і тенденції зміни сумарного значення всіх рядів;
- нормовані гістограми, які показують внесок кожного ряду в сумарне значення.

Лінійні діаграми відрізняються від гістограм лише тим, що значення в них зображуються горизонтальними смугами. Лінійні діаграми використовують тоді, коли потрібно зробити акцент на значення рядів даних (гістограми акцентують увагу на зміні значень рядів у часі).

Графіки, як і гістограми, показують тенденції зміни з часом значень рядів даних. При цьому значення відображаються точками. Крім того, ці точки з'єднуються лініями, отриманими за допомогою лінійної інтерполяції. Так само як гістограми графіки є з абсолютними значеннями, з накопиченням та нормовані.

Діаграми з областями утворюються з графіків у такий спосіб. Лінії даних на графіку є межами ділянок діаграми з областями. Кожна ділянка розфарбовується певним кольором. Серед діаграм з областями найчастіше використовують діаграми з накопиченням.

Кругова діаграма показує співвідношення значень єдиного ряду даних. При цьому значення зображуються як сектори круга.

Кільцева діаграма — це різновид кругової діаграми, призначений для зображення даних кількох рядів. Кожному ряду відповідає кільце, а внескові конкретного значення в загальну суму ряду - сегмент кільця.

Тонкові діаграми використовують для графічного зображення залежності кількох рядів даних від фіксованого ряду, який утворює вісь абсцис. При цьому залежності можуть зображатись як послідовностями точок, так і інтерполяційними лініями. Тонкові діаграми схожі на діаграми- графіки. Принципово вони різняться лише тим, що у графіках значення фіксованого ряду (найчастіше це часовий ряд) змінюються з однаковим інтервалом, у той час як у точковій діаграмі інтервали між значеннями фіксованого ряду можуть бути довільними.

Кулькові діаграми є різновидом точкових діаграм. Їх використовують тоді, коли потрібно зобразити залежність між трьома параметрами. При цьому значення третього параметра визначає величину маркера даних (розмір кульки).

Елементи діаграми. Структуру кожної діаграми утворюють елементи, які визначають при побудові діаграми бажаного вигляду. Розрізняють кілька основних елементів діаграм.

Область побудови діаграми містить графічну частину діаграми.

Заголовки — це назви діаграми та координатних осей.

Ряд даних — це група взаємопов'язаних елементів даних діаграми. Кожний ряд даних відображається певним кольором (і формою маркерів). На діаграмі може бути кілька рядів даних (виняток становлять кругові діаграми). Залежно від розташування даних у таблиці ряди даних поділяють на вертикальні та горизонтальні.


Маркер — це графічний символ, що зображує на діаграмі конкретну точку даних (елемент ряду даних).

Легенда дає змогу ототожнити ряди даних. Вона пояснює, як зображено на діаграмі кожний ряд.

Мітка даних (підпис) — це значення і/або категорія точки даних, що наведені на діаграмі біля маркера даних.

Осі значень та категорії використовують для вимірювання значень даних на діаграмі. Вісь значень, як правило, розташовується вертикально (вісь Y). Категорії зазвичай відображаються на горизонтальній осі (вісь X). При цьому категорії можуть складатись як з числових даних, так і з текстових.

Сітка утворюється координатними лініями.

Для створення нових діаграм використовують **Мастер диаграмм Вставка → Диаграмма...** або кнопка Панелі інструментов **Стандартная**  (рис. 4.10). Зауважимо, що перед запуском **Мастера диаграмм** доцільно виділити дані (значення та категорії), за допомогою яких потрібно побудувати діаграму. У ЕТ *Excel* діаграми створюється у чотири етапи (кроки), кожному з яких відповідає певне діалогове вікно.

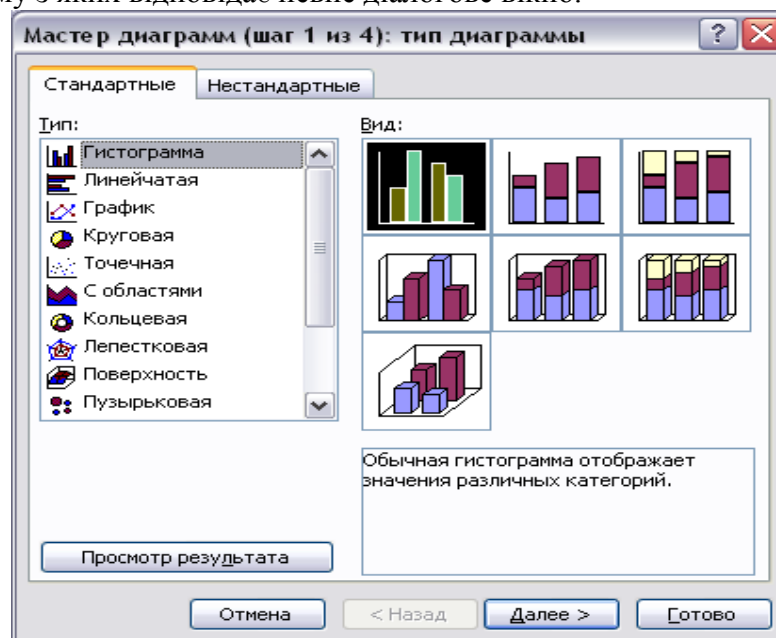


Рис. 4.10. Вікно Мастера диаграмм, перший крок (Excel)

1-й крок. Вибирається тип діаграми. Окрім описаних вже основних стандартних типів діаграм можна використовувати нестандартні типи, розміщені на вкладці *Нестандартные*. Серед них зазначимо насамперед типи чорно-білих діаграм, назви яких починаються з "ЧБ".

2-й крок. На цьому етапі потрібно зазначити область розташування даних для побудови діаграми. Область розміщення даних зазначають вкладка *Диапазон данных* → у полі *Диапазон*.

На цій самій вкладці задають орієнтацію рядів даних — горизонтальну (опція *Ряды в строках*) або вертикальну (опція *Ряды в столбцах*).

На вкладці **Ряд** задають інформацію окремо про кожний ряд даних: його назву, що використовується в легенді (поле *Имя*), область числових значень ряду (поле *Значения*) та область назв/ значень категорії (поле *Подписи оси X* або *Подписи категорий* залежно від типу діаграми).

3-й крок. На якому потрібно задати параметри діаграми. Усі параметри поділені на кілька груп, кожній з яких відповідає певна вкладка вікна:

- на вкладці *Заголовки* задають назву діаграми та осей значень і категорій (якщо вони є в цьому типі діаграми).
- на вкладці *Оси* (відсутня для кругових та кільцевих діаграм) зазначають, які осі повинні бути на діаграмі. Крім того, для осі X вибирають одну з трьох можливих шкал: автоматичну, за категоріями та часову.
- на вкладці *Линии сетки* (відсутня для кругових та кільцевих діаграм) вибирають типи координатних ліній, що відображатимуться на діаграмі. Для горизонтальних та вертикальних координатних ліній є два можливих види відображення: основне лінії (великі інтервали між лініями) та промежуточные линии (малі інтервали між лініями).
- на вкладці *Легенда* зазначають, чи потрібно додавати легенду до діаграми, а також де її розташувати на діаграмі.
- на вкладці *Подписи данных* можна додати підписи до елементів рядів даних. Як правило, вибирають один з трьох видів підписів: значення елемента, категорію елемента чи відносну величину (частку) значення елемента порівняно із сумарною величиною ряду.
- на вкладці *Таблица данных* (відсутня для кругових та кільцевих діаграм) до діаграми можна додати таблицю з даними, на основі яких побудовано діаграму. Як правило, цю можливість використовують при побудові діаграми на окремому аркуші.

4-й крок. На цьому кроці потрібно вибрати місце розташування діаграми:

- на окремому аркуші (ім'я нового аркуша потрібно ввести в поле *отдельно*) — діаграма займає окремий аркуш без клітинок;
- в одному з існуючих аркушів робочої книги (ім'я цього аркуша потрібно вибрати зі списку *имеющемуся*) — діаграма стає об'єктом всередині аркуша.

Якщо діаграму потрібно надрукувати на окремому аркуші паперу, використовують перший спосіб. Другий спосіб використовують, якщо потрібно на одному аркуші паперу надрукувати одночасно кілька діаграм (для порівняння) або діаграму разом з даними, на основі яких її побудовано.

У ET **Calc** послідовність та вигляд вікон декілько відрізняється (Рис.4.11).

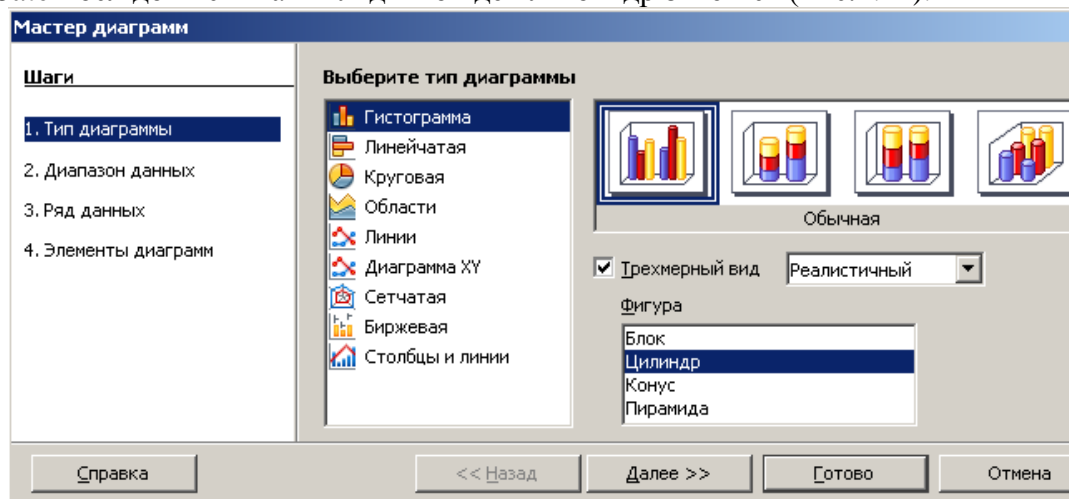


Рис.4.11. Вікно Мастера диаграмм, перший крок (Calc)

Редагування діаграм. Як правило, після створення діаграми **Мастером диаграмм** ще працюють над її зовнішнім виглядом. Якщо Ви забули встановити якісь параметри діаграми під час роботи з майстром, можна повернутися до будь-якого з його чотирьох кроків, скориставшись **Диаграмма → Тип диаграммы → Исходные данные → Параметры диаграммы → Размещение**.

Якщо потрібно розташувати діаграму на тому ж аркуші, що містить табличні дані, то, як правило, необхідно збільшити розмір діаграми. Для цього використовують маркери на метках діаграми. При цьому розмір шрифтів текстових елементів діаграми пропорційно збільшується.

Якщо потрібно відредагувати якийсь елемент діаграми, то зручно скористатися його *контекстним меню* командою **Формат → Название элемента**. Виконавши цю команду, можна отримати доступ до всіх властивостей елемента.

Додаткові поради щодо редагування діаграм.

- Майстер часто створює діаграму із сірим тлом області побудови. Для заощадження тонера принтера слід замінити це тло на біле. Для цього потрібно виконати команду **Формат области построения контекстного меню → тип заливания Обычная**;

- Зазвичай майстер невдало вибирає мінімальні та максимальні значення координатних осей. Діаграма буде точнішою, якщо при зображенні рядів даних повністю використано область побудови. Відредагувати мінімальне та максимальне значення координатної осі можна за допомогою команди **Формат оси → Шкала контекстного меню**.

Щоб надати якомога більше місця графічній частині діаграми, легенду доцільно розташовувати внизу або вгорі діаграми. Це можна зробити на третьому кроці майстра або скористатись командою з меню **Диаграмма Параметры диаграммы → Легенда**.

У разі друкування діаграми на чорно-білому принтері слід пам'ятати, що кольорові елементи діаграми друкуватимуться відтінками сірого кольору. Тому доцільніше створювати діаграму, усі елементи якої будуть забарвлені у відтінки сірого кольору. Кілька типів таких діаграм є серед нестандартних типів вибрати один з них можна командою **Диаграмма → Тип диаграммы → Нестандартные**.

Створення власного нестандартного типу діаграм. Якщо діаграму побудовано і потрібно створити ще кілька діаграм такого самого вигляду, доцільно зберегти тип побудованої діаграми як нестандартний. Для цього необхідно активізувати діаграму, тип якої потрібно зберегти → **Диаграмма → Тип диаграммы → вкладка Нестандартные → опція дополнительные** → кнопка **Добавить** → ввести назву нового типу в поле *Имя*; у разі потреби можна додати опис нового типу в полі *Описание*.

Питання для самоперевірки

1. Для чого використовують діаграми?
2. Основні типи діаграм.
3. Основні елементи діаграм.
4. Опишіть кроки майстра діаграм.
5. Як редагують діаграми?
6. Коли діаграму розміщують на одному аркуші з таблицею?
7. Чи доцільно створювати кольорові діаграми? Відповідь обґрунтуйте.
8. Призначення маркерів даних.
9. Як побудувати діаграму за даними, розташованими в несуміжних діапазонах?
10. Типи нестандартних діаграм.

4.6. Робота з базами даних (списками)

Організувавши табличні дані у формі *списку* (бази даних), користувач одержує можливість виконувати в ЕТ дії, подібні до процедур обробки баз даних у системах керування базами даних (наприклад, введення та редагування даних через спеціальні вікна, сортування та відбір даних, групування та підбиття підсумків). Зазвичай таблиця-список суттєво відрізняється

від баз даних у фахових системах керування базами даних, але наявність спеціальних команд і функцій для керування такою "базою" істотно спрощує роботу й розширює можливості обробки даних. Зауважимо, що крім зазначених можливостей ЕТ підтримують обмін даними із системами керування базами даних, уможливаючи читання або зберігання даних у традиційних форматах "справжніх" баз даних.

Створення списку. Список (таблиця бази даних)— це таблиця на робочому аркуші, побудована за принципами бази даних. Призначений для зберігання великих обсягів структурованої інформації. Список представляється у формі двувимірної електронної таблиці - відношень (relation).

Структурними компонентами списку є *записи, поля* і *заголовки*. Кожний рядок, крім першого, містить окремий *запис списку* (набір даних), кожний стовпчик є *полем даних* і містить однотипну інформацію. Перший рядок списку містить *назви полів*.

Необхідно уникати порожніх рядків і стовпчиків всередині списку. Максимальний розмір списку обмежений хіба що розміром робочого аркуша.

ЕТ розпізнає списки автоматично. Перед виконанням дій зі списком достатньо активізувати будь-яку клітинку всередині списку. Вміст першого рядка ЕТ вважає іменами полів, які він не опрацьовує, як інші дані. Якщо перед обробкою списку виділити окрему частину списку, то ЕТ вважатиме списком тільки виділений діапазон клітинок.

Дії зі списками: організація введення даних або створення; перегляд даних; сортировка та фільтрація даних; підведення ітогів.

Як бачимо, створити список можна, використовуючи традиційні засоби роботи з таблицями. Задавши імена полів і порядок їх розміщення, визначивши ширину стовпчиків і параметри форматування (насамперед для дат, часу, числових значень), можна вводити записи списку. Список може також містити значення полів, отриманих у результаті обчислення формул.

При створенні списків доречно скористатися такими порадами.

Бажано розміщувати список на окремому робочому аркуші, де не повинно бути більше нічого. Якщо це неможливо, то список потрібно відокремити від інших даних принаймні одним порожнім рядком та одним порожнім стовпчиком.

Для виділення рядка з іменами полів використовують можливості форматування у вигляді шрифтів, способів їх написання та рамок. Не варто вставляти окремі рядки з дефісів, знаків рівності або інших символів, тому що Excel може сприйняти таке "підкреслення" за дані.

Якщо потрібно зазначити довге ім'я поля для стовпчика з "вузькими" даними, можна записати ім'я в кількох рядках однієї клітинки або змінити орієнтацію тексту в клітинці.

Для роботи з великими списками, дані яких не відображаються одночасно в робочому вікні, використовують команду **Окно → Закрепить области для фиксирования областей** або **Окно → Разделить** для поділу вікна на підвікна.

Присвоєння імені діапазону клітинок, що містить список, спрощує роботу зі списком, особливо коли він великий.

Використання форми даних (тільки у Excel). Хоча для додавання, видалення та редагування даних списку можна використовувати традиційні можливості програми, Excel для роботи зі списками додатково надає користувачеві стандартну екранну форму даних. Форма даних не тільки пропонує певні зручності при внесенні змін у список, а й дає змогу здійснювати відносно простий відбір даних за зазначеним критерієм, надає інформацію про кількість записів і номер поточного запису. У вікні форми даних можна використовувати до 32 полів списку.

Для виведення на екран вікна форми даних достатньо розмістити активну клітинку в діапазоні списку і скористатися командою **Данные □ Форма...** (Рис. 4.12). Якщо перед викликом вікна форми даних був виділений певний діапазон клітинок, то у вікні форми відображаються тільки поля і записи, включені в діапазон виділення. ЕТ автоматично розпізнає імена полів списку.



Рис.4.12. Введення баз даних у форму (Excel)

Для керування вікном форми даних можна використовувати відповідні кнопки та лінійку прокручування. Наприклад, скориставшись кнопками **Добавить** і **Удалить**, можна ввести новий або видалити поточний запис. За допомогою кнопки **Вернуть** можна скасувати зміни, внесені в поточний запис. Скориставшись кнопкою **Критерии**, можна ввести умову для пошуку та відбору записів списку. При записуванні критеріїв можна використовувати:

- традиційні відношення порівнянь ("=", "<", ">", "<>", "<=", ">=");
- константи відповідних типів;
- для текстових полів крім звичайних символів символи шаблону "*" (для заміни довільної кількості символів) і "?" (для заміни одного символу).

Критерії можна застосовувати також до полів для обчислення. При переміщенні у списку кнопками **Далее** та **Назад** враховується заданий критерій відбору записів. Для видалення заданого критерію і відображення всіх записів можна скористатися кнопкою **Очистить**. Для переміщення між полями та записами списку можна також використовувати комбінації клавіш.

Сортування списків. Користувач може відсортувати (упорядкувати) список за кількома полями, застосовуючи алфавітний (лексикографічний), числовий, хронологічний або власний порядок у відповідних полях. Для кожного з полів сортування можна зазначати незалежно тип впорядкування, що збільшується або зменшується. При цьому можна розрізняти в разі потреби великі та малі літери.



Для роботи з базами даних використовується команда головного меню **Данные**. Для сортування списку достатньо розмістити активну клітинку в діапазоні списку і скористатися командою **Данные → Сортировка...** програма автоматично визначає розмір списку, рядок з іменами полів (який не треба переміщувати) і відкриває вікно діалогу *Сортировка диапазона*, де обирають до трьох полів для сортування з урахуванням їх взаємних пріоритетів та порядку, що збільшується чи зменшується. При сортуванні для першого (найголовнішого) поля можна враховувати порядок, що визначається власним списком користувача. Вибрати потрібний власний список можна у вікні діалогу *Параметри сортировки*, відкривши його натисканням кнопки **Параметры...**

Якщо результат сортування потрібно скасувати, можна скористатися командою **Правка → Отменить Сортировка** або комбінацією клавіш **Ctrl+Z**.

Крім того, доцільно зважати й на такі поради.

Для швидкого відновлення початкового порядку сортування записів у списку після застосування різноманітних складних сортувань можна до виконання цих дій створити додаткове поле з номерами записів і включити його у список. Після цього для відновлення початкового порядку достатньо відсортувати список за цим полем.

Для сортування записів списку за більш як трьома полями можна скористатися послідовністю з кількох сортувань, спочатку здійснивши впорядкування за найменш важливими полями, потім за наступними щодо важливості полями; останній крок сортування здійснює впорядкування за найважливішими полями.

Для сортування за значеннями тільки одного поля можна також скористатися кнопками Сортировка по возрастанию  та Сортировка по убыванию  панелі інструментів Стандартная.

Для сортування тільки частини списку перед звертанням до команд виділяють потрібний діапазон. Але включення в діапазон виділення не всіх полів списку спричинить переміщення тільки частини записів. Тому внутрішні зв'язки інформації в записах буде зруйновано, а список пошкоджено.

При сортуванні списків, що містять формули, потрібно дотримуватись простих правил:

- у формулах посилання на клітинки (адреси), що належать до полів одного запису, оформлювати як відносні;
- посилання на клітинки поза списком оформлювати у формулах тільки як абсолютні;
- уникати у формулах посилань на клітинки, що належать до полів з інших записів.

Відбір даних засобами Автофільтра. За допомогою фільтрів, що вбудовані в ЕТ, можна відібрати та відобразити тільки ті записи, які задовольняють задані критерії (умови). Це спрощує процес пошуку потрібної інформації, редагування та видалення записів, допомагає аналізувати дані. ЕТ надають дві команди для фільтрації даних:

- для простих критеріїв — **Автофільтр**;
- для складних критеріїв — **Расширенный фильтр**.

Після фільтрації відібрані дані можна копіювати традиційними методами (у разі потреби відповідну частину полів запису) в інше місце робочої книги.

Можливостями **Автофільтра** можна скористатися, розмістивши активну клітинку всередині списку і скориставшись командою **Данные→Фильтр→Автофильтр** (Рис.13). Кнопки зі стрілкою, що з'являються справа від кожного імені поля, дають змогу відкрити перелік значень поля і накласти обмеження, створивши критерій відбору за кількома полями. У результаті відбору відображаються тільки ті записи, що відповідають заданому критерію, а рядок стану містить повідомлення про кількість знайдених записів.

Список із запропонованими для вибору значеннями може бути довгий. Для швидкого переходу до певного елемента списку можна ввести початкові літери.

Елемент **Все** списку дає змогу зняти обмеження щодо поля.

За допомогою елемента **Первые 10...** користувач може відібрати певну кількість записів з найбільшими або найменшими значеннями в числовому полі. Вікно діалогу *Наложение условия по списку* дає змогу вибрати **максимальні** або **мінімальні** значення, а також встановити числове або процентне обмеження на кількість потрібних записів.

Можна задавати й складніші критерії, ніж перевірка рівності. Скориставшись елементом списку *Условие...*, можна відкрити вікно діалогу *Пользовательский автофильтр* і з'єднати за допомогою логічних операторів окремі умови відбору записів за одним полем (з будь-якими відношеннями порівняння). При визначенні критеріїв за текстовими полями можна використовувати також символи шаблону "?" та "*".

Застосувати автофільтр можна тільки до одного списку на робочому листі.

Для відображення всіх записів списку можна скористатись командою **Данные→Фильтр→Отобразить все**. Для відміни (деактивації) автофільтра потрібно повторно скористатись командою **Данные→ Фильтр→Автофильтр**.

Якщо перед звертанням до команди автофільтра виділити частину списку, то можливості відбору будуть надані тільки із записів і за полями, що входять у діапазон виділення.

Використання Расширенного фильтра. Для фільтрації списків програм ЕТ крім команди *Автофильтр* пропонує користувачам команду *Расширенный фильтр*, яку використовують для створення складних умов відбору даних (Рис. 4.13).

Команда *Расширенный фильтр* має додаткові можливості порівняно з командою *Автофильтр*, а саме:

- задавати для кількох стовпців умови, з'єднані логічним оператором ИЛИ.;
- задавати три або більше умов для конкретного стовпця з використанням хоча б одного логічного оператора ИЛИ;
- задавати обчислювані умови.

Крім того, команду *Расширенный фильтр* використовують для відбору рядків зі списку за деякою умовою та копіювання відібраних рядків в іншу частину робочого аркуша.

Команда *Расширенный фильтр* вимагає визначення умов відбору рядків в окремому діапазоні клітинок робочого аркуша. Діапазон умов рекомендується задавати в порожніх рядках над списком.

Для використання розширеного фільтра до створеного засобами програми списку необхідно виконати такі дії.

Вставити кілька порожніх рядків у верхній частині робочого аркуша над списком (щоб вивільнити місце для діапазону умов).

Задати діапазон умов відбору рядків, який повинен містити щонайменше два рядки. У верхній рядок вводять один або кілька заголовків стовпців, у другому і наступних задають умови відбору. Між значеннями умов і списком повинен бути щонайменше один порожній рядок. За винятком обчислюваних умов, заголовки в діапазоні умов повинні точно співпадати із заголовками стовпців у списку.

	A	B	C	D	E	F	G	H	I	J	K
	Ф.И.О.	Дата рождения	Возраст	Пол	Образование	Должность	Оклад		Ф.И.О.	Условия	Оклад
2	Антонович А.Б.	27.05.1955	54	ж	в	менеджер	2 000,00 грн.		И*	>1000	
3	Ганина И.В.	03.09.1989	20	ж	ср	лаборант	700,00 грн.				
4	Гресь О.П.	11.02.1970	39	ж	в	программист	1 650,00 грн.				
5	Гришук А.Д.	21.10.1991	18	ж	н/ср	лаборант	600,00 грн.				
6	Иваненко И.А.	12.04.1991	18	м	н/ср	лаборант	600,00 грн.				
7	Иванов И.И.	01.07.1954	55	м	в	программист	1 700,00 грн.				
8	Иванова П.Н.	01.12.1965	44	ж	в	бухгалтер	1 800,00 грн.				
9	Ковалев К.К.	12.08.1963	46	м	в	зав.лаб	1 500,00 грн.				
10	Леонидов В.В.	15.06.1980	29	м	ср	программист	1 000,00 грн.				
11	Леонов П.Р.	22.07.1955	54	м	ср	лаборант	800,00 грн.				
12	Петренко П.Б.	05.12.1963	46	м	в	программист	1 550,00 грн.				
13	Петров П.П.	05.12.1954	55	м	в	менеджер	1 350,00 грн.				
14	Сидоров С.С.	06.01.1950	59	м	в	менеджер	2 150,00 грн.				
15	Федоров Ф.Ф.	15.03.1975	34	м	в	программист	2 050,00 грн.				
16	Результаты расширенного фильтра										
17	Ф.И.О.	Дата рождения	Возраст	Пол	Образование	Должность	Оклад				
18	Иванов И.И.	01.07.1954	55	м	в	программист	1 700,00 грн.				
19	Иванова П.Н.	01.12.1965	44	ж	в	бухгалтер	1 800,00 грн.				

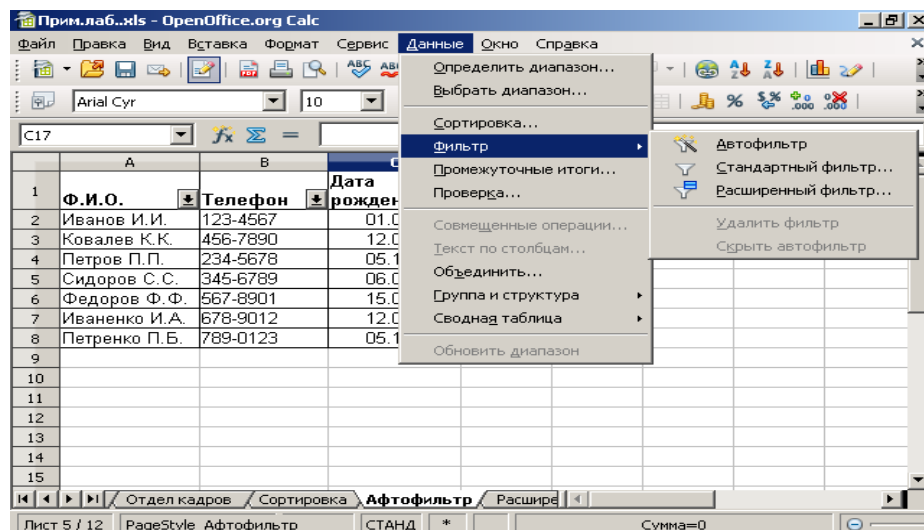


Рис.4.13. Використання інструмента Расширенный фильтр

У діапазоні умов можна ввести будь-яку кількість умов.

Програма сприймає їх за таким правилом:

- умови одного рядка вважаються з'єднаними логічним оператором **И** (тобто всі умови повинні виконуватись одночасно);
- умови у двох рядках вважаються з'єднаними логічним оператором **ИЛИ** (тобто достатньо виконання хоча б однієї умови).

Для того щоб знайти дані, які відповідають одній умові в одному стовпці або іншій умові в іншому стовпці, слід ввести умови відбору *в різні рядки* діапазону умов відбору.

За наявності для одного стовпця двох і більше умов відбору треба ввести їх безпосередньо одна під одною в суміжні рядки.

Для того щоб знайти дані, які відповідають одному з кількох наборів умов, кожний з яких містить умови більше як для одного стовпця, необхідно ввести ці умови відбору в окремі рядки. Для того щоб показати рядки, в яких на елементи одного стовпця одночасно накладається кілька умов, необхідно включити в діапазон умов кілька стовпців з однаковими заголовками.

Задаючи текстові умови, слід дотримуватися таких правил:

- одна літера означає, що пошуку підлягають усі значення, що починаються із зазначеної літери;
- символ ">" або "<" означає, що пошуку підлягають значення, що за абеткою стоять після введеного текстового значення або перед ним;
- формула *=текст* означає, що пошуку підлягають значення, які точно співпадають з введеним рядком символів *текст*.

Після введення діапазону умов відбору слід виконати команду **Данные → Фильтр → Расширенный фильтр**.

У вікні цієї команди треба ввести в поле *Исходный диапазон* діапазон клітинок таблиці, до якого буде застосовано розширений фільтр, а в поле *Диапазон условий* — діапазон клітинок робочого аркуша, де задаються умови відбору рядків, що містять заголовки стовпців.

Перемикач *Обработка* слід встановити в положення *Фильтровать список на месте*.

У результаті виконання команди *Расширенный фильтр* приховуються всі рядки, що не задовольняють введені умови. Номери відібраних рядків виділяються синім кольором.

Умовою відбору може бути *обчислене значення, що є результатом виконання певної формули*. При використанні обчислюваної умови слід дотримуватися таких правил:

- заголовок над обчислюваною умовою повинен відрізнятися від будь-якого іншого заголовка стовпця у списку. Заголовок умови може бути порожній або містити довільний текст;
- посилання на клітинки, що перебувають за межами списку, повинні бути абсолютними;
- посилання на клітинки списку повинні бути відносними (за одним винятком, який розглянемо далі).

Копіювання відфільтрованих рядків в іншу частину робочого аркуша. За допомогою вікна діалогу команди *Расширенный фильтр* можна встановити режим копіювання відібраних рядків в іншу частину аркуша замість виведення відфільтрованого списку. Щоб скопіювати рядки, треба перемикач *Обработка* встановити в положення *Скопировать результат в другое место*, а в полі *Поместить результат в диапазон* зазначити адресу діапазону, куди потрібно скопіювати інформацію.

Режим *Только уникальные записи* у вікні команди *Расширенный фильтр* додає додатковий фільтр до заданих умов відбору. Він приховує рядки, що повторюються.

Проміжні ітоги. Часто буває потрібним знати проміжні підсумки для тих або інших таблиць. Для великих таблиць зручно користуватися командою **Данные → Итоги...** (Рис. 4.14). Проміжні підсумки розраховуються для полів, які мають значення, що повторюються. Такими підсумками можуть бути сума, кількість, мінімальне, максимальне значення тощо. Перед виконанням цієї операції БД має бути відсортована за значеннями поля, які повторюються.

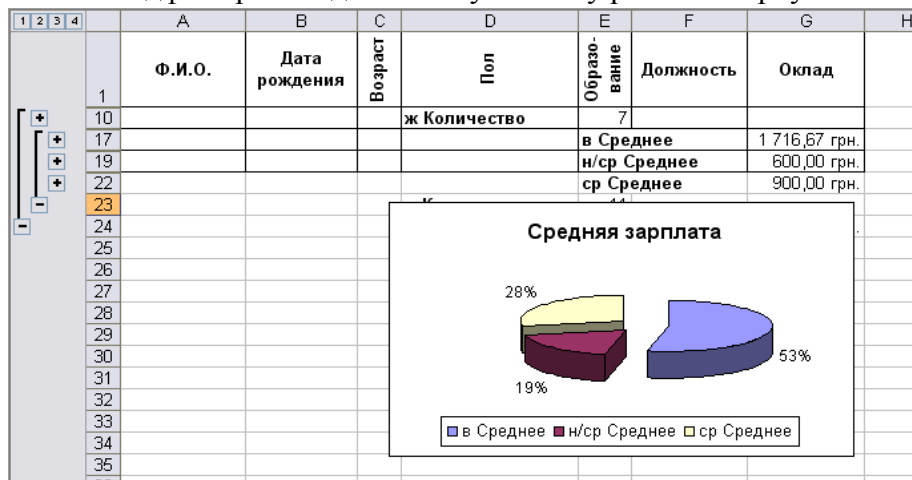
Операції підсумовування є різні: сума, максимум, мінімум, середнє значення, відхилення від норми тощо.

Крім підсумків формується структура, яка узгоджена з угрупованням даних для обчислення проміжних і загальних підсумків. Знак «-» означає, що можна сховати рядки, що відносяться до групи, залишивши тільки проміжний або загальний результат, клацнувши по «-», перетворюється в «+».

Для вилучення розрахованих проміжних підсумків слід установити курсор в БД й активізувати команди **Данные → Итоги... → Убрать все**.

Питання для самоперевірки

1. Що таке список в Excel?
2. Як створити потрібний список?
3. Які можливості надає користувачу форма даних?
4. Як скасувати внесені в запис зміни?
5. Як відсортувати список за власним порядком сортування?
6. Як забезпечити можливість відновлення початкового порядку записів списку?
7. Можливості автофільтра.
8. Можливості команди Расширенный фильтр.
9. Що таке обчислювані умови і як їх задають?
10. Як скопіювати відфільтровані дані в іншу частину робочого аркуша?



Прим.лаб...xls - OpenOffice.org Calc

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Аrial Cyr 10 Ж К Ч

НЗ

	A	B	C	D	E	F	G	H
1	Ф.И.О.	Телефон	Дата рождения	Возраст	Оклад			
2	Иванов И.И.	123-4567	01.07.1954	55	100			
3	Иванов И.И.	123-4567	01.07.1954	55	700			
4	Иванов И.И. Среднее				400			
5	Ковалев К.К.	456-7890	12.08.1963	46	500			
6	Ковалев К.К. Среднее				500			
7	Петров П.П.	234-5678	05.12.1954	55	350			
8	Петров П.П. Среднее				350			
9	Сидоров С.С.	345-6789	06.01.1950	59	150			
10	Сидоров С.С. Среднее				150			
11	Федоров Ф.Ф.	567-8901	15.03.1965	44	50			
12	Федоров Ф.Ф.	567-8901	15.03.1965	44	150			
13	Федоров Ф.Ф. Среднее				100			
14	Иваненко И.А.	678-9012	12.04.1966	43	600			
15	Иваненко И.А. Среднее				600			

Итоги2 / БД1 / БД Сортировка / Лист4 / Лист2

Лист 8 / 12 PageStyle_Итоги2 СТАНД * Сумма=0

Рис.4.14. Отримання підсумкових результатів у базах даних

4.7. Аналіз даних

У ЕТ є спеціальні можливості для аналізу табличних даних, до них можна віднести **Подбор параметров, Таблицы подстановки, Сценарии, Поиск решения**, для великих таблиць – **Сводные таблицы, Консолидация**.

Подбор параметра.

Цей інструмент використовується для рішення задач аналізу даних, особливо має важливе значення для розв'язування задач зворотнього аналізу. Якщо треба розв'язати рівняння, наприклад $f(x)=a$, де $f(x)$ деяка функція, яка повинна набувати певного значення a шляхом зміни (перебіру) значень одного з залежних параметрів x . Процес рішення є ітераційним, тобто на точність рішення впливає кількість ітерацій та відносна погрешність, ці параметри можна встановити **Сервис → Параметры → вкладка Вычисления** (Рис.15).

Для рішення задачі треба в будь-яку клітинку занести формулу $=f(x)$, вибрати цю клітинку і виконати команду **Сервис → Подбор параметра...** → в діалоговому вікні треба заповнити три поля:

- адресу клітинки з формулою (вона буде введена у поле автоматично, якщо клітинка з формулою була вибрана);
- бажане значення виразу у формулі, тобто a ;
- адресу клітинки (*изменяемой ячейки*), де і отримаємо шуканий результат.

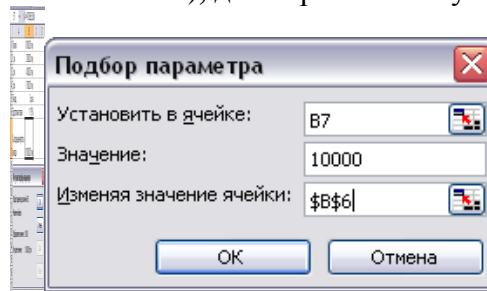


Рис.4.15. Работа инструмента Подбор параметра (Excel)

Таблица подстановки.

Інструмент аналізу **Таблица подстановки** дає змогу провести аналіз одночасно по декільком параметрам функції і в математичному сенсі є зворотною до задач **Подбора параметра**, тому що визначає результати функції при різних значеннях змінних.

	B2		fx	=A2^2
	A	B	C	
1	x	x^2	sinx	
2		0	0	
3	3	9	0,14112	
4	5	25	-0,95892	
5	6	36	-0,27942	

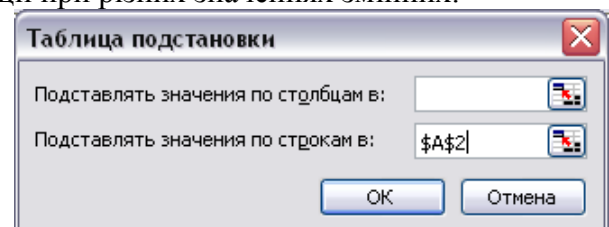


Рис.4.16. Результат використання Таблицы подстановки (Excel)

Для цього спочатку визначають місце під таблицю (**A2:C5**), клітинку **A2** залишають порожньою, тобто це місце шуканого значення x . У діапазоні **A3:A5** вводять різні варіанти x , в **B2** та **C2** записують відповідні функції. Виокремлюють діапазон **A2:C5** та обирають команди **Данные → Таблица подстановки...** (Рис.16). У вікні у відповідному полі встановити адресу шуканої клітинки **A2**. У результаті з'явиться таблиця з результатами підстановки різних значень змінних у одну чи декілько формул.

Сценарії

Сценарії — це набори значень параметрів і значень залежних величин, які подають на екран у зручному для аналізу і прийняття рішень вигляді, а також оформлюються у вигляді звітів. Множену вхідних значень можна зберегти під вказаним іменем, а потім прослідкувати,

як вхідні значення впливають на інші значення. Для використання цього інструмента треба скористатися **Диспетчером сценариев**, спочатку виділити діапазон значень, що змінюються, далі **Сервис** → **Сценарии...** → кнопка **Добавить...** (Рис.17).

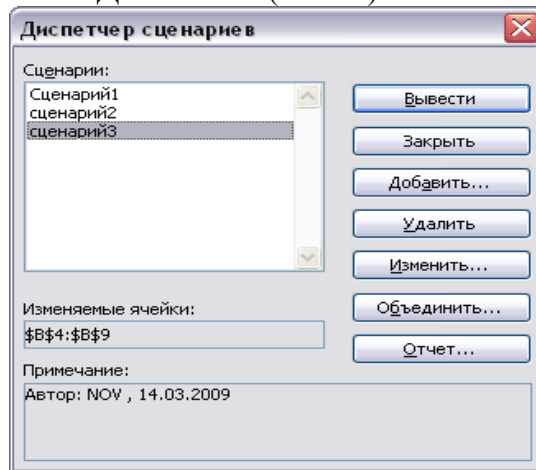


Рис.4.17. Вікно Диспетчера сценариев (Excel)

Потім у відповідних діалогових вікнах дати назву сценарію, обрати діапазон «*изменяемых ячеек*», призначити їм значення та обрати вигляд звіту.

Звіт може бути наведений у вигляді *структури* (Рис.18) або *зведеної таблиці* (Рис.19).

Структура сценария				
Текущие значения:		Сценарий1	сценарий2	сценарий3
Автор: NOV, 14.03.2009		Автор: NOV, 14.03.2009	Автор: NOV, 14.03.2009	Автор: NOV, 14.03.2009
Изменяемые:				
\$B\$4	3	3	3	3
\$B\$5	1	1	2	3
\$B\$6	2	2	2	1
\$B\$7	3	3	3	3
\$B\$8	3	3	1	1
\$B\$9	5	5	5	5
Результат:				
\$D\$10	12829,43	12829,43	15884,93	18848,55

Примечания: столбец "Текущие значения" представляет значения изменяемых ячеек в момент создания итогового отчета по Сценарию. Изменяемые ячейки для каждого сценария выделены серым цветом.

Рис.4.18. Приклад Листа Структура сценария

A1		
1	A	B
1	\$B\$4:\$B\$9 на (Все)	
2		
3	\$D\$10	
4	\$B\$4:\$B\$9	Итог
5	Сценарий1	12829,425
6	сценарий2	15884,925
7	сценарий3	18848,55
8		
9		

Рис.4.19. Приклад Листа зведеної таблиці по сценарію

Поиск решения

Інструмент **Поиск решения** використовується в задачах оптимізації, коли необхідно знайти значення параметрів, при яких рішення набувало оптимального значення (мінімального або максимального), при цьому ці значення повинні задовольняти деяким обмеженням.

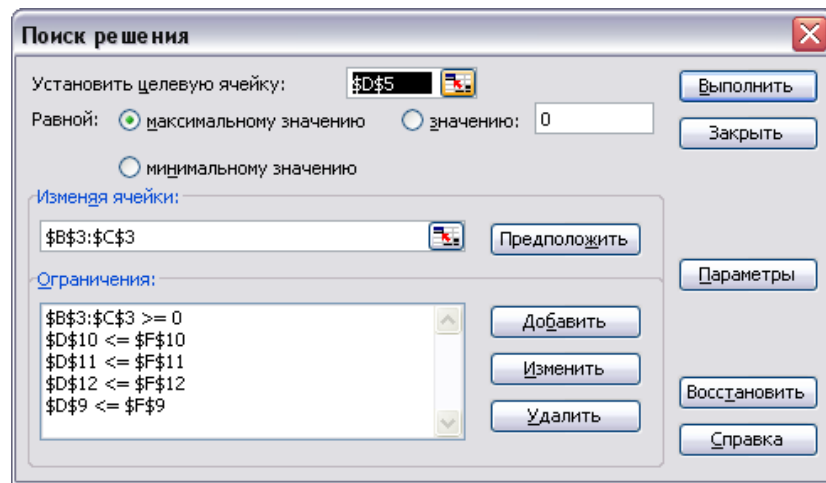


Рис.4.20. Вікно інструмента Поиск решения (Excel)

Для того щоб скористатися цим інструментом, потрібно визначити деякі терміни.

Целевая ячейка – це клітинка, в якій розміщується формула, яка описує цільову функцію, для якої треба знайти максимальне, мінімальне чи задане значення.

Изменяемые ячейки – це клітинки, в яких знаходяться значення параметрів у вигляді або постійних чисел або формул, від яких зажить значення «целевой ячейки».

Поиск решения підбирає значення «изменяемых ячеек» доти, поки не буде знайдено рішення.

Ограничение – це умова, накладена на клітинку. «Ограничения» можуть бути накладені на будь які клітинки, цільову клітинку та «изменяемые ячейки». «Ограничения» визначають область існування цільової функції. Тому їх неправильний вибір робить задачу невирішуючою або рішення може бути не вірним.

Спочатку обирають клітинки для майбутніх «изменяемых ячеек», та в будь яку клітинку вводять формулу «целевой функции» далі визивають **Сервис → Поиск решения...** (Рис.20).

В вікні встановлюють адреси «целевой ячейки», «изменяемых ячеек», визначають яке треба отримати рішення - максимальне чи мінімальне.

Послідовно вводять «ограничения» скориставшись кнопкою **Добавить** у діалоговому вікні.

Змінювати та видаляти «ограничения» можна з допомогою кнопок **Изменить**, **Удалить**.

Кнопкою **Выполнить** запустити на виконання.

Отримані результати будуть виведені на лист в обрані відповідні клітинки (Рис.21).

Результати рішення для подальшого використання можна зберегти трьома способами:

- сохранить найденное решение или восстановить исходные значения на рабочем листе;
- сохранить решение в виде сценария;
- просмотреть любой из встроенных отчетов.

	A	B	C	D	E	F	G
1		Переменные					
2	Имя	X1	X2				
3	Значение	0,00	8,00	ЦФ			
4				Значение	Направ.		
5	Козф. ЦФ Cj		3	8	64,00	Max	
6					=СУММПРОИЗВ(B5:C5;B3:C3)		
7		Ограничения Aij					
8	Вид			Левая ч.	Знак	Правая ч. Bi	
9	Огран.1	10	70	560	<=	560	
10	Огран.2	20	50	400	<=	400	
11	Огран.3	300	400	3200	<=	5600	
12	Огран.4	200	100	800	<=	3400	
13							
14					=СУММПРОИЗВ(B11:C11;B3:C3)		
15							

Рис.4.21. Приклад використання Поиска решения (Excel)

Зведені таблиці

Для більш ефективного аналізу даних великих таблиць та таблиць, які мають різні повторення значень у стовпцях та рядках, в ЕТ використовується такий інструмент, як **зведені таблиці**.

Зведеною називається спеціальна таблиця, яка підсумовує інформацію з окремих полів списку або бази даних. Зведені таблиці дають змогу представляти великий обсяг інформації у стислому та зручному вигляді підсумкових даних та автоматично розміщувати проміжні та загальні ітоги у таблицю. Зведені таблиці є «трьохвимірною», тому що до стовпців, рядків додається ще атрибут – сторінка.

Створення **зведеної таблиці** відбувається за допомогою **Мастера сводных таблиц** (Рис.22) командою **Данные → Сводная таблица (Ooo Calc Данные → Сводная таблица → Запустить...)**

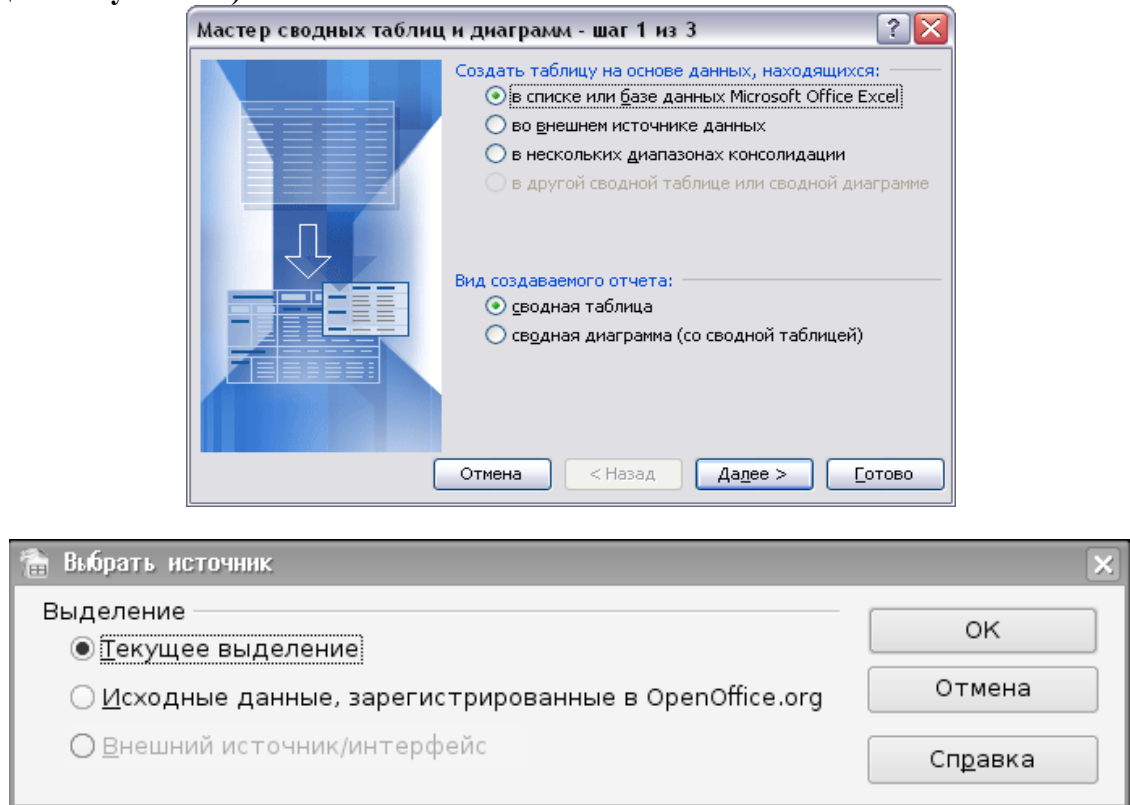


Рис.4.22. Вікно Мастера сводной таблицы (Excel та Calc)

При створенні зведеної таблиці за допомогою майстра можна задати потрібні поля, структуру таблиці (її макет), тип обчислень, що виконуються. У разі потреби після побудови таблиці можна приховувати або відображати її структурні рівні. Розглянемо основні етапи роботи із зведеною таблицею.

На першому етапі потрібно зазначити, на основі яких даних створюється таблиця. Як правило, це список, тому залиште перемикач *в списке или в базе данных* встановленим.

Друге вікно майстра, в якому пропонується ввести діапазон, що містить вихідні дані.

На третьому етапі майстра зведених таблиць формується *макет* (Рис.23) нової зведеної таблиці, створюється її структура та визначаються її функції. Для цього натисніть кнопку **Макет** і перемістіть кнопки з назвами полів у потрібну область рядків, стовпців або даних чи сторінок. Зони *Строка*, *Столбец*, *Далее* повинні мати відповідне поле. Інформація в зоні *Далее*, як правило, обробляється математично. Тому в ній розміщують числові поля. При переміщенні кнопки поля в область *Далее* програма вказує тип операції, яку здійснить з даними. За замовчуванням обчислюється сума. Для зміни типу операції слід перемістити кнопку **Поля в**

область **Данные** і двічі клацнути лівою кнопкою миші на ній — відкриється вікно, в якому можна вибрати потрібну функцію обробки даних. Заповнювати область **Страница** не обов'язково.

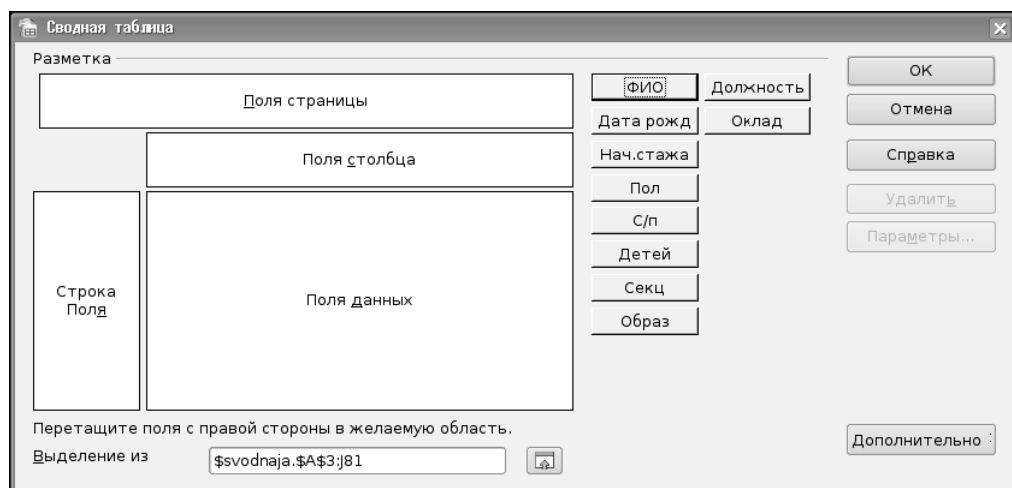
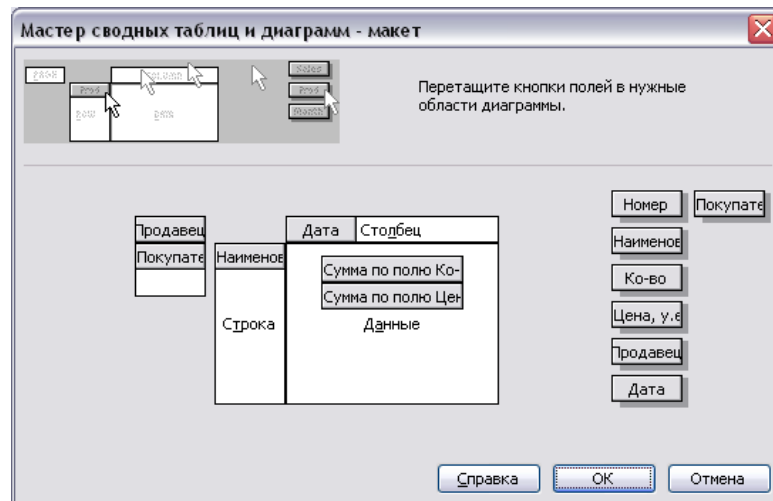


Рис.4.23. Макет звезденої таблиці (Excel, Calc)

Виберіть місце розташування звезденої таблиці (наявний або новий робочий лист) → кнопка **Параметри** і введіть назву звезденої таблиці.

Майстер створить таблицю. Відкриється також панель інструментів **Сводные таблицы**. За допомогою цієї панелі можна змінювати вигляд і параметри вже побудованої таблиці (Рис. 4.24).

До готової таблиці можна застосувати будь-який автоформат, для цього потрібно виконати **Формат** → **Автоформат** і вибрати відповідний формат таблиці. Якщо вибраний автоформат не забезпечує потрібного числового форматування, можна змінити числовий формат даних звезденої таблиці. Для цього слід виділити будь-яку клітинку таблиці, на Панелі інструментів **Сводные таблицы** → кнопка **Сводная таблица** → **Параметри поля** → у вікні діалогу кнопка **Формат**, вибрати потрібний формат → двічі натиснути кнопку **OK**.

Зміна розміщення полів. Вигляд таблиці можна змінити, перемістивши кнопки полів в інші зони, а також змінивши порядок відображення поля в області стовпців або рядків звезденої таблиці. Для видалення поля із звезденої таблиці кнопку поля треба перемістити за межі макету.

Оновлення звезденої таблиці. Зведена таблиця динамічно пов'язана з базою даних, яку було використано при її створенні. Якщо значення в базі даних змінилися, треба виконати **Данные** → **Обновить данные** або натиснути кнопку **Обновить данные** на панелі інструментів **Сводные таблицы**. Проте цей спосіб не підходить, якщо в базі даних з'являються нові рядки або стовпці.

У цьому разі необхідно повернутися до майстра зведених таблиць і зазначити новий діапазон записів, що додаються до таблиці.

12	Продавец	(Все)			
13	Покупатель	Банк			
14					
15			Дата		
16	Наименование товара	Данные	15.03.1999	25.03.1999	Общий итог
17	Компьютер	Сумма по полю Ко-во		2	2
18		Сумма по полю Цена, у.е.		1000	1000
19	Модем	Сумма по полю Ко-во	2		2
20		Сумма по полю Цена, у.е.	20		20
21	Принтер/копир/сканер	Сумма по полю Ко-во		1	1
22		Сумма по полю Цена, у.е.		663	663
23	Итог Сумма по полю Ко-во		2	3	5
24	Итог Сумма по полю Цена, у.е.		20	1663	1683
25					

	А	В	С
1	Фильтр		
2	Образ	в	
3			
4	Среднее значение - Оклад	Пол	
5	Должность	ж	Итог Результат
6	адм.зала	1 300,00р.	1 300,00р.
7	бухгалтер	1 500,00р.	1 500,00р.
8	директор	2 000,00р.	2 000,00р.
9	зав.секц	1 350,00р.	1 350,00р.
10	зам.зав.секц	1 262,50р.	1 262,50р.
11	коммерч.агент	1 600,00р.	1 600,00р.
12	продавец-2	1 030,00р.	1 030,00р.
13	товаровед	1 250,00р.	1 250,00р.
14	экономист	1 450,00р.	1 450,00р.
15	юрист	1 450,00р.	1 450,00р.
16	Итог Результат	1 380,71р.	1 380,71р.
17			

Рис.4.24. Приклад отриманої зведеної таблиці

Перейменування полів і елементів. Імена полів і елементів, що призначені майстром зведених таблиць, можна змінити, відредагувавши їх заголовки у зведеній таблиці.

Сортування елементів. Майстер зведених таблиць упорядковує елементи за їх збільшенням. Якщо треба впорядкувати елементи за їх зменшенням або за значеннями поля даних, необхідно скористатись **Данные → Сортировка**.

Відображення або приховування елементів поля рядка чи стовпця зведеної таблиці. Можна приховати певні елементи полів у зведеній таблиці. Для цього слід розгорнути список поля і встановити (зняти) прапорці для елементів, які потрібно відобразити (відповідно приховати).

Відображення або приховування елементів внутрішніх полів. Зведена таблиця для кожного елемента зовнішнього поля повторює елементи внутрішнього поля. Щоб приховати елементи внутрішнього поля, потрібно двічі клацнути на відповідному елементі зовнішнього поля. Щоб вивести на екран елементи внутрішнього поля, знову слід двічі клацнути на заголовку зовнішнього поля.

Відображення або приховування детальних даних. Можна побачити детальну інформацію, що використовується для обчислення значення поля даних, якщо двічі клацнути на цьому значенні. При цьому здійснюється копіювання детальних даних на новий аркуш книги.

Групування елементів внутрішнього поля. Майстер зведених таблиць автоматично групує елементи внутрішнього поля для кожного заголовка зовнішнього поля і в разі потреби створює проміжні підсумки для кожної групи елементів внутрішнього поля. Проте іноді зручніше

групувати елементи в інший спосіб. Щоб створити групу, слід виділити її елементи і виконати команду **Данные → Структура → Сгруппировать**.

Створення діаграми на основі зведеної таблиці. Діаграму на основі зведеної таблиці можна побудувати так само, як і для будь-яких інших типів даних робочого аркуша, скориставшись кнопкою **Мастер диаграмм** на Панелі інструментів або **Вставка → Діаграма**. Щоб досягти найкращих результатів при створенні діаграми на основі зведеної таблиці, дотримуйтесь таких рекомендацій:

- видаліть з таблиці проміжні та загальні підсумки;
- прослідкуйте, щоб таблиця мала не більше двох полів у зоні стовпців і рядків;
- приховуйте всі зайві елементи

Консолідація даних

Консолідація – це об'єднання даних з різних таблиць, розташованих у різних місцях, які мають однотипні дані в одну таблицю. Команда **Данные → Консолидация** (Рис.24) може об'єднувати інформацію з вихідних листів (до 255) в один підсумковий лист. Вихідні листи можуть розміщуватись як у тій самій книзі, що й підсумковий лист, так і в інших.

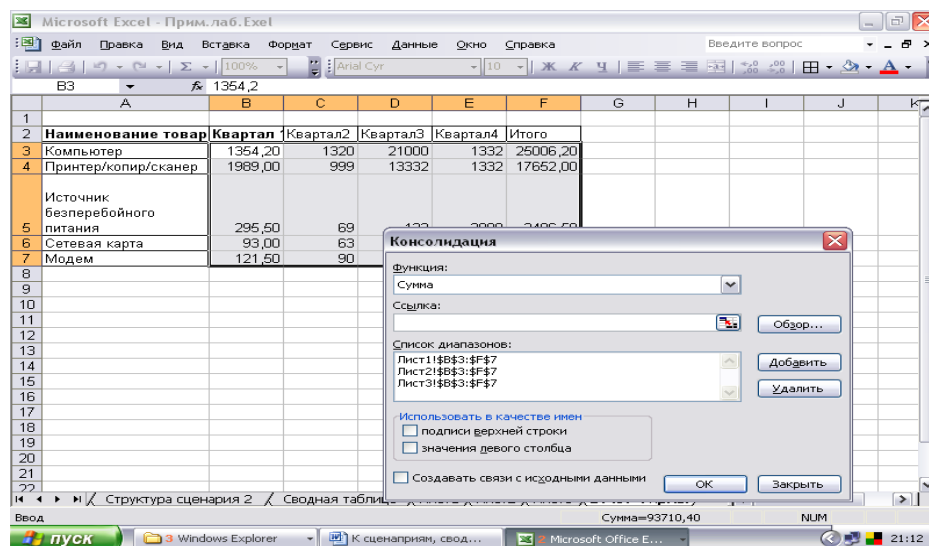


Рис.4.24. Результат консолідації (Excel)

Команду **Консолидация** можна використовувати різними способами:

- **створити зв'язки даних** у підсумковому листі з вихідними даними, щоб наступні зміни у вихідних аркушах відображалися в підсумковому аркуші;
- **консолідувати дані без створення зв'язків** — за місцем розташування і категоріями. Розглянемо останні два способи консолідації.

Консолідація за місцем розташування. У цьому разі програма збирає інформацію з однаково розміщених клітинок кожного вихідного листа. При консолідації листів використовують функцію, яку можна вибрати зі списку *Функція* у вікні команди *Консолидация*. За замовчуванням використовується функція *Сумма*, яка підсумовує дані з кожного аркуша і розміщує результат у підсумковому листі. Для консолідації вихідних листів у підсумковому листі треба виконати такі дії:

- активізувати підсумковий лист і виділити в ньому діапазон клітинок, куди заноситимуть дані, що консолідуються далі виконати **Данные → Консолидация**;
- вибрати функцію для опрацювання даних з кожного робочого листа, наприклад *Среднее* для усереднення значень з кожного робочого листа;
- опції у вікні діалогу залишити невстановленими;
- клацнути лівою кнопкою миші в полі *Ссылка* і перейти до першого робочого листа;
- виділити за допомогою миші область для початкового діапазону;
- натиснути кнопку **Добавить**;

- повторити попередні три пункти для кожного робочого листа.

Програма усереднює підсумкові дані й розміщує їх у підсумковому листі.

Консолідація за категоріями. За основу для об'єднання листів використовують заголовки стовпців або рядків. Наприклад, якщо стовпець "Січень" розміщується в одному робочому листі у стовпці B, а в іншому — у стовпці D, то все одно їх можна об'єднати. Зауважимо, що у підсумковому листі не зазначають заголовки рядків. Вони вставляються автоматично. Для консолідації даних, треба виконати такі дії:

- виділити в підсумковому листі область для розміщення даних. Вона повинна містити порожній стовпець A, щоб програма могла ввести заголовки консолідованих рядків. Ця область повинна містити стільки рядків, скільки унікальних елементів рядків (наприклад, прізвищ працівників) міститься в усіх таблицях → **Данные** → **Консолидация**;

- вибрати функцію для опрацювання даних з кожного робочого листа, наприклад *Среднее* для усереднення значень з кожного робочого листа;

- для консолідації за рядками встановити опцію *значения левого столбца* в секції *Использовать в качестве имен*;

- підсумковий лист вже має заголовки стовпців, тому їх можна опустити з посилань на вихідні робочі листи. Проте вихідні посилання повинні містити всі заголовки рядків і стовпці починаючи від стовпця заголовків рядків (наприклад, "Прізвища") і завершуючи стовпцем, що позначений іменем вибраної функції (наприклад, "Середнє"). Тому в полі *Спінка* необхідно ввести або вказати за допомогою миші потрібні вихідні діапазони і програма заповнить підсумковий аркуш. Тепер підсумковий лист містить рядки, що відповідають унікальним елементам рядків консолідованих робочих листів (у розглядуваному випадку для кожного працівника).

Створення зв'язків з початковими робочими листами. У попередніх прикладах ми консолідували дані за допомогою вибраної функції. У результаті було отримано ряд значень у підсумковому листі. При цьому зміни у вихідних листах не впливатимуть на підсумковий лист доти, поки не буде повторено консолідацію.

Якщо виконати **Данные** → **Консолидация** → опцію *Создавать связи с исходными данными* і виконати консолідацію, то в подальшому в разі зміни даних вихідних листів одразу відбуватимуться зміни в підсумковому листі.

Питання для самоперевірки

1. Для чого призначені сценарії?
2. Як побудувати сценарій?
3. Для розв'язання задач якого типу використовують Поиск решения?
4. Як запустити Поиск решения?
5. Призначення команди Консолидация.
6. Різновиди команди Консолидация.
7. Що таке зведені таблиці?
8. Як формується макет зведеної таблиці?
9. Які операції можна виконувати над даними у зведених таблицях?
10. Як побудувати діаграму на основі зведеної таблиці?

5. ЗАГАЛЬНІ ПОНЯТТЯ БАЗ ДАНИХ.

Поняття бази даних. Характеристики баз даних. Поняття СУБД. Функції СУБД. Ієрархічна модель даних. Мережева модель даних. Реляційна модель даних. Рівні моделі даних.

5.1 Бази даних

5.1.1 Поняття бази даних і СУБД

Сприйняття реального миру можна співвіднести з послідовністю різних, хоча іноді і взаємозв'язаних, явищ. З давніх часів люди намагалися описати ці явища (навіть тоді, коли не могли їх зрозуміти). Такий опис називають **даними**.

Активна діяльність по відшукуванню прийнятних способів усунування безперервно зростаючого об'єму інформації привела до створення на початку 60-х років спеціальних програмних комплексів, званих "**Системи управління базами даних**" (СУБД).

СУБД - це програмна система, що підтримує наповнення і маніпулювання даними, що представляють інтерес для користувачів при вирішенні прикладних завдань. Іншими словами, СУБД є інтерфейсом між базою даних і прикладними завданнями.

Основна особливість СУБД - це наявність процедур для введення і зберігання не тільки самих даних, але і описів їх структури. Файли, забезпечені описом що зберігаються в них даних і СУБД, що знаходяться під управлінням, почали називати банки даних, а потім бази даних (БД).

Існує велика кількість визначень поняття бази даних. Приведемо декілька визначень.

База даних - сукупність взаємозв'язано що зберігаються разом даних за наявності такої мінімальної надмірності, яка допускає їх використання оптимальним чином для одного або декількох застосувань.

База даних - це реалізована за допомогою комп'ютера інформаційна структура (модель), що відображає стан об'єктів і їх відношення.

База даних (БД) - це засіб накопичення і організації великих масивів інформації про об'єкти деякої предметної області (ПО). БД повинна відображати поточні дані про предметну область, накопичувати, зберігати інформацію і надавати різним категоріям користувачів швидкий доступ до даних. Для цього дані в базі мають бути структуровані відповідно до деякої моделі, що відображає основні об'єкти ПО, їх властивості і зв'язки між ними. БД є частиною складної системи, званої банком даних або системою баз даних (СБД).

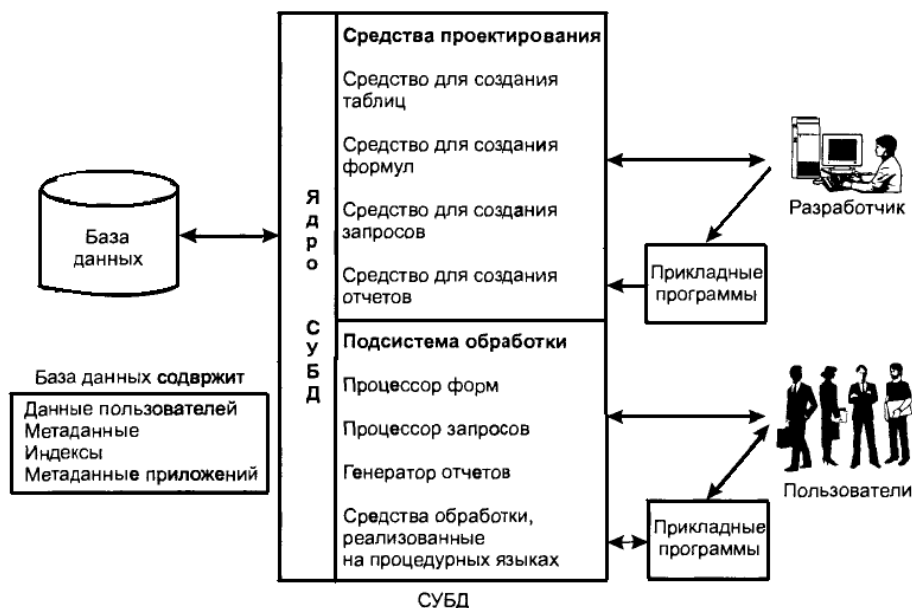


Рис. 5.1.1 Компоненты системы базы даних

Досвід використання баз даних дозволяє виділити загальний набір їх робочих характеристик:

- **повнота** - чим повніше база даних, тим ймовірніше, що вона містить потрібну інформацію (проте, не повинно бути надмірної інформації);
- **правильна організація** - чим краще структурована база даних, тим легко в ній знайти необхідні відомості;
- **актуальність** - будь-яка база даних може бути точною і повною, якщо вона постійно оновлюється, тобто необхідно, щоб база даних в кожен момент часу повністю відповідала стану об'єкту, що відображався нею;
- **зручність для використання** - база даних має бути проста і зручна у використанні і мати розвинені методи доступу до будь-якої частини інформації.

Нижче перераховані основні функції СУБД.

1. *Визначення даних* - визначити, яка саме інформація зберігатиметься в базі даних, задасть властивості даних, їх тип (наприклад, число цифр або символів), а також вказати, як ці дані зв'язані між собою. В деяких випадках є можливість задавати формати і критерії перевірки даних.
2. *Обробка даних* - дані можуть оброблятися самими різними способами. Можна вибирати будь-які поля, фільтрувати і сортувати дані. Можна об'єднувати дані з іншою, пов'язаною з ними, інформацією і обчислювати підсумкові значення.
3. *Управління даними* - можна вказати, кому дозволено знайомитися з даними, коректувати їх або додавати нову інформацію. Можна також визначати правила колективного доступу.

Вхідні до складу сучасних СУБД засоби спільно виконують наступні функції:

- опис даних, їх структури (звичайний опис даних і їх структури відбувається при ініціації нової бази даних або додаванні до існуючої бази нових розділів (стосунків); опис даних необхідний для контролю коректності використання даних, для підтримки цілісності бази даних);
- первинне введення, поповнення інформації в базі даних;
- видалення застарілої інформації з бази даних;
- коректування даних для підтримки їх актуальності;
- впорядкування (сортування) даних по деяких ознаках;
- пошук інформації по деяких ознаках (для опису запитів є спеціальна мова запитів, він забезпечує також інтерфейс між базою даних і прикладними програмами користувачів, дозволяє цим програмам використовувати бази даних);
- підготовку і генерацію звітів (засоби підготовки звітів дозволяють створювати і роздруковувати зведення по заданих формах на основі інформації бази даних);
- захист інформації і розмежування доступу користувачів до неї (деякі розділи бази даних можуть бути закриті для користувача зовсім, відкриті тільки для читання або відкриті для зміни; крім того, при многопользовательському режимі роботи з базою даних необхідно, щоб зміни вносилися коректно; для збереження цілісності даних служить механізм трансакцій при маніпулюванні даними - виконання маніпуляцій невеликими пакетами, результати кожного з яких у разі виникнення некоректності операцій "відкатуються" і дані повертаються до початкового стану);
- резервне збереження і відновлення бази даних, яке дозволяє відновити втрачену при збоях і аваріях апаратури інформацію бази даних, а також накопичити статистику роботи користувачів з базою даних;
- підтримку інтерфейсу з користувачами, який забезпечується засобами ведення діалогу (у міру розвитку і вдосконалення СУБД цей інтерфейс стає все більш дружнім; дружність

існуючих засобів інтерфейсу припускає наявність розвиненої системи допомоги (підказки), до якої у будь-який момент може звернутися користувач, не перериваючи сеансу роботи з комп'ютером і базою даних;

- захист від необдуманих дій, застережливий користувача і запобігаючу втрату інформації у разі поспішних або помилкових команд;
- наявність декількох варіантів виконання одних і тих же дій, з яких користувач може вибрати найбільш зручні для себе, відповідні його підготовці, кваліфікації, звичкам;
- ретельно продуману систему ведення людино-машинного діалогу, відображення інформації на дисплеї, використання клавіш клавіатури).

Розрізняють три типи СУБД:

- ієрархічна;
- мережева;
- реляційна.

5.1.2 Ієрархічна модель даних

Найбільш відомим і поширеним представником такої моделі даних є СУБД IMS (Information Management System) компанії IBM. Перша версія системи з'явилася в 1968 р.

Ієрархічна БД складається з впорядкованого набору дерев; точніше, з впорядкованого набору декількох екземплярів одного типу дерева. Тип дерева складається з одного «кореневого» типу запису і впорядкованого набору з нуля або більш за типи піддерев (кожне з яких є деяким типом дерева). Тип дерева в цілому є ієрархічно організованим набором типів запису. Або іншими словами, дані представляються у вигляді дерева з одним кореневим вузлом і з умовами, що кожен вузол нижче кореневого може бути пов'язаний з одним вищестоящим вузлом і з декількома нижчестоячими вузлами.

Розглянемо приклад.

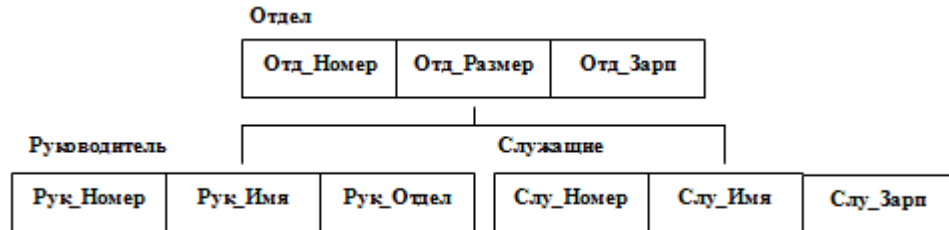


Рис.5.1.2 Приклад типу дерева

У розглянутому прикладі тип запису Відділ є предком для типів запису Керівник і Службовці, а Керівник і Службовці - нащадки типу запису Відділ. Сенс полів типів записів в основному має бути зрозумілий по їх іменам. Поле Рук_отдел типу запису Керівник містить номер відділу, в якому працює службовець, що є даним керівником (передбачається, що він працює не обов'язково в тому ж відділі, яким керує). Між типами запису підтримуються зв'язки.

Один екземпляр дерева приведенного в прикладі мав би наступний вигляд:

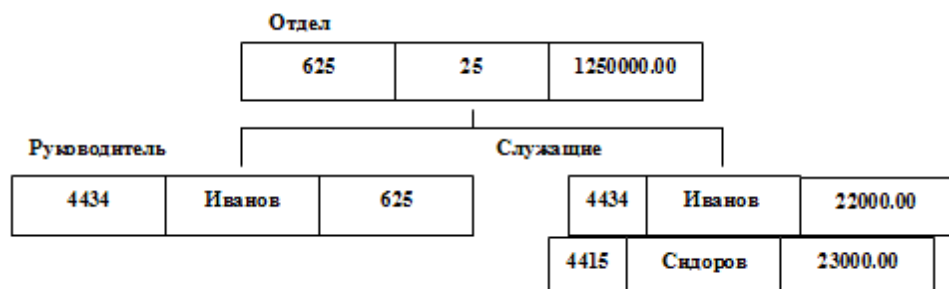


Рис.5.1.3 Приклад ієрархічної бази даних

Всі екземпляри даного типу нащадка із загальним екземпляром типу предка називаються *близнюками*. Для ієрархічної бази даних визначається повний порядок обходу дерева: зверху-вниз, зліва-направо.

У ієрархічній моделі даних автоматично підтримується цілісність посилань між предками і нащадками. Основне правило: ніякий нащадок не може існувати без свого батька.

Недоліки: якщо дані не мали деревовидної структури, то виникала маса складнощів при побудові ієрархічної моделі і бажанні добитися потрібної продуктивності.

5.1.3 Мережева модель даних

Типовим представником систем, заснованих на мережевій моделі даних, є СУБД IDMS (Integrated Database Management System), розроблена компанією Cullinet Software, Inc. і спочатку орієнтована на використання на мейнфреймах компанії IBM. Архітектура системи заснована на пропозиціях Data Base Task Group (DBTG) організації CODASYL (Conference on Data Systems Languages), яка відповідала за визначення мови програмування COBOL. Звіт DBTG був опублікований в 1971 р., і незабаром після цього з'явилося декілька систем, що підтримують архітектуру CODASYL, серед яких присутня і СУБД IDMS. В даний час IDMS належить компанії Computer Associates.

Мережевий підхід до організації даних є розширенням ієрархічного підходу. У ієрархічних структурах запис-нащадок повинен мати в точності одного предка; у мережевій структурі даних у нащадка може бути будь-яке число предків.

Мережева БД складається з набору записів і набору зв'язків між цими записами, а якщо говорити точніше, з набору екземплярів кожного типу із заданого в схемі БД набору типів запису і набору екземплярів кожного типу із заданого набору типів зв'язку.

Тип зв'язку визначається для двох типів запису: предка і нащадка. Екземпляр типу зв'язку складається з одного екземпляра типу запису предка і впорядкованого набору екземплярів типу запису нащадка. Для даного типу зв'язку L з типом запису предка P і типом запису нащадка C повинні виконуватися наступні дві умови:

- кожен екземпляр типу запису P є предком тільки в одному екземплярі типу зв'язку L;
- кожен екземпляр типу запису C є нащадком не більше ніж в одному екземплярі типу зв'язку L.

Розглянемо приклад схеми мережевої БД.

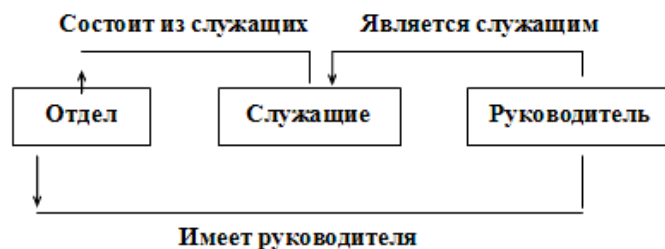


Рис.5.1.4 Приклад схеми мережевої бази даних

На рисунку показано три типи запису: *Відділ*, *Службовці* і *Керівник* і три типи зв'язку: *Складається із службовців*, *Має керівника* і *Є таким, що служить*.

У типі зв'язку *Складається із службовців* типом записи-предком є Відділ, а типом записи-потомком - Службовці (екземпляр цього типу зв'язку зв'язує екземпляр типу запису Відділ з багатьма екземплярами типу запису Службовці, відповідними всім службовцям даного відділу).

У типі зв'язку *Має керівника* типом записи-предком є Відділ, а типом записи-потомком - Керівник (екземпляр цього типу зв'язку зв'язує екземпляр типу запису Відділ з одним екземпляром типу запису Керівник, відповідним керівникові даного відділу).

Нарешті, в типі зв'язку *Є таким, що служить* типом записи-предком, є Керівник, а типом записи-потомком - Службовці (екземпляр цього типу зв'язку зв'язує екземпляр типу запису Керівник з

одним екземпляром типу запису Службовці, відповідним тому службовцеві, яким є даний керівник).

Недоліки: складність структури.

5.1.4 Реляційна модель даних

За минулі десятиліття реляційна модель розвивалася в двох напрямках. Перший напрям заклав знаменитий експериментальний проект компанії IBM System R. У цьому проекті виникла мова SQL, спочатку заснований на ідеях Кодда, але що порушує деякі принципи розпорядження реляційної моделі.

Другий напрям, починаючи з 1990-х рр., очолює Крістофер Дейт, до якого пізніше прилучився Хью Дарвен. Проте в 1990-і рр. Дейт і Дарвен прийшли до висновку, що спотворення реляційної моделі даних, властиві мові SQL, досягли настільки високого рівня, що прийшло час запропонувати альтернативу, що спирається на неспотворені ідеї Едгара Кодда і забезпечує всі можливості як SQL, так і об'єктно-орієнтованого підходу до організації баз даних і СУБД.

Назва “реляційна” (у перекладі з англійського relation - відношення) пов'язана з тим, що кожен запис в таблиці містить інформацію, що відноситься тільки до одного конкретного об'єкту.

Детальніше реляційну модель даних розглянемо пізніше.

5.1.5 Рівні моделі даних.

Проектування бази даних треба починати з аналізу наочної області і виявлення вимог до неї окремих користувачів. Проектування зазвичай доручається людині (групі осіб) - **адміністраторові бази даних (АБД)**. Їм може бути як спеціально виділений співробітник організації, так і майбутній користувач бази даних, досить добре знайомий з машинною обробкою даних.

Виділяють три рівні моделі даних:

- інфологічна;
- даталогічна;
- фізична.



Рис.5.1.5 Рівні моделей даних

Інфологічна модель описує предметну область на змістовному рівні. На першому етапі при її розробці здійснюється аналіз предметної області, вирішуваних завдань, запитів користувачів і документів, що відображають події і процеси, що протікають в ПО. Результатом цього аналізу є списки об'єктів предметної області, переліки їх властивостей або атрибутів, визначення зв'язків між об'єктами і опис структури ПО у вигляді діаграми. Для кожного з атрибутів указуються обмеження на їх можливі значення, визначувані властивостями ПО.

Такі обмеження називаються обмеженнями цілісності даних.

Інфологіческая модель об'єднує в єдине узагальнене уявлення вимоги окремих користувачів і служить засобом спілкування між ними, тому розробляється без урахування особливостей представлення даних в пам'яті ЕОМ.

Концептуальна або даталогіческая модель описує об'єкти і зв'язки ПО на формальному рівні. Її розробка ведеться на другому етапі і ґрунтується на інфологіческой моделі, отриманій на першому етапі. В процесі розробки здійснюється вибір типу моделі даних,

і визначаються її елементи. Кожна СУБД підтримує тільки одну з моделей. Вибір моделі даних і вибір СУБД тісно взаємозв'язані.

Внутрішня, або фізична, модель даних визначає спосіб розміщення даних безпосередньо на машинному носіїві, враховує розподіл даних, методи доступу і способи індексування. У сучасних прикладних програмних засобах цей рівень організації забезпечується автоматично без втручання користувача. Користувач, як правило, оперує в прикладних програмах і універсальних програмних засобах представленнями СУБД на організацію даних.

Таким чином основне завдання проектування полягає в створенні інфологічеської моделі ПО і концептуальною БД.

Контрольні питання:

1. Що таке БД?
2. Що таке СУБД?
3. Які ви знаєте функції СУБД?
4. Які існують типи СУБД? У чому їх відмінності?
5. Дати визначення інфологічеської моделі БД.
6. Дати визначення концептуальної моделі БД.

5.2. Інфологічеська модель даних "суть-зв'язок"

Основні поняття інфологічеського моделювання. Суть. Атрибут. Ключ. Зв'язок. Основні класи суті. ER-діаграми і мова інфологічеського моделювання. Чотири види зв'язків.

5.2.1 Поняття, використовувані в інфологічному моделюванні.

Модель була запропонована Пітером Ченом (Peter Chen) в 1976 р. Моделювання наочної області базується на використанні графічних діаграм, що включають невелике число різнорідних компонентів.

Мета інфологічного моделювання - забезпечення найбільш природних для людини способів збору і представлення тієї інформації, яку передбачається зберігати в створюваній базі даних. Тому інфологічну модель даних намагаються будувати по аналогії з природною мовою. Основними конструктивними елементами інфологічеських моделей є суть, зв'язки між ними і їх властивості (атрибути).

Суть - будь-який помітний об'єкт (об'єкт, який ми можемо відрізнити від іншого), інформацію про яке необхідно зберігати в базі даних. Суттю можуть бути люди, місця, літаки, рейси, смак, колір і так далі. Необхідно розрізняти такі поняття, як тип суті і екземпляр суті.

Поняття **тип суті** відноситься до набору однорідних осіб, предметів, подій або ідей, промовців як ціле. **Екземпляр суті** відноситься до конкретної речі в наборі. Наприклад, типом суті може бути МІСТО, а екземпляром - Москва, Київ і так далі.

Атрибут - поійменована характеристика суті. Його найменування має бути унікальним для конкретного типу суті, але може бути однаковим для різного типу суті (наприклад, КОЛІР може бути визначений для багатьох суті: СОБАКА, АВТОМОБІЛЬ, ДІМ і так далі). Атрибути використовуються для визначення того, яка інформація має бути зібрана про суть. Прикладами атрибутів для суті АВТОМОБІЛЬ є ТИП, МАРКА, НОМЕРНИЙ ЗНАК, КОЛІР і так далі.

Тут також існує відмінність між типом і екземпляром. Тип атрибуту КОЛІР має багато екземплярів або значень: Червоний, Синій, Банановий, Біла ніч і так далі, проте кожному екземпляру суті привласнюється тільки одне значення атрибуту.

Ключ - мінімальний набір атрибутів, по значеннях яких можна однозначно знайти необхідний екземпляр суті. Мінімальність означає, що виключення з набору будь-якого атрибуту не дозволяє ідентифікувати суть по тих, що залишилися. Для суті Розклад ключем є атрибут «Номер рейса» або набір: «Пункт відправлення», «Час вильоту» і «Пункт призначення» (за умови, що з пункту в пункт вилітає в кожен момент часу один літак).

Зв'язок - асоціювання два або більш за суті. Якби призначенням бази даних було тільки зберігання окремих, не зв'язаних між собою даних, то її структура могла б бути дуже простій. Проте одна з основних вимог до організації бази даних - це забезпечення можливості відшукування однієї суті по значеннях інших, для чого необхідно встановити між ними певні зв'язки. А оскільки в реальних базах даних нерідко містяться сотні або навіть тисячі суті, то теоретично між ними може бути встановлене більше мільйона зв'язків. Наявність такої безлічі зв'язків і визначає складність інфологічних моделей.

5.2.2 Основні класи суті.

Існують три основні класи суті: стрижньові, асоціативні і характеристичні, а також підклас асоціативної суті - позначення.

Стрижньова суть (стрижень) - це незалежна суть. Наприклад стрижнями є: "Студент", "Квартира", "Чоловіки", "Лікар", "Брак".

Асоціативна суть (асоціація) - це зв'язок виду "многие-ко-многим" між двома або більш суттями. Асоціації розглядаються як повноправна суть: вони можуть брати участь в інших асоціаціях і позначеннях точно так, як і стрижньова суть; можуть володіти властивостями, тобто мати не тільки набір ключових атрибутів, необхідних для вказівки зв'язків, але і будь-яке число інших атрибутів, що характеризують зв'язок. Наприклад, асоціація "Брак" містять ключові атрибути "Код_м", "Код_ж" і "Табельний номер чоловіка", "Табельний номер дружини", а також уточнюючі атрибути "Номер свідоцтва", "Дата реєстрації", "Место_реєстрації", "Номер запису в книгу ЗАГС" і так далі

Характеристична суть (характеристика) - це зв'язок виду "многие-к-одной" або "одна-к-одной" між двома суттями (окремий випадок асоціації). Єдина мета характеристики в рамках даної наочної області полягає в описі або уточненні деякій іншій суті.

Позначаюча суть або позначення - це зв'язок виду "многие-к-одной" або "одна-к-одной" між двома суттями і відрізняється від характеристики тим, що не залежить від суті, що позначається.

Позначення і характеристики не є повністю незалежною суттями, оскільки вони припускають наявність деякій іншій суті, яка "позначатиметься" або "характеризуватиметься". Проте вони все ж таки є окремими випадками суті і можуть, звичайно, мати властивості, можуть брати участь в асоціаціях, позначеннях і мати свої власні (нижчого рівня) характеристики. Підкреслимо також, що всі екземпляри характеристики мають бути обов'язково пов'язані з яким-небудь екземпляром суті, що характеризується. Проте допускається, щоб деякі екземпляри суті, що характеризується, не мали зв'язків.

Перевизначимо тепер стрижньову суть як суть, яка не є ні асоціацією, ні позначенням, ні характеристикою. Така суть має незалежне існування.

На закінчення розглянемо приклад побудови інфологічної моделі бази даних "Живлення", де повинна зберігатися інформація про блюда, їх щоденне споживання, продукти, з яких готуються ці блюда, і постачальників цих продуктів. Інформація використовуватиметься кухарем і керівником невеликого підприємства громадського харчування, а також його відвідувачами.

За допомогою вказаних користувачів виділені наступні об'єкти і характеристики проектованої бази:

- Блюда, для опису яких потрібні дані, що входять в їх кулінарні рецепти: номер блюда (наприклад, з книги кулінарних рецептів), назва блюда, вид блюда (закуска, суп, гарячіше і тому подібне), рецепт (технологія приготування блюда), вихід (вага порції), назва, калорійність і вага кожного продукту, що входить в блюдо.
- Для кожного постачальника продуктів: найменування, адреса, назва продукту, що поставляється, дата постачання і ціна на момент постачання.
- Щоденне споживання блюд (витрата): блюдо, кількість порцій, дата.

Аналіз об'єктів дозволяє виділити:

- *Стрижні*: Блюда, Продукти і Міста;
- *Асоціації*: Склад (пов'язує Блюда з Продуктами) і Постачання (пов'язує Постачальників з Продуктами);
- *Позначення*: Постачальники;
- *Характеристики*: Рецепти і Витрата.

5.2.3 ER- діаграми і мова інфологічного моделювання (ЯІМ)

При побудові інфологічних моделей можна використовувати мову ER-діаграмм.

У них суть зображається поміченими прямокутниками, асоціації - поміченими ромбами або шестикутниками, атрибути - поміченими овалами, а зв'язки між ними - ненапрямленими ребрами, над якими може проставлятися ступінь зв'язку (1 або буква, замінююча слово "багато") і необхідне пояснення.



Рис.5. 2.1 Позначення, використовувані в ER-діаграммах.

Мова ER-діаграмм використовується для побудови невеликих моделей і ілюстрації окремих фрагментів великих. Частіше ж застосовується менш наочна, але змістовніша **мова інфологічного моделювання (ЯІМ)**, в якому суть і асоціації представляються пропозиціями вигляду:

СУТЬ (атрибут 1, атрибут 2 , ..., атрибут n)

АСОЦІАЦІЯ [СУТЬ S1, СУТЬ S2 ...]

(атрибут 1, атрибут 2, ..., атрибут n)

ХАРАКТЕРИСТИКА (атрибут 1, атрибут 2, ...)

{ СПИСОК, ЩО ХАРАКТЕРИЗУЄ СУТЬ }

ПОЗНАЧЕННЯ (атрибут 1, атрибут 2, ...)

[СПИСОК, ЩО ХАРАКТЕРИЗУЄ СУТЬ]

де S - ступінь зв'язку, а атрибути, що входять в ключ, мають бути відмічені за допомогою підкреслення.

Для прикладу бази даних "Живлення" модель на мові ЯІМ має наступний вигляд:

Блюда (БЛ, Блюдо, Вигляд)

Продукти (ПР, Продукт, Калорійність)

Постачальники (ПОС, Місто, Постачальник) [Місто]

Склад [Блюда М, Продукти N] (БЛ, ПР, Вага (г))

Постачання [Постачальники М, Продукти N] (ПОС, ПР, Дата_п, Ціна, Вага (кг))

Міста (Місто, Країна)

Рецепти (БЛ, Рецепт) { Блюда }

Витрата (БЛ, Дата_р, Порцій) { Блюда }

ER-діаграма моделі бази даних "Живлення" наступний вигляд:

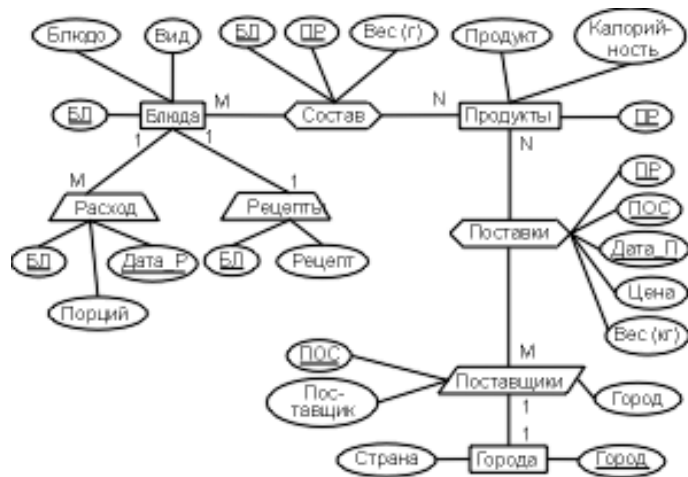
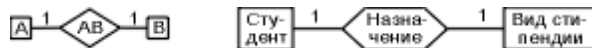


Рис.5.2.2 Инфологическая модель базы данных "Живления"

5.2.4 Види зв'язків

Між двома суттями, наприклад, А і В можливі чотири види зв'язків.

1. зв'язок **ОДИН-К-ОДНОМУ (1:1)**: у кожен момент часу кожному представникові (екземпляру) суті А відповідає 1 або 0 представників суті В:



Наприклад: студент може не "заробити" стипендію, отримати звичайну або одну з підвищених стипендій.

2. зв'язок **ОДИН-КО-МНОГИМ (1:m)**: одному представникові суті А відповідають 0, 1 або декілька представників суті В.



Наприклад: квартира може бути порожньою, в ній може жити один або декілька мешканців.

3. **МНОГИЕ-К-ОДНОМУ (M:1)**

4. **МНОГИЕ-КО-МНОГИМ (M:N)**.

Приклад. Якщо зв'язок між суттю ЧОЛОВІКА і ЖІНКИ називається БРАК, то існує чотири можливі представлення такого зв'язку:



Характер зв'язків між суттю не обмежується перерахованими. Існують і складніші зв'язки:

— безліч зв'язків між однією і тією ж суттю



(пацієнт, маючи одного лікаря, що лікує, може мати також декілька лікарів-консультантів; лікар може бути лікарем декількох пацієнтів, що лікує, і може одночасно консультувати дещо інших пацієнтів);

— тренарні зв'язки



(лікар може призначити декілька пацієнтів на декілька аналізів, аналіз може бути призначений декількома лікарями декільком пацієнтам і пацієнт може бути призначений на декілька аналізів декількома лікарями);

— зв'язки вищих порядків, семантика (сенси) яких іноді дуже складна.

Контрольні питання:

1. Чим відрізняються тип суті та екземпляр суті?
2. Що таке ключ?
3. Які три класи суті ви знаєте?
4. Для чого використовується ER-діаграмм?
5. Чим ER-діаграмм відрізняється від ЯІМ?
6. Які види зв'язків ви знаєте? Приведіть приклади?

5.3. Реляційна база даних

Основні поняття. Правила побудови реляційних баз даних. Нормалізація. Процес проектування.

5.3.1 Основні поняття, використовувані в реляційних базах даних

У 1970 р. Е.Ф. Кодд (E.f. Codd) опублікував свою статтю, в якій він застосував концепції розділу математики, званого реляційною алгеброю, до проблеми зберігання великих об'ємів даних. Стаття Кодда поклала початок руху у сфері проектування баз даних, яке привело декілька років опісля до створення реляційної моделі бази даних. Ця модель є певним способом структуризації і обробки бази даних.

Перевага реляційної моделі полягає в способі зберігання даних, який мінімізує їх дублювання і виключає певні типи помилок обробки, що виникають при інших способах зберігання даних. Дані зберігаються у вигляді таблиць.

Згідно реляційної моделі, не всі види таблиць однаково прийнятні. За допомогою процесу, званого нормалізацією, небажана таблиця може бути перетворена в дві або прийнятніші.

Введемо наступні позначення:

- **Суть** - Таблиця (іноді Файл),
- **Екземпляр суті** - Рядок (іноді Запис),
- **Атрибут** - Стовпець, Поле.

При цьому приймається, що "запис" означає "екземпляр запису", а "поле" означає "ім'я і тип поля".

Ключ або **можливий ключ** - це мінімальний набір атрибутів, по значеннях яких можна однозначно знайти необхідний екземпляр суті. Мінімальність означає, що виключення з набору будь-якого атрибуту не дозволяє ідентифікувати суть по тих, що залишилися. Кожна суть володіє хоч би одним можливим ключем. Один з них береться за **первинний ключ**. При виборі первинного ключа слід віддавати перевагу нескладеним ключам або ключам, складеним з мінімального числа атрибутів. Недоцільно також використовувати ключі з довгими текстовими значеннями (переважно використовувати цілочисельні атрибути). Первинний ключ має бути унікальним.

Реляційна база даних - це сукупність стосунків, що містять всю інформацію, яка повинна зберігатися в БД. Проте користувачі можуть сприймати таку базу даних як сукупність таблиць.

Так на рис. 3.1 показані таблиці бази даних, побудовані по інфологічеськой моделі бази даних "Живлення" рис. 2.2

Блюда

БЛ	Блюдо	Вид
1	Лобио	Закуска
2	Харчо	Суп
3	Шашлык	Горячее
4	Кофе	Десерт

Витрати

БЛ	Порций	Дата_Р
1	158	1/9/94
2	144	1/9/94
3	207	1/9/94
4	235	1/9/94
...

Продукти

ПР	Продукт	Калор.
1	Фасоль	3070
2	Лук	450
3	Масло	7420
4	Зелень	180
5	Мясо	1660
6	Томаты	240
7	Рис	3340
8	Кофе	2750

Рецепти

БЛ	Рецепт
1	Ломаную очищ
...	...

Склад

БЛ	ПР	Вес (г)
1	1	200
1	2	40
1	3	30
1	4	10
2	5	80
2	2	30
2	6	40
2	7	50
2	3	15
2	4	15
3	5	180
3	6	100
3	2	40
3	4	20
4	8	8

Постачання

Постачальники		
ПОС	Поставщик	Город
1	"Полесьє"	Київ
2	"Наталка"	Київ
3	"Хуанхэ"	Пекин
4	"Лайма"	Рига
5	"Юрмала"	Рига
6	"Даугава"	Рига

Міста	
Город	Страна
Київ	Україна
Пекин	Китай
Рига	Латвія

ПОС	ПР	Вес (кг)	Цена	Дата_П
1	6	120	0.45	27/8/94
1	3	50	1.82	27/8/94
1	2	50	0.61	27/8/94
2	2	100	0.52	27/8/94
2	5	100	2.18	27/8/94
2	4	10	0.88	27/8/94
3	1	250	0.37	24/8/94
3	7	75	0.44	24/8/94
3	8	40	2.87	24/8/94
4	3	70	1.56	30/8/94
5	5	200	2.05	30/8/94
6	6	15	0.99	30/8/94

Рис. 5.3.1 База даних "Живлення"

5.3.2 Правила побудови реляційних баз даних

Розглянемо основні правила побудови реляційних баз даних:

1. Кожна таблиця складається з однотипних рядків і має унікальне ім'я.
2. Рядки мають фіксоване число полів (стовпців) і значень (множинні поля і групи, що повторюються, недопустимі). Інакше кажучи, в кожній позиції таблиці на перетині рядка і стовпця завжди є в точності одне значення або нічого.
3. Рядки таблиці обов'язково відрізняються один від одного хоч би єдиним значенням, що дозволяє однозначно ідентифікувати будь-який рядок такої таблиці.
4. Стовпцям таблиці однозначно привласнюються імена, і в кожному з них розміщуються однорідні значення даних (дати, прізвища, цілі числа або грошові суми).
5. Повний інформаційний зміст бази даних представляється у вигляді явних значень даних і такий метод уявлення є єдиним. Зокрема, не існує яких-небудь спеціальних "зв'язків" або показників, що сполучають одну таблицю з іншою. Так, зв'язки між рядком з БЛ = 2 таблиці "Блюда" на рис. 3.1 і рядком з ПР = 7 таблиць продукти (для приготування Харчо потрібний Ріс), представляється не за допомогою показників, а завдяки існуванню в таблиці "Склад" рядка, в якому номер блюда дорівнює 2, а номер продукту - 7.
6. При виконанні операцій з таблицею її рядка і стовпці можна обробляти у будь-якому порядку безвідносно до їх інформаційного змісту. Цьому сприяє наявність імен таблиць і їх стовпців, а також можливість виділення будь-якого їх рядка або будь-якого набору рядків з вказаними ознаками (наприклад, рейсів з пунктом призначення "Париж" і часом прибуття до 12 годин).

5.3.3 Поняття універсального відношення

Припустимо, що проектування бази даних "Живлення" починається з виявлення атрибутів і підбору даних, зразок яких (частина блюд виготовлених і реалізованих 1/9/94 р.) показаний на рис. 3.2.

Цей варіант таблиці "Живлення" не є відношенням, оскільки більшість її рядків не атомарні. Атомарними є лише значення полів Блюдо, Вигляд, Рецепт (хоча він і великий), Порцій і Дата_р решта полів таблиці рис. 3.2- же множинна. Для додання таким даним форми відношення необхідно реконструювати таблицю. Найпростіше це зробити за допомогою простого процесу вставки, результат якої показаний на рис. 3.3. Проте таке перетворення приводить до виникнення великого об'єму надмірних даних.

Таблиця на рис. 3.3 є екземпляром коректного відношення. Його називають універсальним відношенням проектованої БД. У одне універсальне відношення включаються всі атрибути, що представляють інтерес, і воно може містити всі дані, які передбачається розміщувати в БД в майбутньому. Для малих БД (що включають не більше 15 атрибутів) універсальне відношення може використовуватися як відправна крапка при проектуванні БД.

При використанні універсального відношення виникає декілька проблем:

1. **Надмірність.** Дані практично всіх стовпців багато разів повторюються. Повторюються і деякі набори даних (Блюдо-Вид-рецепт, Продукт-Калорійність, Поставщик-Город-страна). Небажане повторення рецептів, деякі з яких набагато більше рецепту "Лобіо". І вже зовсім погано, що всі дані про блюдо (включаючи рецепт) повторюються кожного разу, коли це блюдо включається в меню.

2. **Потенційна суперечність (аномалії оновлення).** Унаслідок надмірності можна відновити адресу постачальника в одному рядку, залишаючи його незмінним в інших. Якщо постачальник кави повідомив про свій переїзд до Харбіну і був оновлений рядок з продуктом кави, то у постачальника "Хуанхе" з'являється дві адреси, один з яких не актуальний. Отже, при оновленнях необхідно проглядати всю таблицю для знаходження і зміни всіх відповідних рядків.

3. **Аномалії включення.** У БД не може бути записаний новий постачальник ("Нярінга", Вільнюс, Литва), якщо продукт (Огірки), що поставляється ним, не використовується ні в одному блюді. Можна, звичайно, помістити невизначені значення в стовпці Блюдо, Вигляд, Порцій і Вес (г) для цього постачальника. Але якщо з'явиться блюдо, в якому використовується цей продукт, чи не забудемо ми видалити рядок з невизначеними значеннями?

По аналогічних причинах не можна ввести і новий продукт (наприклад, Баклажани), який пропонує існуючий постачальник (наприклад, "Полісся"). А як ввести нове блюдо, якщо в нім використовується новий продукт (Краби)?

4. **Аномалії видалення.** Зворотна проблема виникає при необхідності видалення всіх продуктів, що поставляються даним постачальником або всіх блюд, що використовують ці продукти. При таких видаленнях будуть втрачені відомості про такого постачальника.

Багато проблем цього прикладу зникнуть, якщо виділити в окремі таблиці відомості про блюда, рецепти, витрату блюд, продукти і їх постачальників, а також створити таблиці, що пов'язують, "Склад" і "Постачання".

Блюдо	Вид	Рецепт	Порций	Дата Р	Продукт	Калор.	Вес (г)	Поставщик	Город	Страна	Вес (кг)	Цена (\$)	Дата П
Лобио	Закуска	Лом.	158	1/9/94	Фасоль	3070	200	"Хуанхэ"	Пекин	Китай	250	0.37	24/8/94
					Лук	450	40	"Наталка"	Киев	Украина	100	0.52	27/8/94
					Масло	7420	30	"Лайма"	Рига	Латвия	70	1.55	30/8/94
					Зелень	180	10	"Даугава"	Рига	Латвия	15	0.99	30/8/94
Харчо	Суп	...	144	1/9/94	Мясо	1660	80	"Наталка"	Киев	Украина	100	2.18	27/8/94
					Лук	450	30	"Наталка"	Киев	Украина	100	0.52	27/8/94
					Томаты	240	40	"Полесье"	Киев	Украина	120	0.45	27/8/94
					Рис	3340	50	"Хуанхэ"	Пекин	Китай	75	0.44	24/8/94
					Масло	7420	15	"Полесье"	Киев	Украина	50	1.62	27/8/94
					Зелень	180	15	"Наталка"	Киев	Украина	10	0.88	27/8/94
Шашлык	Горячее	...	207	1/9/94	Мясо	1660	180	"Юрмала"	Рига	Латвия	200	2.05	30/8/94
					Лук	450	40	"Полесье"	Киев	Украина	50	0.61	27/8/94
					Томаты	240	100	"Полесье"	Киев	Украина	120	0.45	27/8/94
					Зелень	180	20	"Даугава"	Рига	Латвия	15	0.99	30/8/94
Кофе	Десерт	...	235	1/9/94	Кофе	2750	8	"Хуанхэ"	Пекин	Китай	40	2.87	24/8/94

Рис. 3.2 Дані, необхідні для створення бази даних "Живлення"

Блюдо	Вид	Рецепт	Порций	Дата Р	Продукт	Калор.	Вес (г)	Поставщик	Город	Страна	Вес (кг)	Цена (\$)	Дата П
Лобио	Закуска	Лом.	158	1/9/94	Фасоль	3070	200	"Хуанхэ"	Пекин	Китай	250	0.37	24/8/94
Лобио	Закуска	Лом	158	1/9/94	Лук	450	40	"Наталка"	Киев	Украина	100	0.52	27/8/94
Лобио	Закуска	Лом	158	1/9/94	Масло	7420	30	"Лайма"	Рига	Латвия	70	1.55	30/8/94
Лобио	Закуска	Лом	158	1/9/94	Зелень	180	10	"Даугава"	Рига	Латвия	15	0.99	30/8/94
Харчо	Суп	Лом	144	1/9/94	Мясо	1660	80	"Наталка"	Киев	Украина	100	2.18	27/8/94
Харчо	Суп	Лом	144	1/9/94	Лук	450	30	"Наталка"	Киев	Украина	100	0.52	27/8/94
Харчо	Суп	Лом	144	1/9/94	Томаты	240	40	"Полесье"	Киев	Украина	120	0.45	27/8/94
Харчо	Суп	Лом	144	1/9/94	Рис	3340	50	"Хуанхэ"	Пекин	Китай	75	0.44	24/8/94
Харчо	Суп	Лом	144	1/9/94	Масло	7420	15	"Полесье"	Киев	Украина	50	1.62	27/8/94
Харчо	Суп	Лом	144	1/9/94	Зелень	180	15	"Наталка"	Киев	Украина	10	0.88	27/8/94
Шашлык	Горячее	Лом	207	1/9/94	Мясо	1660	180	"Юрмала"	Рига	Латвия	200	2.05	30/8/94
Шашлык	Горячее	Лом	207	1/9/94	Лук	450	40	"Полесье"	Киев	Украина	50	0.61	27/8/94
Шашлык	Горячее	Лом	207	1/9/94	Томаты	240	100	"Полесье"	Киев	Украина	120	0.45	27/8/94
Шашлык	Горячее	Лом	207	1/9/94	Зелень	180	20	"Даугава"	Рига	Латвия	15	0.99	30/8/94
Кофе	Десерт	Лом	235	1/9/94	Кофе	2750	8	"Хуанхэ"	Пекин	Китай	40	2.87	24/8/94

Рис.5.3.3 Універсальне відношення "Живлення"

5.3.4 Нормалізація

Нормалізація - це розбиття таблиці на дві або більш, що володіють кращими властивостями при включенні, зміні і видаленні даних. Остаточна мета нормалізації зводиться до отримання такого проекту бази даних, в якому кожен факт з'являється лише в одному місці, тобто виключена надмірність інформації. Це робиться не стільки з метою економії пам'яті, скільки для виключення можливої суперечності даних, що зберігаються.

Існують наступні нормальні форми:

Таблиця знаходиться в **першій нормальній формі (1нф)** тоді і тільки тоді, коли жодна з її рядків не містить в будь-якому своєму полі більш за одне значення і жодне з її ключових полів не порожньо.

Таблиця знаходиться в другій **нормальній формі (2нф)**, якщо вона задовольняє визначенню 1нф і всі її поля, що не входять в первинний ключ, зв'язані повною функціональною залежністю з первинним ключем.

Таблиця знаходиться в **третьій нормальній формі (3нф)**, якщо вона задовольняє визначенню 2нф і не одне з її не ключових полів не залежить функціонально від будь-якого іншого не ключового поля.

5.3.5 Процедура проектування

Процес проектування інформаційних систем є достатньо складним завданням. Він починається з побудови інфологічеської моделі даних (п. 2), тобто ідентифікації суті. Потім необхідно виконати наступні кроки процедури проектування даталогічеської моделі.

1. Представити кожен стрижень (незалежну суть) таблицею бази даних (базовою таблицею) і специфікувати первинний ключ цієї базової таблиці.
2. Представити кожен асоціацію (зв'язок виду "многие-ко-многим" або "многие-ко-многим-ко-многим" і так далі між суттю) як базову таблицю. Використовувати в цій таблиці зовнішні ключі для ідентифікації учасників асоціації і специфікувати обмеження, пов'язані з кожним з цих зовнішніх ключів.
3. Представити кожен характеристику як базову таблицю із зовнішнім ключем, що ідентифікує суть, що описується цією характеристикою. Специфікувати обмеження на зовнішній ключ цієї таблиці і її первинний ключ - ймовірно, комбінації цього зовнішнього ключа і властивості, яка гарантує "унікальність в рамках описуваної суті".
4. Представити кожне позначення, яке не розглядалося в попередньому пункті, як базову таблицю із зовнішнім ключем, що ідентифікує суть, що позначається. Специфікувати пов'язані з кожним таким зовнішнім ключем обмеження.
5. Представити кожен властивість як поле в базовій таблиці, що представляє суть, яка безпосередньо описується цією властивістю.
6. Для того, щоб виключити в проекті ненавмисні порушення яких-небудь принципів нормалізації, виконати процедуру нормалізації.
7. Якщо в процесі нормалізації було проведено розділення яких-небудь таблиць, то слід модифікувати інфологічеську модель бази даних і повторити перераховані кроки.
8. Вказати обмеження цілісності проектованої бази даних і дати (якщо це необхідно) короткий опис отриманих таблиць і їх полів.

Контрольні питання:

1. Які основні поняття реляційних баз даних ви знаєте?
2. Які основні правила створення реляційної бази даних?
3. Що таке універсальне відношення?
4. Для чого використовується універсальне відношення?
5. Які нормальні форми ви знаєте?
6. Яка процедура проектування реляційної бази даних?

5.4. Microsoft Access. Создание таблиц

Запуск програми Microsoft Access. Об'єкти бази даних Access. Створення таблиць в режимі таблиць. Створення таблиць в режимі конструктора. Створення таблиці за допомогою майстра.

5.4.1 Запуск программы Microsoft Access

Розглядатимемо програму Microsoft Access версії 2003. Запускаючий файл зазвичай знаходиться по наступному шляху:

C:\Program Files\Microsoft Office\Office\MSAccess.exe

Всі можливості Microsoft Access зведені в два меню і одну інструментальну панель. Першим меню є системне, розташоване у верхній частині робочого вікна Microsoft Access. Це Файл, Правка, Вигляд, Вставка, Сервіс, Вікно і Допомога. Другим призначенням для користувача меню є контекстно-залежне, таке, що викликається натисненням правою клавшею миші. Паралельно з меню існує панель інструментів. В результаті роботи програми Microsoft Access створюється файл з розширенням *.mdb.

Вся початкова інформація зберігається в чітко певних таблицях. Під чітким визначенням мається на увазі така структура таблиці, в якій кожен рядок має унікальний ідентифікатор (наприклад, номер рядка), а дані представлені стовпцями. Таким чином, будь-яка таблиця є одновимірним набором записів. Неодмінним правилом створення таблиці в СУБД є строге визначення вмісту самої таблиці. У її осередках може зберігатися тільки фактична, і лише незмінна інформація.

Після запуску програми з'являється вікно наступного вигляду

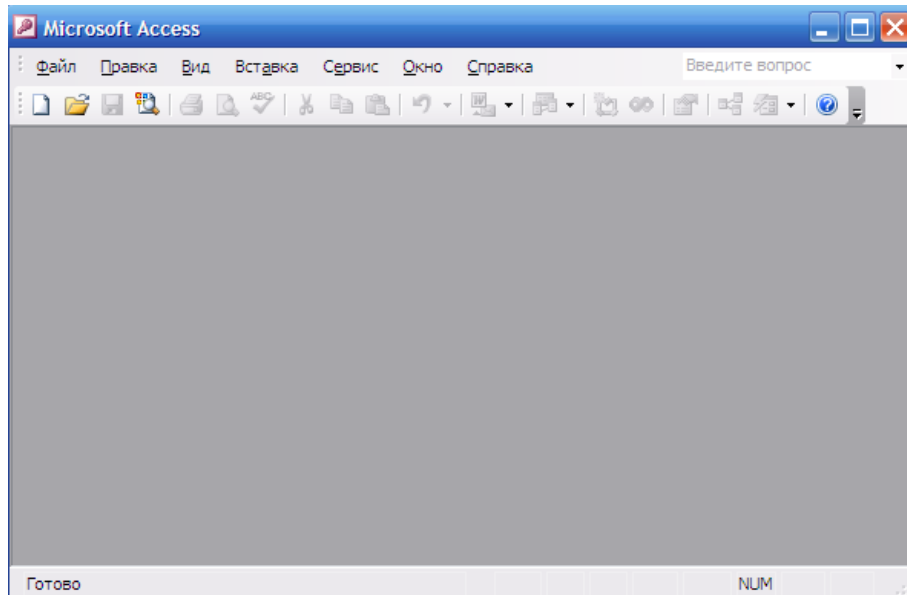




Рис.5.4.1 Вікно запуску Access

Для початку роботи необхідно відкрити існуючу базу даних або створити нову. Для цього необхідно вибрати в меню Файл → Створити () або Файл → Відкрити ().

Після вибору одного або іншого способу роботи з базою з'явиться вікно бази даних, що складається з набору панелей з вкладками, кожна з яких відповідає одному з 6 типів об'єктів бази даних Access: **таблиці, запити, форми, звіти, макроси і модулі і сторінки.**

Схема взаємодії об'єктів бази даних зображена на рис. 5.4.2. Основною одиницею зберігання даних тут є таблиця.

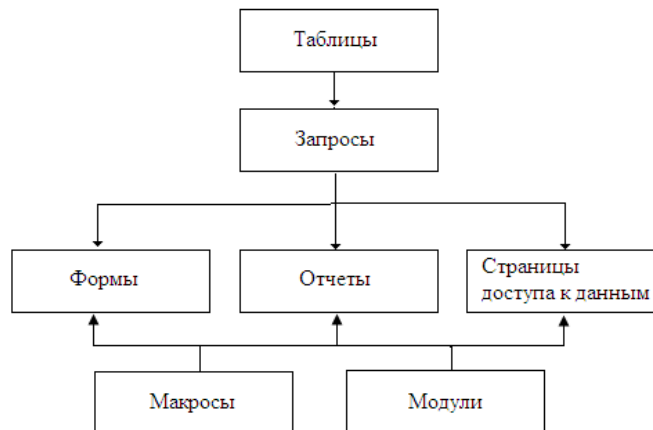


Рис.5.4.2 Схема взаємодії об'єктів бази даних

Розглянемо основне призначення об'єктів бази даних Access.

Таблиця - це об'єкт, визначуваний для зберігання даних. Кожна таблиця включає інформацію про об'єкт реального миру, наприклад про клієнтів фірми. Таблиця складається із заголовка і тіла. Заголовок включає імена атрибутів об'єкту (стовпців) і їх властивості, наприклад прізвище, телефон і адресу клієнта. Тіло містить кортежі (рядки), кожен рядок представляє безліч значень стовпців, в яких зберігаються дані про конкретний екземпляр об'єкту.

Запит - це об'єкт, який дозволяє користувачеві отримати потрібні дані з однієї або декількох базових таблиць, і інших запитів. У запиті можна вказати умови, яким повинні задовольняти дані. Завдяки цьому запит дозволяє з великого масиву інформації, що зберігається в БД, витягувати тільки потрібні дані. Для створення запиту використовують запит за зразком (*QBE*) або інструкції *SQL*.




Форма - це об'єкт, призначений в основному для введення даних, відображення їх на екрані або управління роботою додатку. Форми використовуються для того, щоб реалізувати вимоги замовника до представлення даних з таблиць і запитів. Форми можна роздрукувати. За допомогою форми можна у відповідь на деяку подію запустити *макрос* або *процедуру*, що виконують певну обробку даних.

Звіт - це об'єкт, призначений для створення документа, який згодом може бути роздрукований або включений в документ іншого застосування. Перш ніж виводити звіт на принтер, його можна проглянути на екрані.

Макрос - це об'єкт, що є структурованим описом одного або декількох дій, які виконуватимуться у відповідь на певну подію. Наприклад, можна визначити макрос, який у відповідь на вибір деякого елементу в основній формі відкриває іншу форму.

Модуль - це об'єкт, що містить програми на Microsoft Access Visual Basic, які можуть розроблятися користувачем для реалізації нестандартних процедур при створенні додатку.

Розглянемо зовнішній вигляд вікна програми (див. рис. 5.4.3).

Кожна область робочого екрану містить вгорі три кнопки: Відкрити ( Открыть), Створити ( Создать) и Конструктор ( Конструктор).

Кнопка **Відкрити** призначена для активізації виділеного елементу з наявних в даному проекті. Відкриваючись, ці елементи предстають в тому вигляді, який використовується для його перегляду. Таблиці і запити відкриваються у вигляді таблиці. Форми і звіти - в тому

вигляді, в якому вони повинні представати перед користувачем. Лише макроси і модулі відкриваються у вигляді для редагування.

Всі об'єкти в Access можуть бути створені користувачем за допомогою **конструктора** або за допомогою різних **майстрів**. *Майстри* допомагають користувачеві в режимі діалогу створювати об'єкти, дають підказки, пропонують свої рішення, що полегшує роботу початкуючим і непрофесійним користувачам.

Також для створення нових елементів того або іншого вигляду можна використовувати кнопку **Створити**.

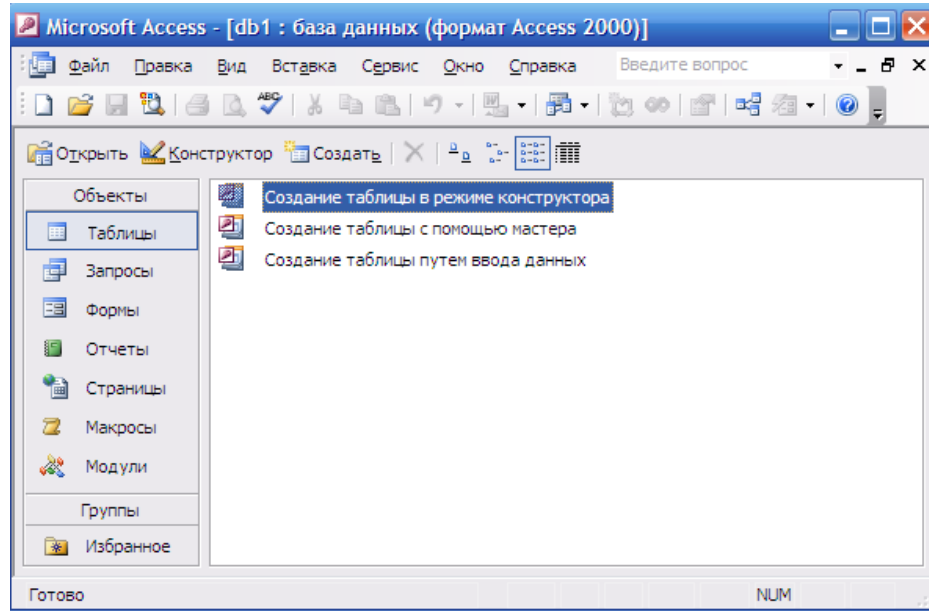


Рис. 5.4.3 Вікно для роботи з базою даних

5.4.2 Створення таблиць

При роботі з Access таблиці є одним з основних об'єктів, на їх базі здійснюється побудова всіх інших елементів, таких, як форми, запити і звіти. У таблиці збираються дані по конкретній темі, наприклад вся інформація про клієнтуру фірми.

БД Access може складатися з декількох таблиць, в кожній з яких зберігається інформація на одну тему. У одній таблиці можуть зберігатися відомості про клієнтів, в іншій - про всі торгові угоди, які поміщені з тими або іншими клієнтами, в третій - інформація про витрати, податки і витрати на розвиток фірми, в четвертій - інвентаризаційний список складського фонду, в п'ятій - терміни проведення виставок і презентацій і так далі

У вікні БД клацніть на кнопці Таблиці і натисніть кнопку Створити, щоб приступити до проектування нової таблиці. Access відкриє діалогове вікно (див. рис. 5.4.4), в якому буде запропоновано скористатися одним з наступних способів створення таблиці.

Режим таблиці - створення таблиці в табличному уявленні (проектування таблиці відбувається в ході її заповнення по аналізу даних, що вводяться).

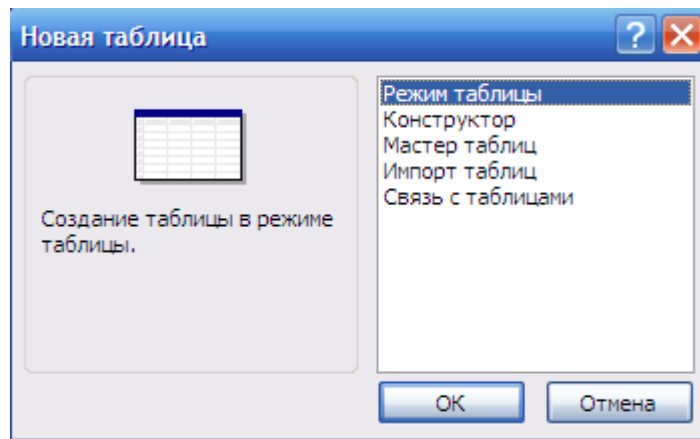


Рис. 5.4.4 Засоби для створення таблиць

Конструктор - створення таблиці за допомогою конструктора таблиць.

Майстер таблиць - створення таблиці за допомогою майстра таблиць на основі колекції таблиць і полів.

Імпорт таблиць - створення таблиці шляхом імпорту даних із зовнішнього файлу або з іншої БД.

Зв'язок з таблицями - приєднання зовнішнього файлу або таблиці іншої БД.

Розглянемо докладніше кожен засіб створення таблиць.

5.4.2.1 Режим таблиць

У цьому режимі дані представлені у вигляді таблиці. У цьому режимі можна вводити або редагувати дані, а також перевизначати саму таблицю. Кожен рядок відповідає запису, а кожен стовпець - полю. Останній рядок таблиці відмічений * позначає місце для нового запису.

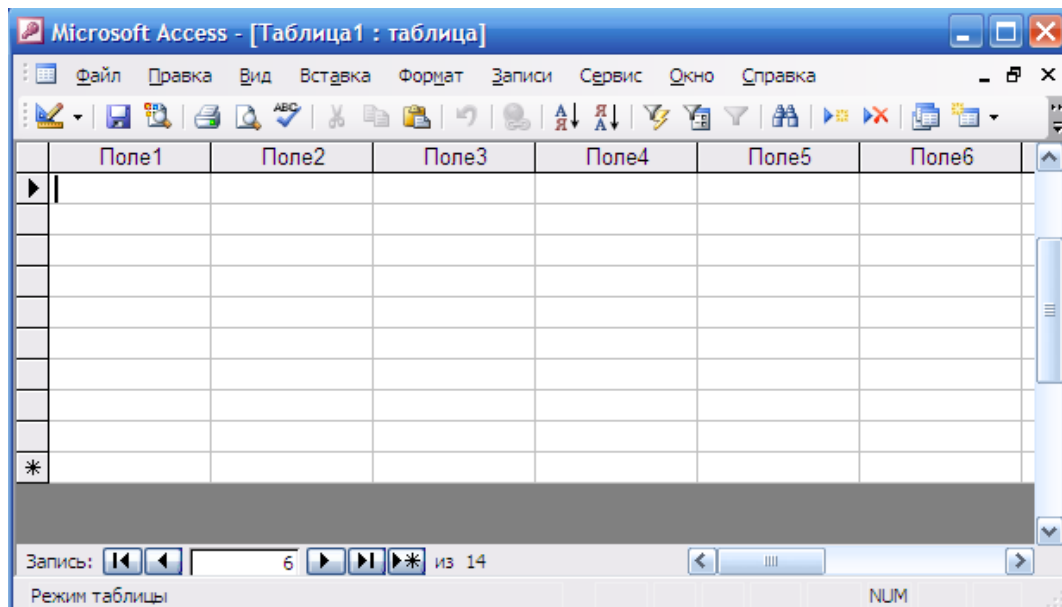






Рис. 5.4.5 Нова таблиця в режимі створення таблиць

Нижній рядок вікна режиму таблиць призначений для швидкого переміщення по записах. Елемент  відображає номер поточного запису. Щоб перейти до заданого запису, необхідно ввести потрібний номер запису в це поле. Елементи  та  дозволяють пересуватися по записах, крайні - пересування в самий початок і в самий кінець соотв.,

внутрішні - на один запис назад або вперед. Елемент  відповідає за переміщення в полі введення нового запису.

Управління структурою і видом таблиці:

— Щоб вставити новий запис в середині таблиці, необхідно клацнути на заголовок того рядка, в який ви хочете її ввести і вибрати **Новий запис**.

— Для переміщення стовпця мишею, необхідно клацнути на його назві і не відпускаючи кнопку миші і перетягнути стовпець на нове місце.

— Щоб змінити ім'я стовпця, для цього необхідно двічі натиснути на заголовок стовпця і ввести нове ім'я.

— Щоб змінити розмір стовпця, необхідно підвести покажчик миші до правої межі його заголовка і перетягнути межу до потрібної ширини. Розмір рядка змінюється для всіх рядків одночасно.

— Щоб деякі стовпці завжди залишалися видимими при горизонтальній прокрутці вікна, необхідно встановити покажчик на заголовок стовпців, виділити їх, клацнути правою кнопкою миші і вибрати **Закріпити стовпці**. Закріплені стовпці переміщуються до лівого краю вікна.

— Можна сховати/отобразити стовпці для цього натиснути праву клавішу миші і вибрати **Приховати стовпці**. Для відображення необхідно встановити покажчик миші на вільну від стовпців і рядків частину вікна, натиснути праву клавішу миші і вибрати **Відобразити стовпці** і у вікні, що з'явилося, галочками позначити які стовпці ви хочете відобразити.

5.4.2.2 Режим конструктора

У режимі Конструктора ми можемо створити нову таблицю або відредагувати що існує (рис. 5.4.6).

Щоб створити нове поле, необхідно провести наступні дії:

1. Ввести його назву в ім'я **поля**.
2. У стовпці **Тип даних** вибрати відповідний тип із списку, що розкривається.
3. При необхідності додати опис поля в колонці **Опис**.

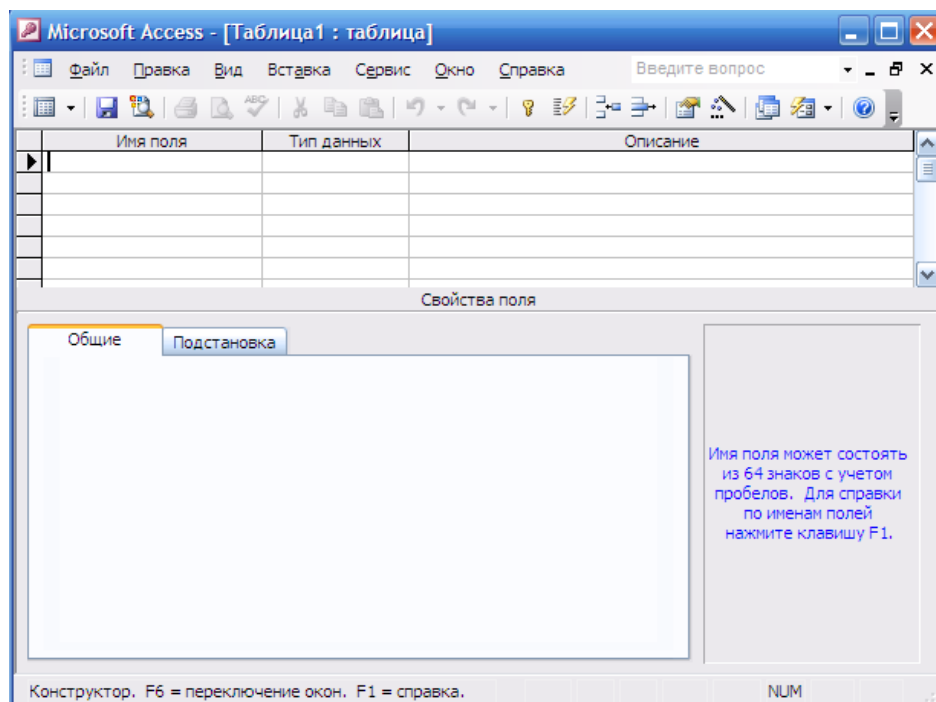


Рис. 5.4.6 Створення таблиць в режимі конструктора

Властивість **Ім'я поля (Fieldname)** визначає ім'я поля в таблиці. Введіть ім'я, що задовольняє угодам про імена об'єктів Microsoft Access. Це ім'я не повинне збігатися з ім'ям іншого поля в цій таблиці.

Угоди про імена - це набір правил, що обмежують допустимі імена об'єктів Microsoft Access. Імена об'єктів повинні містити не більше 64 символів і можуть включати будь-які комбінації букв, цифр пропусків і спеціальних символів за винятком крапки (.), знаку (!) оклику, надрядкового символу (') і прямих дужок ([]). Відзначимо, що ім'я не повинне починатися з пропуску і містити символи, що управляють (з кодами ASCII 00 - 31).

Прагніть не включати в імена об'єктів пропуски, особливо, якщо передбачається часто використовувати посилання на ці імена у виразах. Уникайте дуже довгих імен: такі імена важко запам'ятовувати і на них незручно посилатися.

Властивість **Тип даних (Datatype)** визначає тип даних, що зберігаються в полі таблиці. У кожне поле допускається введення даних тільки одного типу.

Існують наступні типи даних:

Тип	Вміст поля	Розмір
Текстовий	(Значення за умовчанням). Текст або числа, що не вимагають проведення розрахунків, наприклад, номери телефонів	Число символів, мінімальне, що не перевищує, з двох значень: 255 або значення властивості Розмір поля
Поле МЕМО	Довгий текст або комбінація тексту і чисел	До 65535 символів
Числовий	Числові дані, використовувані для проведення розрахунків	1, 2, 4 або 8 байт
Дата/час	Дати і час, що відносяться до років з 100 по 9999, включно	8 байт
Грошовий	Грошові значення і числові дані, використовувані в математичних розрахунках, що проводяться з точністю до 15 знаків в цілій і до 4 знаків в дробовій частині	8 байт
Счетчик	Унікальні послідовно зростаючі (на 1) або випадкові числа, що автоматично вводяться при додаванні кожному новому запису в таблицю. Значення полів типу счетчик оновлювати не можна	4 байт
Логічний	Логічні значення, а також поля, які можуть містити одне з двох можливих значень (True/False, Да/Ні)	1 бит
Поле об'єкту OLE	Об'єкт (наприклад, електронна таблиця Microsoft Excel, документ Microsoft Word, малюнок, звукозапис або інші дані в двійковому форматі), зв'язаний або упроваджений в таблицю Microsoft Access	До 1 Гбайт (обмежується об'ємом диска)
Гіперпосилання	Рядок, що складається з букв і цифр, і що представляє адресу гіперпосилання	до 2048 символів
Майстер підстановок	Створює поле, в якому пропонується вибір значень із списку, або з поля із списком, що містить набір постійних значень або значень з іншої таблиці. Вибір цього параметра в списку в осередку запускає майстра підстановок, який визначає тип поля	4 байт


Зміна типу поля після введення даних в таблицю викличе те, що займає достатньо довгий час перетворення даних при збереженні таблиці. Несумісність існуючих даних з новим значенням властивості Тип даних (Datatype) може привести до втрати даних або видачі помилки.


Властивість **Опис (Description)** визначає текст, що містить опис окремих полів таблиці. Максимальна довжина опису складає 255 символів.

Область **Властивість поля** (див. рис. 5.4.6) містить список властивостей, доступних для типу даних вибраного в даний момент поля. Закладка Загальні цієї області містить наступні властивості:

- **Розмір поля** - визначає максимальний розмір даних, які можуть зберігатися в полях з типом даних Текстовий, Числовий або Счетчик.
- **Формат поля** - дозволяє вказати формати виведення тексту, чисел, дат і значень часу на екран і на друк.
- **Маска введення** - задає маску введення, що полегшує введення даних в полі. Наприклад, зручно створити наступну маску введення для поля «Телефон», що дозволяє вводити тільки цифри і що автоматично додає проміжні символи: (____) ____-____. Значення даної властивості визначається автоматично при використанні майстра по створенню масок введення.
- **Підпис** - визначає текст, який виводиться в підписах об'єктів в різних режимах. Якщо користувач не вказує текст підпису поля таблиці, то як текст підпису елементу управління і заголовка стовпця в режимі таблиці використовується значення властивості Ім'я поля.
- **Значення за умовчанням** - дозволяє вказати значення, що автоматично вводиться в поле при створенні нового запису. Наприклад, в таблиці «Адреси» може виявитися зручним вказати автоматичне введення значення «Москва» в полі «Місто». При заповненні таблиці користувачі зможуть залишити в цьому полі стандартне значення або, при необхідності, вказати інше місто.
- **Умова на значення** - визначає вимоги до даних, що вводяться в запис, в полі або в елемент управління. Максимальна довжина значення складає 2048 символів. Для елементів управління може бути вказане будь-який правильний вираз. Вираз, вказаний як умова для поля, не повинен містити посилання на інші поля.
- **Повідомлення про помилку** - дозволяє вказати текст повідомлення, що виводиться на екран, якщо введені дані порушують умову, визначену у властивості Умова на значення. Максимальна довжина значення властивості Повідомлення про помилку складає 255 символів.
- **Обов'язкове поле** - вказує, чи вимагає поле обов'язкового введення значення. Якщо ця властивість має значення «Так», то при введенні нового запису необхідно ввести значення в це поле.





У режимі Конструктора можна виконувати наступні дії:

- Щоб перемістити поле в списку, клацніть на заголовку цього поля, відпустите кнопку миші і перетягнете поле в нове місце.
- Щоб вставити або видалити поля (рядки), встановіть покажчик на необхідному рядку, клацніть правою кнопкою миші, і виберіть відповідну команду контекстного меню.
- Щоб додати поле з наданих таблиць Access, встановіть покажчик в тому місці списку, де ви хочете вставити нове поле, клацніть правій миші і виберіть команду контекстного меню **Побудувати**. У вікні, що з'явилось, вибрати зразок таблиці, а потім зразок поля.
- Для позначення поля як ключового, необхідно виділити потрібний рядок або групу рядків і вибрати піктограму  або Правка ☐ Ключове поле.

— Щоб ввести конкретні значення в таблицю, що вийшла, необхідно зберегти таблицю, а потім вибрати **Вигляд** □ **Режим таблиці** або натиснути піктограму .

5.4.2.3 Майстер таблиць

Дозволяє створити таблицю, використовуючи зразки таблиць, що поставляються разом з Access. Створення таблиці поділене на кроки. На кожному кроці майстра проводяться різні налаштування для майбутньої таблиці. Перехід до наступного кроку здійснюється натисненням кнопки **Далі**.

На першому кроці необхідно вибрати зразок таблиці і полів. Зразки таблиць діляться на ділове застосування і особисте застосування. Вікно для вибору потрібного зразка і полів з цього зразка виглядає, як показано на рисунку 5.4.7. Для вибору потрібного поля використовуються кнопки:  - для вибору одного поля,  - для вибору всіх полів зразка таблиці,  - для видалення одного вибраного поля,  - для видалення всіх вибраних полів. На цьому ж кроці вибраним полям можна привласнити інші імена. Для цього необхідно вибрати поле і натиснути кнопку **Перейменувати поле** і ввести нове ім'я.

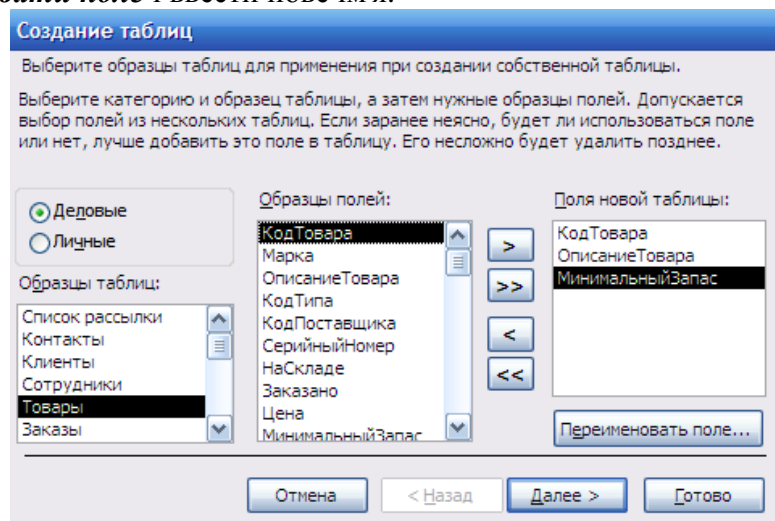


Рис. 5.4.7 Майстер таблиць. Вікно для вибору полів.

Після вибору полів для своєї таблиці, переходиться до наступного кроку, в якому можна ввести ім'я для нової таблиці і вибрати чи будете ви самостійно вибирати ключове поле або надасте цю операцію Access (див. рис. 5.4.8).

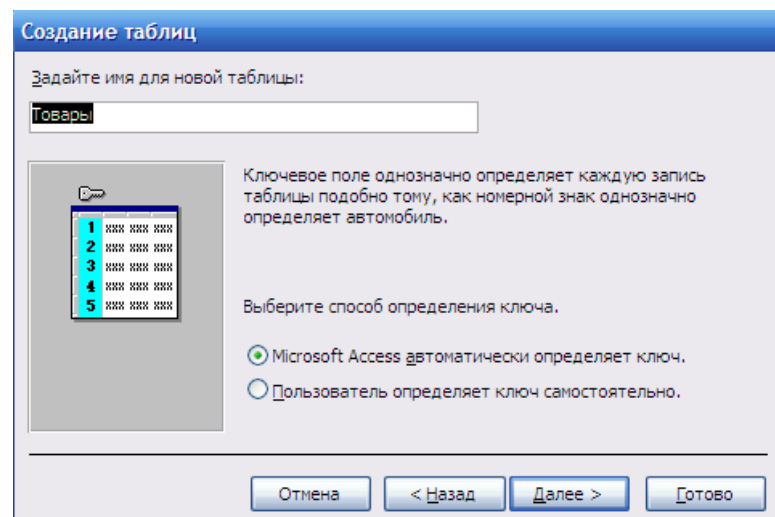


Рис. 5.4.8 Майстер таблиць. Вікно для вибору способу визначення ключа.

Якщо ви вибрали самостійне визначення ключового поля, то на наступному кроці, необхідно вибрати поле, яке буде ключовим, і які дані міститимуться в ключовому полі:

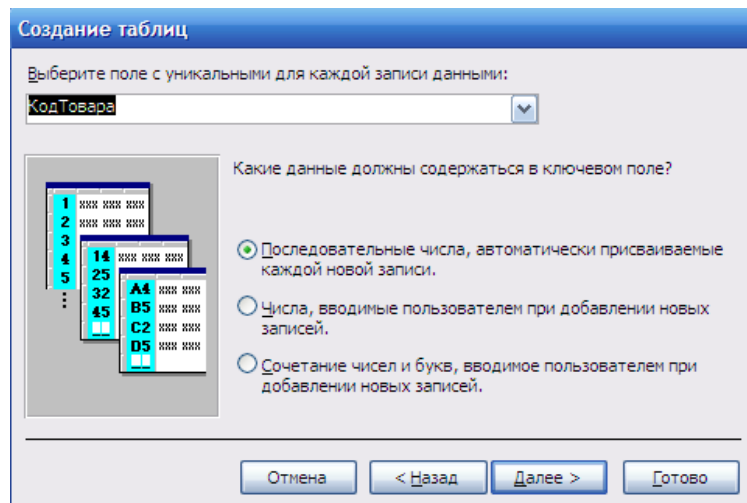


Рис. 5.4.9 Мастер таблиц. Вікно для вибору ключового поля

На завершальному кроці ви вибираєте подальші дії, які ви хочете виконати після створення таблиці (див. рис. 5.4.10). До таких дій відносяться: зміна структури таблиці, тобто відкриття створеної таблиці в режимі конструктора; введення даних, тобто відкриття таблиці в режимі таблиці; введення даних за допомогою форми, тобто створення макету форми для створеної таблиці.

У завершенні роботи майстра необхідно натиснути кнопку **Готово**.

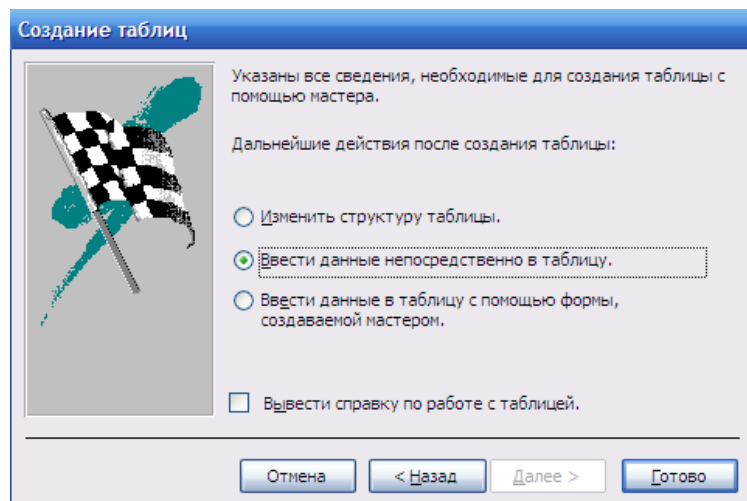


Рис.5.4.10 Мастер таблиц.
Вікно для вибору подальших дій після створення таблиць

5.4.3. Види представлення таблиці

Після створення таблиці ми ще не раз можемо звернутися до таблиці, щоб проглянути, обробити, змінити або проаналізувати дані. Залежно від конкретних цілей ми можемо представляти таблицю в тому або іншому вигляді, найбільш відповідному нам.

Для відкриття створеної таблиці необхідно виконати подвійне клацання на її назві. За умовчанням таблиця відкриється у вигляді **Таблиця**. Всього існує 4 види представлення

таблиць, перемикаючи їх можна або кнопкою панелі інструментів, або за допомогою вибору відповідних пунктів меню Вигляд:

- **Таблиця** - вигляд, найбільш оптимальний для представлення табличних даних, він відкриває природну структуру їх уявлення.
- **Конструктор** - вид представлення таблиці, що відображає вичерпну інформацію про поля таблиці і їх властивості, - так звані *Метадани таблиці*.
- **Звідна таблиця** - вид представлення великих об'ємів даних, що робить їх зручними для аналізу і обробки.
- **Звідна діаграма** - ще зручніший вид представлення великих об'ємів даних, представлення даних в цьому вигляді наочніше, але менш інформативно, але в той же час з використанням звідних діаграм аналізувати дані значно простіше.

5.4.4. Редагування проекту таблиці

Скористаємося можливістю внесення змін в структуру таблиці. Зміни вноситимемо в режимі конструктора. Хай в існуючу таблицю, наприклад Товар, необхідно додати нове поле Вид товару, яке матиме тільки два значення: штучний або ваговий. Для додавання такого поля необхідно вказати позицію для його вставки, зайти в меню Вставка] Поле підстановок. Після вибору **Поле підстановок** активізується **майстер підстановок**, який візьме управління на себе.

На першому кроці майстра підстановок вимагає визначити вид джерела даних для підстановки. Наприклад, для нашого завдання, виберемо Фіксований набір значень (див. рис. 5.4.11)

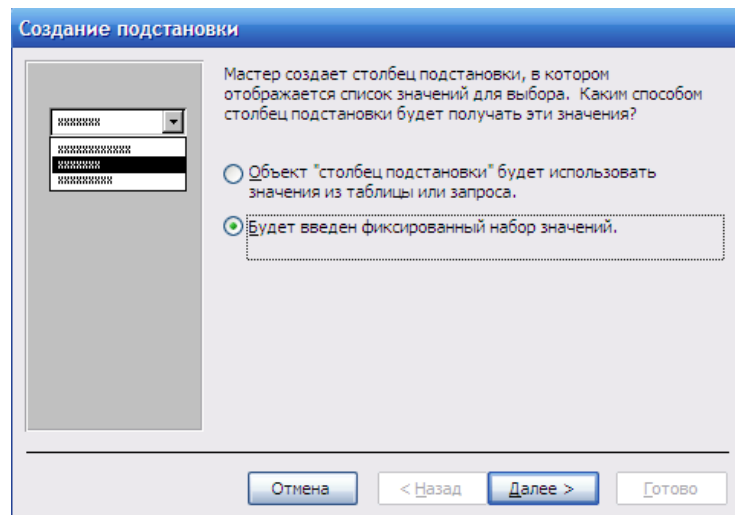


Рис.5.4.11 Перший крок майстра підстановок

На другому кроці вимагає визначити список значень для підстановки. Список значень вводиться користувачем в стовпець з ім'ям Столбец1 (див. рис. 5.4.12).

Создание подстановки

Выберите значения, которые будут содержать столбец подстановки. Введите число столбцов списка и значения для каждой ячейки.

Перетащите правую границу заголовка столбца на нужную ширину или дважды щелкните ее для автоматического подбора ширины.

Число столбцов:

Столбец1
Штучный
Весовой
*

Отмена < Назад Далее > Готово

Рис. 5.4.12 Другой шаг мастера підстановок

На третьому кроці необхідно ввести підпис для створюваного поля (див. рис. 5.4.13).

Создание подстановки

Задайте подпись, которую содержит столбец подстановки.

Вид товара

Указаны все сведения, необходимые мастеру, чтобы создать столбец подстановки.

☐ Вывести справку, после того как мастер создаст столбец подстановки.

Отмена < Назад Далее > Готово

Рис. 5.4.13 Завершальный шаг мастера підстановок

Налаштування підстановки здійснюється в розділі Властивості поля, закладка Підстановка.

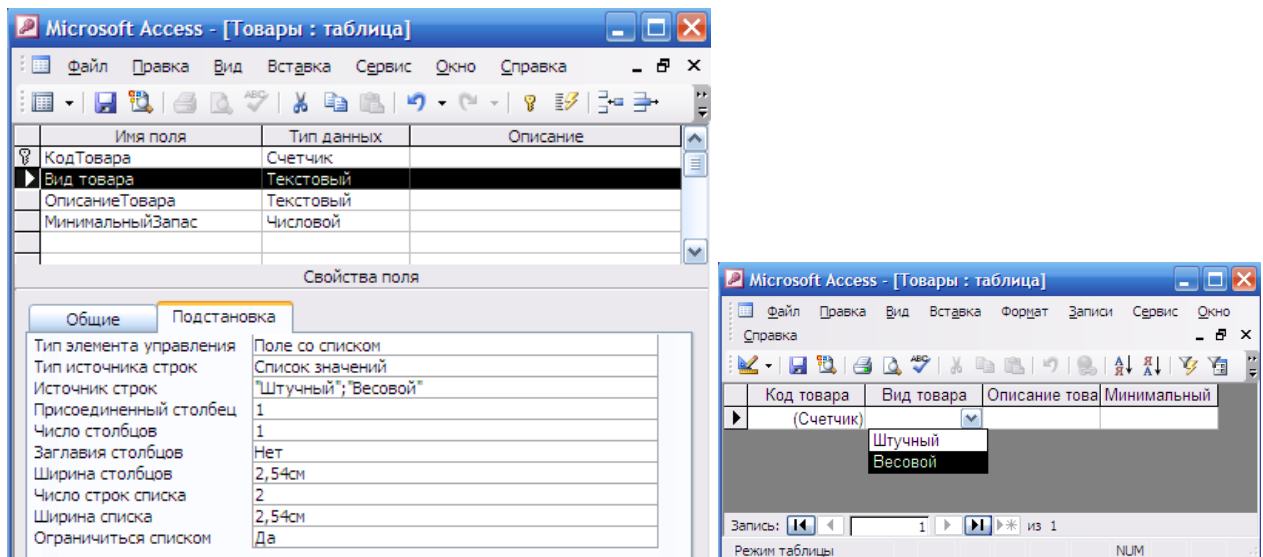


Рис. 5.4.14 Результат работы мастера подстановок

Розглянемо налаштування підстановки (див. рис. 5.4.14):

- Тип елемента управління - Визначає вид управління, що виводиться для поля елемента (в даному випадку це *Поле із списком*).
- Тип джерела рядків - Визначає тип джерела даних. Оскільки дані вводилися безпосередньо користувачем, то ця властивість має значення *Список значень*
- Джерело рядків - Визначає джерело даних для поля.
- Приєднаний стовпець - Номер стовпця, значення якого виводитиметься в полі.
- Число стовпців - Визначає кількість стовпців для показу.
- Заголовки стовпців - Визначає, чи слід виводити для поля із списком заголовки.
- Ширіна стовпців - Визначає ширину стовпців в списку.
- Число рядків списку - Визначає кількість елементів в списку.
- Ширіна списку - Визначає ширину списку, що виводиться.
- Обмежитися списком - Визначає вибір значень для даного поля: *Так* - значення можуть належати списку, що тільки виводиться

Контрольні питання:



1. Які 6 типів об'єктів бази даних Access ви знаєте?
2. Скільки способів ви знаєте для створення таблиць?
3. Чим відрізняються ці способи створення таблиць?
4. Які типи даних використовуються в Access?
5. Що таке поле підстановок? Для чого використовується? Як його зробити?

5.5. Визначення зв'язків між таблицями

Встановлення зв'язків між таблицями. Зміна зв'язків. Забезпечення цілісності даних.

Після створення різних таблиць, що містять дані, бази даних, що відносяться до різних аспектів, розробник повинен продумати, яким чином Microsoft Access об'єднуватиме ці дані при їх витяганні з бази даних. Першим кроком при цьому є визначення зв'язків між таблицями.

Зв'язок між таблицями встановлює стосунки між співпадаючими значеннями в ключових полях, зазвичай між полями різних таблиць, що мають однакові імена. В більшості випадків з ключовим полем однієї таблиці, унікальним ідентифікатором кожного запису, що є, зв'язується зовнішній ключ іншої таблиці.

Для того, щоб визначити зв'язок між таблицями, слід вибрати Сервіс  Схема даних або клацнути на панелі інструментів на піктограму . Якщо зв'язки ще не були встановлені, то з'явиться вікно в якому необхідно вибрати таблиці і/або запити між якими буде встановлений зв'язок (див. рис. 5.5.1).

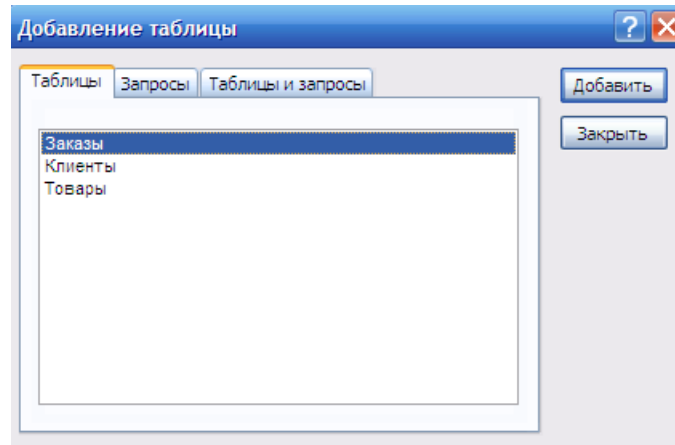



Рис. 5.5.1 Вікно для додавання таблиць для встановлення зв'язків

Також вікно для додавання таблиць можна викликати з панелі інструментів натиснувши на кнопку Відобразити таблицю () або вибравши меню Зв'язки → Додати таблицю.

Після додавання потрібних таблиць, Access автоматично виставить зв'язки, ґрунтуючись на однакових іменах полів (див. рис. 5.5.2).

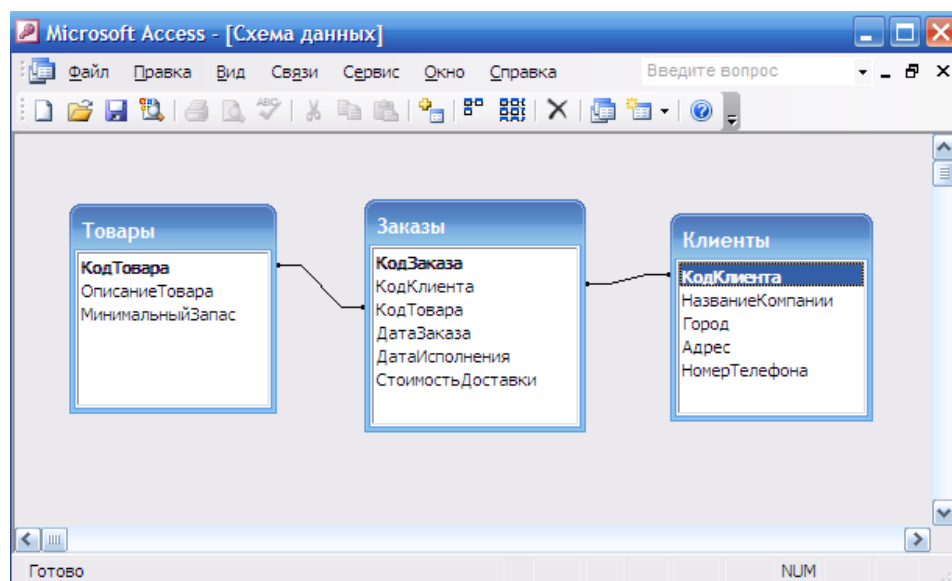


Рис. 5.5.2 Вікно Схеми даних.

Для ручного встановлення зв'язків необхідно за допомогою миші перенесіть поле, яке слід використовувати для установки зв'язку, із списку однієї таблиці до відповідного поля іншої таблиці. На екрані з'явиться діалогове вікно *Зміна зв'язків* (див. рис. 5.3).

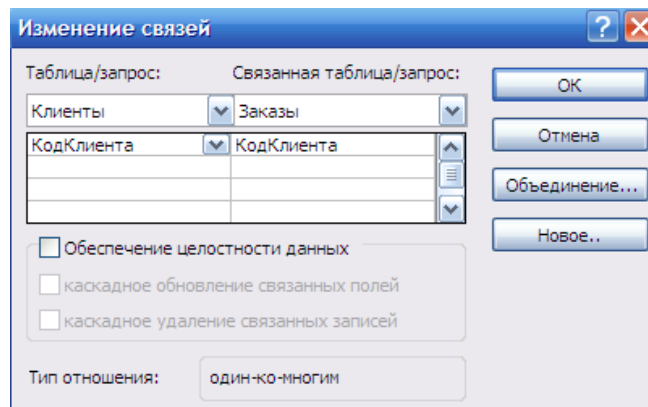


Рис. 5.5.3 Вікно для установки зв'язків між полями двох таблиць

Встановлення опції перевірки посилальної цілісності **Забезпечення цілісності даних** забезпечує перевірку посилальної цілісності зв'язку між обома таблицями. Ця перевірка дозволяє уникнути ряду помилок, що допускаються при видаленні записів з первинної таблиці і введенні інформації в зв'язану таблицю. Завдяки перевірці посилальної цілісності можна уникнути наступних помилкових ситуацій:

- додавання в зв'язану таблицю записів, для яких відсутній відповідний запис в первинній таблиці;
- здійснення змін в головній таблиці, які приведуть до появи «усиротілих» записів в зв'язаній таблиці;
- видалення записів в головній таблиці, на які посилаються записи із зв'язаної таблиці.

Контрольні питання:

1. Для чого встановлюють зв'язки між таблицями?
2. Які кроки необхідно зробити, щоб встановити зв'язок між таблицями?
3. Для чого використовується опція Забезпечення цілісності даних?

5.6. Створення запитів

Поняття запиту. Види запитів. Способи створення запитів. Запити в режимі конструктора. Запити на мові SQL. Приклади запитів.

Запити - засіб відбору і сортування даних - в Access є проміжною ланкою між, таблицями і формами. Запроси- це своєрідні віртуальні таблиці, які, володіючи всіма властивостями таблиць, насправді сховищем даних не є.

Запити призначені для крупних операцій по вибірці інформації, а також для додавання і видалення записів в існуючих таблицях і створення нові.

Існують наступні види запитів:

1. Запит на вибірку даних
2. Параметризований запит
3. Табличний запит
4. Запити-дії

Запит на вибірку даних - основний вид запитів. Він вибирає дані, відповідні вказаному критерію, з однієї або декількох таблиць і поміщає їх в нову таблицю. У запит цього вигляду можна включити різні обчислення і підрахунок підсумкових значень. Значення результуючої таблиці можна редагувати і внесені зміни запам'ятовуються в полях початкових таблиць.

Параметризований запит – це запит на вибірку, критерій якого визначається під час запуску. Параметри критерію вводяться в спеціальному діалоговому вікні, що з'являється при запуску запиту, що параметризується.

Підрахунок середнього, суми або кількості значень, в одному полі, згрупованому по іншому, здійснюється за допомогою **табличного запиту**.

Запити-дії призначені для одночасної зміни декількох записів. Вони використовуються в тих випадках, коли, наприклад, необхідно підняти ціни по всій таблиці або додати префікс до всіх номерів частин продуктів певної лінії.

Запити можна створювати за допомогою двох режимів:

- режим конструктора мова QBE (Query By Example);
- режим SQL.

Таблична мова запитів QBE (скорочення від Query-by-example або Запитів за зразком), разом з мовою SQL, використовується для створення різних запитів до реляційних БД. Мова QBE є наочнішою і простішою для розуміння в порівнянні з SQL, хоча і більш обмеженою в можливостях.

Для створення нового запиту необхідно натиснути кнопку Створити. В результаті з'явиться вікно з інструментами для створення запитів.

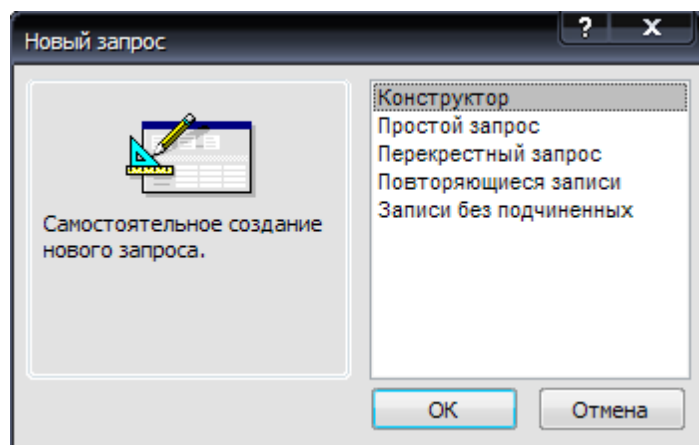


Рис. 5.6.1 Вікно вибору способу створення запиту

Access надає наступні можливості для створення нового запиту:


1. **Конструктор** - створення запиту в режимі конструктора.
2. **Простий запит** - створення запиту на вибірку з певних полів.
3. **Перехресний запит** - створення запиту, що виводить дані в компактному форматі, подібному формату електронної таблиці.
4. **Записи, що повторюються**, - створення запитів на пошук записів, що повторюються, в простій таблиці або запиті.
5. **Записи без підлеглих** - створення запитів на пошук записів, яким не відповідає жоден запис в підлеглий таблиці.

5.6.1 Створення запитів в режимі конструктора

При виборі режиму конструктора або при натисненні кнопки Конструктор з'являється вікно Додавання таблиці. У ній необхідно вибрати таблицю або декілька таблиць, які будуть вам необхідні для побудови нового запиту. Їх додавання відбувається після натиснення кнопки додати. Після додавання потрібних таблиць закрийте це вікно.

Режим конструювання запитів має вид наступного вікна: (див. рис. 5.6.2)

Вікно запиту розбите горизонтально посередині. У верхній половині відображаються списки полів всіх вибраних таблиць, що беруть участь в запиті, а в нижній - власне специфікація запиту. Таблиці додаються в запит за допомогою кнопки панелі інструментів

(Додати таблицю) або за допомогою контекстного меню, викликаного для верхньої половини вікна. Запуск запиту на виконання здійснюється натисненням на кнопку панелі інструментів  (Запуск).

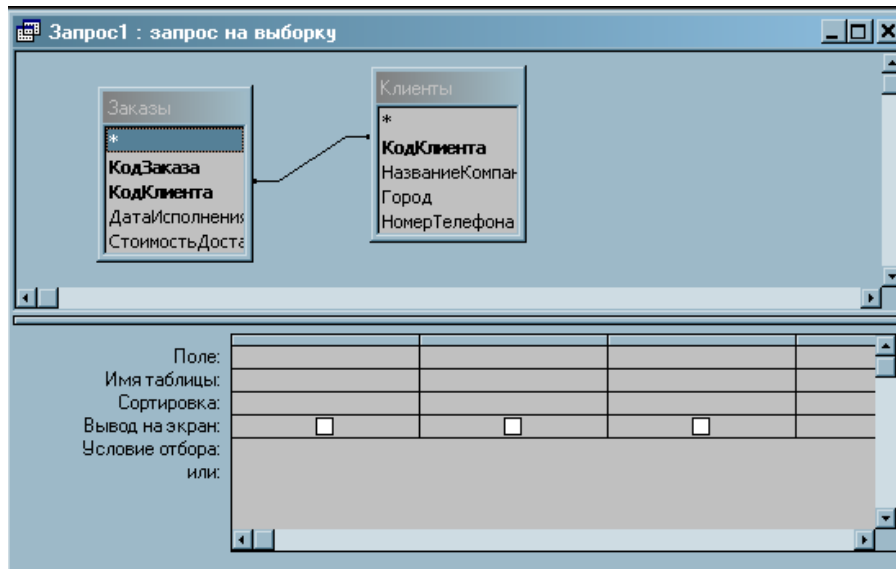



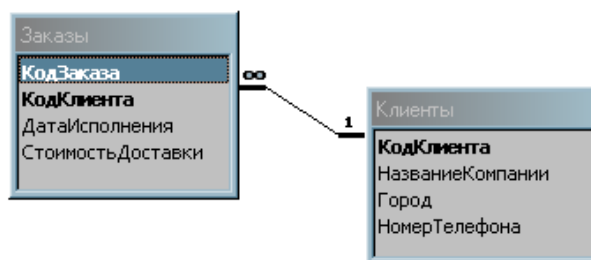
Рис. 5.6.2 Вікно створення запитів в режимі Конструктора

У нижній частині вікна розташована сітка побудови запиту. Кожен стовпець відповідає полю, дані з якого включаються в запит. Сітка складається з наступних рядків:

- **Поле** - щоб вставити поле якоїсь таблиці в запит, його потрібно вибрати із списку полів випадного в цьому рядку. Зірочка * у списку полів позначає включення всіх полів з відповідної таблиці.
- **Ім'я таблиці** - указується ім'я таблиці, поле якого ми вибрали раніше.
- **Групові операції** - використовується для підрахунку ряду обчислень. Якщо такий рядок відсутній в сітці, то вона додається при натисненні кнопки панелі інструментів .
- **Сортування** - в цьому рядку визначається по яких полях будуть відсортовані результати виконання запиту. При сортуванні по декількох полях Access сортує дані в порядку появи полів в сітці побудови запиту зліва направо. Положення стовпця поля можна змінити, виділивши його клацанням миші на заголовку і перетягнувши його на нове місце.
- **Вивід на екран** -установлюється, якщо дані з поля відповідного стовпця мають бути включені в результат запиту.
- **Умова відбору** - в цьому рядку вводиться критерій даних.

5.6.2 Приклади запитів

Для розгляду прикладів створення запитів в режимі конструктора, створимо 2 таблиці: Замовлення і Клієнти.



1. Необхідно відсортувати назви клієнтів за збільшенням. Тоді конструктор запитів матиме наступний вигляд

Поле:	НазваниеКомпании
Имя таблицы:	Клиенты
Сортировка:	по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>
Условие отбора:	
или:	

2. Необходимо вывести поля Кодзаказа і Датаісполнения з таблиці Замовлення і поле Названієкомпанії з таблиці Клієнти і відсортувати Датуісполнения за збільшенням. Тоді запит матиме наступний вигляд:

Поле:	КодЗаказа	НазваниеКомпании	ДатаИсполнения
Имя таблицы:	Заказы	Клиенты	Заказы
Сортировка:			по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
или:			

3. Необходимо підрахувати суму доставок всіх замовлень. Тоді запит має вигляд:

Поле:	СтоимостьДоставки
Имя таблицы:	Заказы
Групповая операция:	Sum
Сортировка:	
Вывод на экран:	<input checked="" type="checkbox"/>
Условие отбора:	
или:	

4. Необходимо підрахувати суму доставки по кожному клієнтові. Тоді запит має вигляд:

Поле:	КодКлиента	СтоимостьДоставки
Имя таблицы:	Заказы	Заказы
Групповая операция:	Группировка	Sum
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

5. Необходимо вывести Названієкомпанії і Суммудоставки, якщо Суммудоставки > 500. Тоді запит має вигляд:

Поле:	НазваниеКомпании	СтоимостьДоставки
Имя таблицы:	Клиенты	Заказы
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	> 500	
или:		

6. Необходимо вывести Названієкомпанії і Датуісполнения, якщо Датуісполнения знаходиться між 01.04.2003 і 30.04.2003. Тоді запит має вигляд:

Поле:	НазваниеКомпании	ДатаИсполнения
Имя таблицы:	Клиенты	Заказы
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Between #01.04.2003# And #30.04.2003#	
или:		

7. Необходимо вывести НазваниеКомпании и СуммуДоставки, якщо НазваниеКомпании починається з букви «а». Тоді запит має вигляд:

Поле:	НазваниеКомпании	СтоимостьДоставки
Имя таблицы:	Клиенты	Заказы
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like "а"	
или:		



8. Вывести НазваниеКомпании з деякого міста. Ім'я міста задавати як параметр. Тоді запит має вигляд:

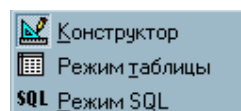
Поле:	Город	НазваниеКомпании
Имя таблицы:	Клиенты	Клиенты
Сортировка:		
Вывод на экран:	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like [City]	
или:		

5.6.3 Створення запитів в режимі SQL

Мова SQL (Structured Query Language) використовується при створенні запитів, а також для оновлення і управління реляційними базами даних, такими як бази дані Microsoft Access. Коли користувач створює запит в режимі конструктора запиту, Microsoft Access автоматично створює еквівалентну інструкцію SQL. Користувач має можливість переглядати і змінювати інструкції SQL в режимі SQL. Зміни, внесені до запиту в режимі SQL, приведуть до відповідних змін в бланку запиту в режимі конструктора. Деякі запити не можуть бути визначені в бланку запиту конструктора. Для створення таких запитів потрібно ввести інструкцію SQL безпосередньо у вікно запиту в режимі SQL.

Для перегляду і зміни інструкції SQL необхідно виконати наступні кроки:

- створити або відкрити існуючий запит.
- Натисніть на панелі інструментів кнопку  (Вигляд) і виберіть Режим SQL .



5.6.4 Інструкції SQL

1. Виведення даних з однієї таблиці

SELECT <имя поля1>,...,<имя поляN>
FROM <имя таблицы>;

Наприклад:

- Вивести всі поля таблиці Клієнти

```
SELECT *
FROM Клиенты;
```

— Вивести назву компаній і місто знаходження цих компаній

```
SELECT НазваниеКомпании, Город
FROM Клиенты;
```

2. Виведення даних з однієї таблиці по деякій умові

```
SELECT <имя поля1>,...,<имя поляN>
FROM <имя таблицы>
WHERE <имя поля> Условие Значение;
```

Наприклад:

— Вивести назву компанії з міста ABC

```
SELECT НазваниеКомпании
FROM Клиенты
WHERE Город = 'ABC';
```

— Вивести код замовлення, код клієнта, вартість доставки, якщо вартість доставки більше 400

```
SELECT КодЗаказа, КодКлиента, СтоимостьДоставки
FROM Заказы
WHERE СтоимостьДоставки>400;
```

— Вивести назву компаній назва "Київ";

```
SELECT НазваниеКомпании
FROM Клиенты
WHERE НазваниеКомпании<"Киев";
```

— Вивести код замовлення і дата виконання замовлення, якщо дата виконання знаходиться між 01.03.2003 і 15.05.2003.

```
SELECT КодЗаказа, ДатаИсполнения
FROM Заказы
WHERE ДатаИсполнения BETWEEN #03/01/03# AND #05/15/03#;
```

— Вивести назву компанії, місто якої починається з букви В

```
SELECT НазваниеКомпании
FROM Клиенты
WHERE Город Like "В*";
```

— Вивести назву компанії, місто якої задається у вигляді параметра

```
SELECT НазваниеКомпании
FROM Клиенты
WHERE Город Like [City];
```

- Вивести код замовлення і вартість доставки, якщо вартість доставки більше параметра, що задається

```
SELECT КодЗаказа, СтоимостьДоставки
FROM Заказы
WHERE СтоимостьДоставки > [Fee];
```

3. Об'єднання записів з однаковими значеннями

```
SELECT <имя поля1>,..., <имя поляN>
FROM <имя таблицы>
WHERE <имя поля> Условие Значение
GROUP BY <имя группируемого поля>;
```

Наприклад:

- Вивести суму доставки по кожному клієнтові

```
SELECT Sum (СтоимостьДоставки) AS Sum
FROM Заказы
GROUP BY КодКлиента;
```

- Вивести код клієнта і кількість замовлень по кожному клієнтові

```
SELECT КодКлиента, Count(КодКлиента) AS [Num of zakaz]
FROM Заказы
GROUP BY КодКлиента;
```

4. Накладення умов на згруповані записи

```
SELECT <имя поля1>,..., <имя поляN>
FROM <имя таблицы>
WHERE <имя поля> Условие Значение
GROUP BY <имя группируемого поля>
HAVING <имя поля> Условие Значение;
```

Наприклад:

- Вивести код клієнта і суму доставки по кожному клієнтові, якщо сума доставки перевищує 100.

```
SELECT КодКлиента, Sum(СтоимостьДоставки)
FROM Заказы
GROUP BY КодКлиента
HAVING Sum(СтоимостьДоставки) > 100;
```

5. Сортуння в порядку зростання або убуння

```
SELECT <имя поля1>,..., <имя поляN>
FROM <имя таблицы>
WHERE <имя поля> Условие Значение
ORDER BY <имя поля1> [ASC | DESC ], <имя поля2> [ASC | DESC ];
```

За умовчанням здійснюється сортування за збільшенням. **ASC** - за збільшенням, **DESC** - по зменшенню.

Наприклад:

- Вивести на екран код замовлення і вартість доставки по всіх замовленнях і відсортувати за збільшенням вартість доставки

```
SELECT КодКлиента, СтоимостьДоставки
FROM Заказы
ORDER BY СтоимостьДоставки;
```

Або

```
SELECT КодКлиента, СтоимостьДоставки
FROM Заказы
ORDER BY СтоимостьДоставки ASC;
```

Контрольні питання:

1. Для чого використовуються запити?
2. Які типи запитів ви знаєте?
3. Як створити запит в Access?
4. Які режими створення запитів ви знаєте?
5. Що таке SQL?

5.7. ACCESS. СТВОРЕННЯ ФОРМ

Способи створення форм. Мастре форм. Створення форм в режимі конструктора.

Розробник, працюючи над створенням бази даних, ретельно продумує структуру всіх таблиць і запитів, які в неї входитимуть. Але подальшим інформаційним заповненням цієї бази, він вже займатися не буде. Це робитимуть спеціальні користувачі БД. Дуже часто вони можуть виявитися людьми з кваліфікацією недостатньою для роботи з базою даних в чистому вигляді. Ось саме для цих людей і для максимального спрощення їх праці - створюються **форми**.

Сама форма є електронним бланком, в якому є ряд поименованных (підписаних) полів для введення інформації. Користувач заповнює ці поля інформацією, і вона автоматично заноситься у відповідні таблиці бази даних.

Після натиснення на кнопку Створити, розміщену на закладці Форми, головного вікна бази даних з'явиться вікно з вибором способів створення форм.

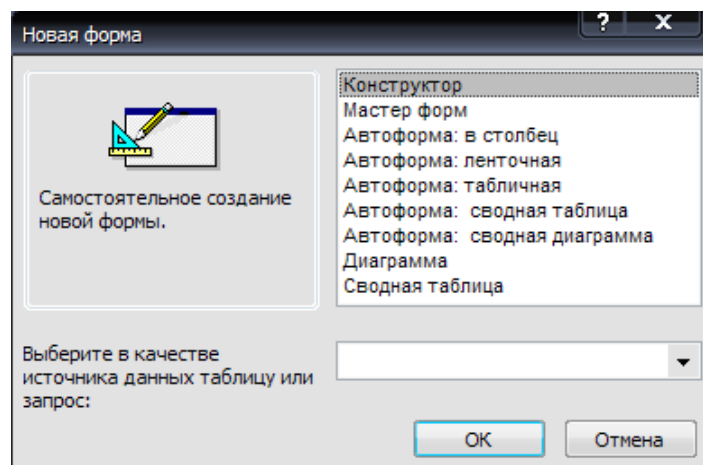


Рис. 5.7.1 Вікно створення форм

Існують наступні способи створення форм:

- **Конструктор** - самостійне створення конструктора, тобто уручну.
- **Майстер форм** - автоматичне створення форми на основі вибраних полів.
- **Автоформа в стовпець** - автоматичне створення форм з полями, розташованими в один стовпець.
- **Автоформа стрічкова** - автоматичне створення стрічкових форм.
- **Автоформа таблична** - автоматичне створення табличних форм.
- **Діаграма** - створення форми з діаграмою.
- **Звідна таблиця** - створення форми із звідною таблицею Excel.

5.7.1 Майстер форм.

Вікно Нова форма, окрім вибору режиму створення форми, просить вказати джерело даних для форми. Це може бути таблиця або запит. Список вже наявних об'єктів даних типів знаходиться в нижній частині вікна. Якщо зараз не вказати джерело даних і натиснути ОК, то майстер не почне лаятися, а просто почне свою роботу. На наступному етапі його роботи буде можливість знову вибрати джерело даних.

Етапи роботи Майстра:

Шаг 1. полягає у виборі полів таблиць, дані, для заповнення яких можна буде вводити в створюваній формі. Виконується це виділенням відповідного пункту в списку Доступні поля і натисненням на одну з двох кнопок переміщення (одинарний значок переносить виділене поле, а подвійний - відразу всі доступні поля). При цьому назва поля переходить в список Вибрані поля і це означає, що воно буде доступне для заповнення на створюваній формі. Якщо помилково було додано непотрібне поле, то його можна перемістити назад, виконавши аналогічні маніпуляції в списку вибраних полів і скориставшись кнопками із значками направленними в протилежному напрямі.

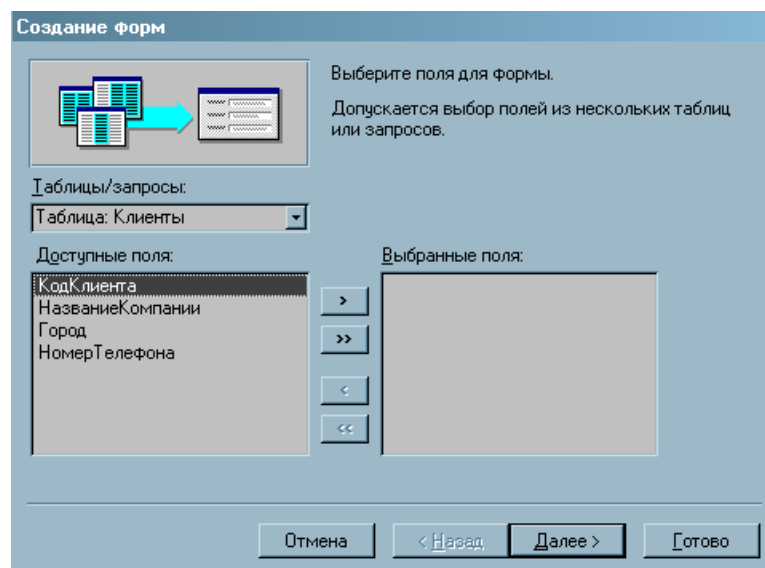


Рис. 5.7.2 Вікно першого кроку Майстра форм

Шаг 2. дозволяє вибрати один з чотирьох можливих видів представлення форми:

1. **У один стовпець** - розміщує поля для введення даних у форму, одне під іншим, утворюючи стовпець. Таким чином, завжди видно тільки одна поточний запис. Перемикання між сусідніми записами проводиться за допомогою кнопок переходу в нижній частині екрану.

2. **Стрічковий** - у формі одночасний відображається ціла група сусідніх записів. Це більш необхідно для роботи із записами зв'язаними між собою. Наприклад, коли для введення наступного значення необхідно знати ряд попередніх.

3. **Табличний** - практично не відрізняється на вигляд від звичайної таблиці. Єдина відмінність полягає в тому, що для заповнення відриті тільки вибрані при створенні форми поля. Всі останні залишаються недоступними з даної форми.

4. **Вирівняний** - створює форму, поля на якій займають практично все вільне місце і при цьому вирівняні по відношенню один до одного. Іноді корисна для створення форм, що містять велику кількість полів, які необхідно умістити на екрані, без виконання скролінгу (прокрутки).

Шаг 3. На цьому кроці буде запропоновано вибрати стиль оформлення форми. Це фоновий малюнок або набір певних квітів елементів. При виборі будь-якого з можливих варіантів у вікні попереднього перегляду відразу буде створений приклад для наочного відображення стилю.

Шаг 4. Це останній крок, на якому майстер запитає ім'я форми, під яким вона буде збережена в базі, і запропонує два варіанти подальших дій

- Відкриття форми - приведе до запуску форми, після чого з її допомогою можна почати процес додавання нових записів в базу і редагування тих, що вже є.
- Зміна макету - відкриє створену форму в режимі конструктора. Що дасть можливість змінити і відредагувати структуру всіх полів форми.

5.7.2 Режим Конструктора

Структура форми в режимі конструктора розділена на три окремі розділи: заголовок форми, область даних і примітка форми.

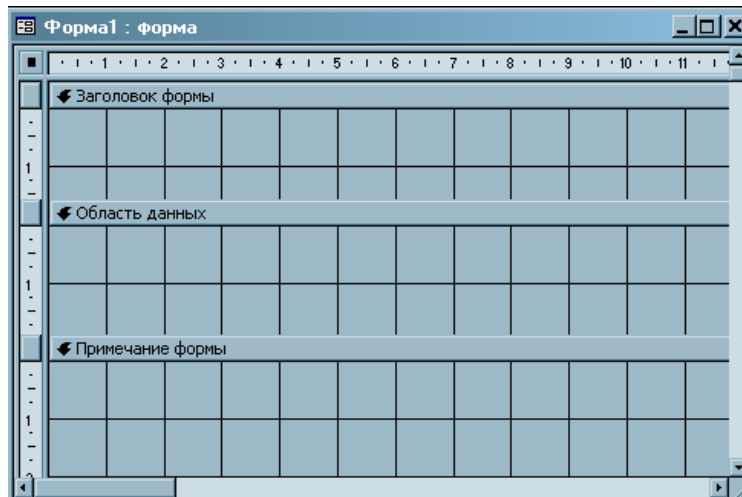


Рис. 5.7.3 Структура форми в режимі Конструктора

Заголовок форми - може містити елементи, необхідні для зовнішнього оформлення або просто для прикраси форми. Вони будуть присутні на формі завжди, незалежно від того, створюється новий запис або редагується та, що вже існує. Це може бути напис, малюнок, логотип фірми або інший графічний елемент.

Область даних - це основний розділ будь-якої форми. Без нього немає форми, оскільки вона містить елементи управління. Тут здійснюються всі динамічні процеси, що відбуваються при роботі форми. У нашому випадку там будуть тільки два типи елементів управління - зв'язане поле і приєднаний напис.

У **зв'язане поле** здійснюється введення даних, які синхронно поступають в однойменне поле відповідної таблиці, на основі якої створена форма. Поле форми і поле таблиці зв'язані між собою.




Приєднаний напис це статичний елемент, що містить текстові пояснення. Вона переміщається по полю форми тільки разом зі своїм елементом управління (приєднана до нього).


















Примітка форми - схоже по своєму призначенню на заголовок, але знаходиться в нижній частині форми. Дуже часто містить інструкцію по заповненню форми або пам'ятку з вказівкою необхідній довідковій інформації.

Фоновий малюнок, що знаходиться під елементами управління і розграфлений сіткою, показує реальні розміри робочої області форми. Ті, в яких вона виводиться на екран. При наведенні на межі областей, покажчик міняє свою форму, і будь-яка з меж у цей момент може бути переміщена методом перетягання. Розташування елементів теж можна змінити, провівши клацання в полі елемента і не відпускаючи кнопку миші перетягнув його на нове місце. Приєднаний напис при цьому теж автоматично переміщатиметься.

Також за допомогою контекстного меню, викликаного до області форми, можуть бути додані верхній і нижній колонтитули.

Існують наступні елементи управління:

1.  – вибір об'єктів.
2.  – майстер, засіб для ставлення питань і на основі відповідей будує відповідний об'єкт.
3.  – напис, призначена для виведення описового тексту.



4.  – поле, використовується для відображення, введення, зміни даних в джерелі записів форми для виведення результатів обчислень, а також для прийому даних, що вводяться користувачем.
5.  – група перемикачів, використовується для розміщення набору прапорців, перемикачів або вимикачів, що представляє набір альтернативних значень.
6.  – вимикач, використовується як окремий елемент управління, що приймає дії користувача в спеціальному діалоговому вікні.
7.  – перемикач, використовується як окремий елемент управління, що приймає дії користувача в спеціальному діалоговому вікні.
8.  – прапорець, використовується як окремий елемент управління, що приймає дії користувача в спеціальному діалоговому вікні.
9.  – поле із списком, складовий елемент управління, об'єднуючий поле і список, що розкривається. У таке поле можна ввести значення або вибрати його із списку.
10.  – список, створює список, що допускає прокрутку. Вибране в списку значення можна ввести в новий запис або використовувати для зміни в існуючому записі.
11.  – кнопка, використовується для виконання дій.
12.  – малюнок, використовується для відображення незмінного малюнка.
13.  – вільна рамка об'єкту, використовується для відображення вільного об'єкту OLE.
14.  – приєднана рамка об'єкту, використовується для відображення об'єктів OLE, таких як набір малюнків. При переході на новий запис вводяться нові об'єкти.
15.  – розриви сторінки, використовується для початку нового екрану у формі.
16.  – набір вкладок, використовується для створення форми з декількома вкладками.
17.  – підлегла форма, використовується для виводу у формі даних з декількох таблиць.
18.  – лінія, використовується для відділення даних.
19.  – прямокутник, використовується для просторового угруповання споріднених даних.
20.  – додаткові елементи.

5.7.3 Заповнення Форми в режимі конструктора

Додавання напису в заголовок форми розглянемо на конкретному прикладі. Допустимо, нам необхідно створити форму для заповнення таблиця Клієнти(Кодклієнта, Названієкомпанії, Місто, Номертелефона).

У заголовок форми додамо напис «Додавання нового клієнта».

Для цього необхідно:

1. Збільшити область заголовка. Для цього перетягнете вниз межу розділу заголовка і області даних. Виділіть достатньо місця для створення необхідного напису.
2. Знайдіть на панелі елементів об'єкт Напис, призначений для створення текстових написів  (якщо панель елементів відсутня, то її можна включити через Вигляд - Панель елементів, головного меню програми ).
3. Клацання на елементі Напис, міняє покажчик на заголовну букву А і маленький плюс, за допомогою якого можна вказати місце на формі, де розташовуватиметься створюваний напис. При наборі тексту немає необхідності піклується про його форматування, та і первинне

місце не має особливого значення. Все це можна змінити після завершення введення і натиснення клавіші Enter.

4. Для форматування об'єкту його необхідно виділити. У такому режимі об'єкт можна перетягнути на інше місце, змінити розміри, змінити формат тексту і т.п.

В результаті цих дій отримаємо заголовок створюваної форми.

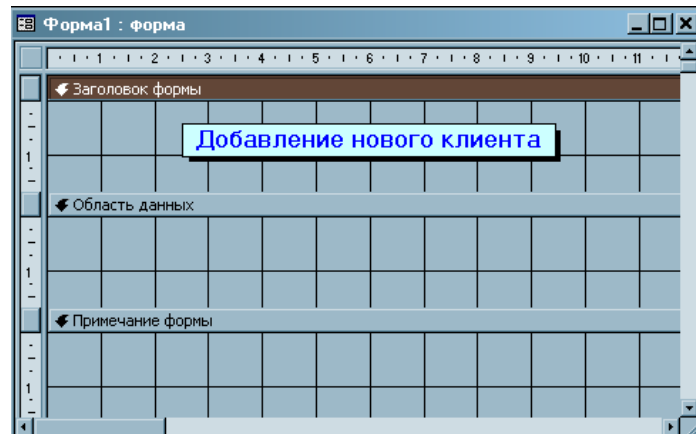



Рис. 5.7.4. Додавання заголовка у форму.

Додаткові елементи форми, такі як заголовок, не пов'язані ні з одним з полів, який або таблиці в базі даних. Тому такий елемент ще прийнято називати вільним полем.

Текст, вказаний в ньому при створенні, залишається незмінним незалежно від того, який запис з БД ми проглядаємо, редагуємо або додаємо за допомогою форми в даний момент часу. Це статичний текст.

Форма насамперед призначена для роботи з полями таблиці. Тому на кожній формі обов'язково присутні елементи управління, звані зв'язаними полями. Це інтерфейс між користувачем і формою, і далі між формою і таблицею. Такий зв'язок дуже важливий і має бути строго зафіксована.

Наступний крок - це додавання полів введення або зв'язаних полів в область даних. Для цього необхідно:

1. При натисненні на  (Поле), покажчик міняє свій вигляд на зображення поля введення і редагування з маленьким плюсом, за допомогою якого можна виділити місце для майбутнього елемента на формі. В результаті в області даних форми з'явиться вільний зв'язаний елемент. Слово Вільний означає, що зв'язок для даного елемента поки не визначений.

2. Разом з появою вільного зв'язаного елемента автоматично з'являється додатковий елемент управління, званий **приєднаний напис**. Вона переміщається формою разом зі своїм зв'язаним полем і тим самим утворює з ним єдине ціле. Поле0 - це його назва з порядковим номером, що служить для іменування елементів на формі.

Розміри кожного з елементів можна змінити, розтягуючи поля за маркери розміру. При необхідності "відірвати" поле від прикріпленого напису, скористайтесь спеціальним маркером, розташованим в лівому верхньому кутку кожного з елементів. При наведенні на нього покажчик міняється на зображення долоні з вказівним пальцем. У цей момент зв'язане поле можна відокремити від приєданого напису (або навпаки напис від поля) і переміщати за формою самостійного.

Для точнішого позиціонування елементів на формі (з точністю до одного пікселя), служать курсорні клавіші клавіатури, використовувані спільно з CTRL і SHIFT. При утриманні клавіші SHIFT відбувається зміна розмірів виділеного елемента, а при натиснутому CTRL - змінюється положення елемента на формі.

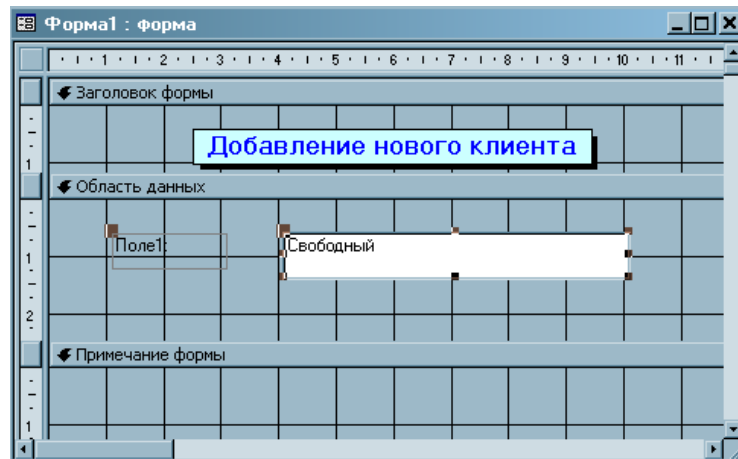


Рис. 5.7.5 Додавання вільно зв'язаного елементу і напису.

Для зміни тексту прикріпленому напису необхідно перемістити покажчик в її центральну область (при цьому він міняється на текстовий курсор) і провести клацання. Поле стане доступним для редагування. Змініте привласнений за умовчанням напис на необхідну і натисніть **Enter**. По правій кнопці миші буде доступне додаткове контекстне меню, що дозволяє вибрати тип оформлення і кольору елементів.

3. Тепер необхідно вказати для зв'язаного поля, з якою таблицею воно працюватиме. Для цього виділяємо його і натискаємо праву кнопку миші. Тепер нас цікавить пункт Властивості і закладка Дані. На ній можна набудувати зв'язок з таблицею. Якщо при створенні форми було вказано джерело, то поля відповідної таблиці будуть доступні для вибору в графі Дані.

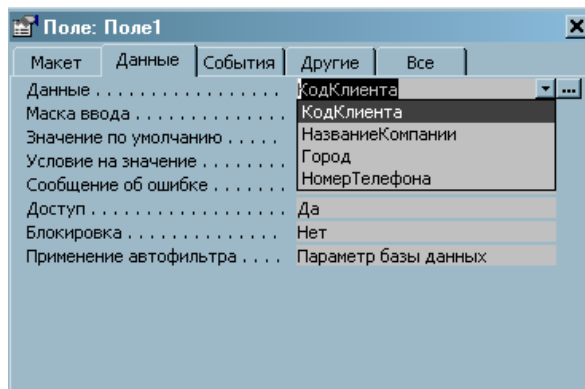


Рис. 5.7.6 Вибір поля таблиці для зв'язаного поля.

Поле Доступ у властивостях поля, показує, чи буде воно активно на формі чи ні. Неактивне поле виводиться у втопленому і затіненому вигляді. Властивість Блокування приводить до заборони або дозволу редагування інформації в полі. За умовчанням редагування і доступ до даних дозволений. Змініте ці властивості на свій розсуд.

4. Якщо при створенні форми джерело не було вказане, то необхідно буде використовувати Будівника виразів, що відкривається за допомогою кнопки із зображенням трикрапки в пункті Властивості, закладка Дані і властивості Дані. У додатковому вікні необхідно уручну вказати шлях до необхідної таблиці і вибрати поле, що цікавить Вас.

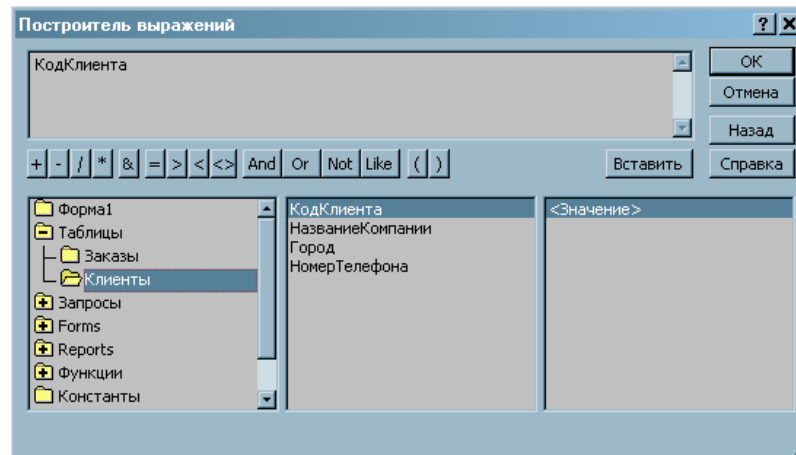



Рис. 5.7.7 Вікно Будівника виразів.

5.7.4 Виклик однієї форми з іншої

Хай створена форми для заповнення таблиць Клієнти і Замовлення. Необхідно додати у форму для введення замовлень можливість з неї відкривати форму для заповнення клієнтів. Для виконання цього завдання скористаємося елементом управління Кнопка .

Для цього необхідно виконати наступні дії:

1. Відкрийте форму для замовлень в режимі конструктора.
2. Виберіть в панелі елементів управління властивість Кнопка і помістіть її на форму.

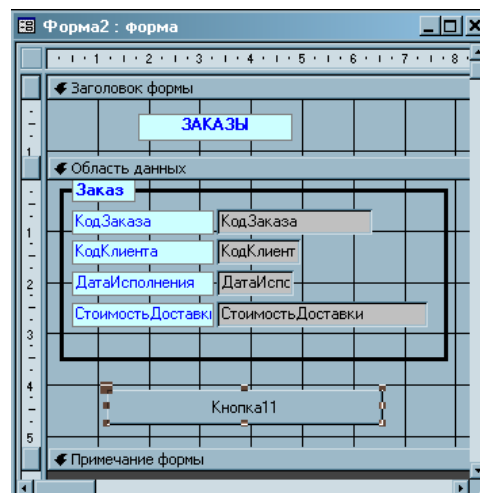


Рис. 5.7.8 Додавання кнопки у форму

3. Виберіть пункт Властивості в контекстному меню для створеної кнопки. Потім у вкладці Події властивість Натиснення кнопки. В результаті з'явиться вікно Будівник з трьома пунктами для вибору: вирази, макроси, програми. Нас цікавитимуть Макроси. В результаті з'явиться вікно з проханням назвати макрос, залишимо запропоноване ім'я Макрос 1. Будівника макросів має наступний вигляд:

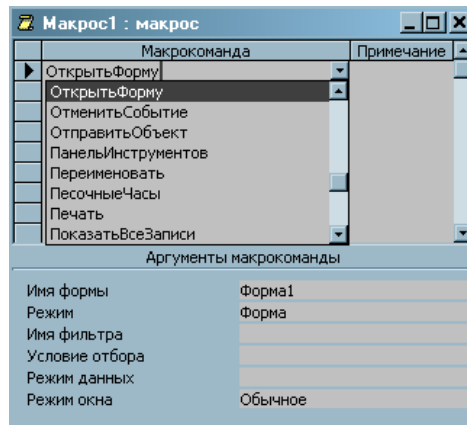


Рис. 5.7.9 Вікно для побудови макросів

4. У полі Макрокоманда з випадного списку вибираємо Відкрити Форму, а у властивостях Аргументи макрокоманди, в ім'я форми вибираємо ім'я необхідної форми. Закриваємо будівника макросів.

5. Збережете форму і перейдіть в режим форми.
Результатом натиснення на кнопку буде виклик форми для введення клієнтів.

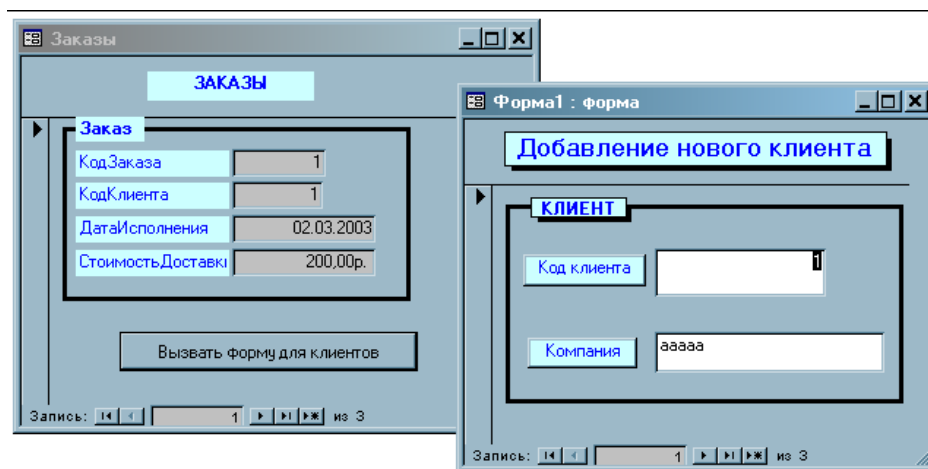


Рис. 5.7.10 Виклик однієї форми з іншої.

5.7.5 Додавання в Примітку форми поточної дати і час

Відкрийте форму Замовлення в режимі конструктора. При необхідності збільште область примітки форми. Додайте в цю область елемент Поле. У написі зітріть весь текст і новий не вводите. Зробіть активним елемент з ім'ям Вільний (тобто необхідно клацнути на нім мишкою). По правій клавіші миші викличте його контекстне меню і в нім виберіть пункт Властивості.

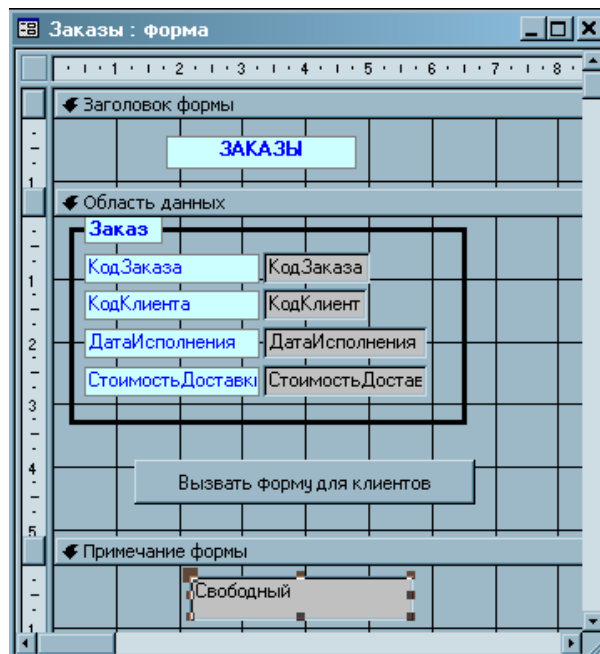


Рис. 5.7.11 Додавання поля для дати і часу в примітку форми

У вікні властивостей виберіть закладку Дані. У ній пункт Доступ змініте на «Відео». Потім перейдіть в полі Дані, натисніть на кнопку . У вікні, що з'явилось, виберіть Вираз. У вікні будівника виразів вибираєте теку Функції, в ній теку вбудовані функції. У сусідньому полі вибираєте пункт Дата/Час. Потім в останньому полі вибираєте функцію *Now* і двічі клацаєте по ній. Для завершення роботи з будівником виразів натисніть ОК.

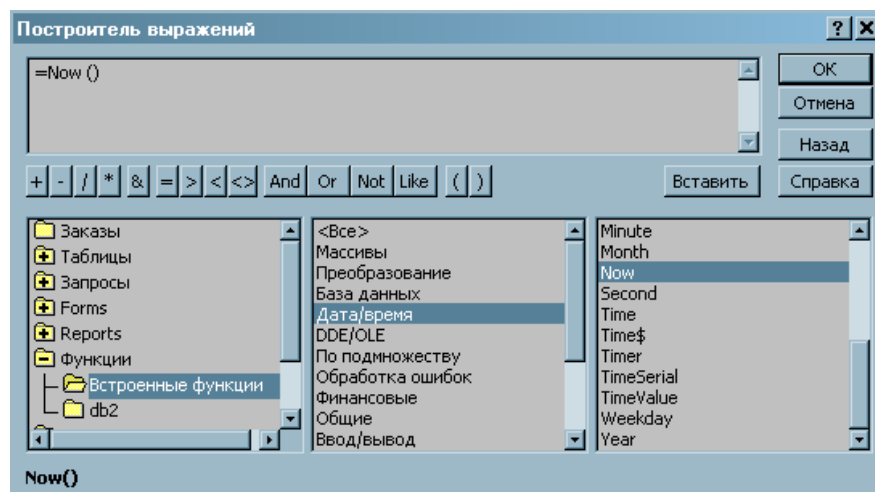


Рис. 5.7.12 Вибір функції дати і часу

В результаті після запуску форми в області примітки висвічуватиметься поточна дата і час.

Рис. 5.7.13 Форма з поточною датою і часом.

Якщо немає необхідності виводити дату і час разом, то замість функції `Now`, можна використовувати наступні функції:

- **`Date()`** – для виведення системної дати
- **`Time()`** – для виведення системного часу

Контрольні питання:

1. Що таке форми? Для чого вони використовуються?
2. Які режими створення форм ви знаєте?
3. Які чотири можливих видів представлення форми ви знаєте?
4. Які області є в конструкторі форм? Для чого вони призначенні?
5. Як додати необхідний елемент управління до форми?
6. Що таке будівник виразів?

5.8. ACCESS. СТВОРЕННЯ ЗВІТІВ

Способи створення звітів. Створення звіту в режимі матсера. Створення звіту в режимі конструктора.

Звіти багато в чому схожі на форми і теж дозволяють отримати результати роботи запитів в наочній формі, але тільки не на екрані, а у вигляді роздруку на принтері. Таким чином, в результаті роботи звіту створюється паперовий документ.

За відсутності принтера звіти створювати все-таки можна. Досить виконати програмну установку за допомогою команди операційної системи: Пуск □ Налаштування □ Принтери □ Установка принтера, після чого зареєструвати драйвер принтера, або узявши його з гнучкого диска, або вибравши один з драйверів, що додаються до самої операційної системи.

Велика частина того, що було сказано про форми, відноситься і до звітів. Так звіти можна створити декількома способами

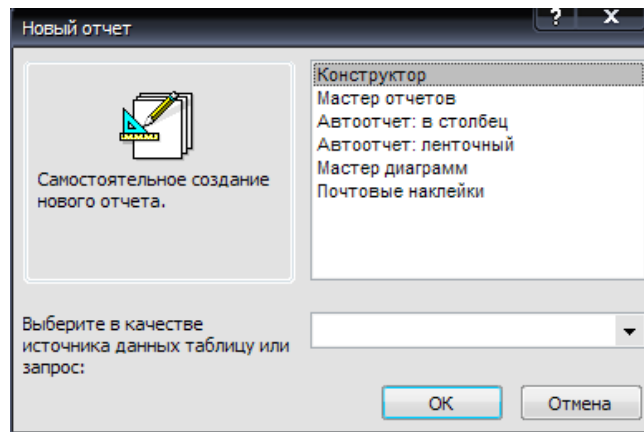


Рис.5.8.1 Способи створення звітів

Режими створення звітів:

- Конструктор.
- Майстер звітів
- Автозвіт: у стовпець
- Автозвіт: стрічковий
- Майстер діаграм
- Поштові наклейки - створення звіту, що відформатував для друку поштових наклеюк.

5.8.1 Кроки створення звіту в режимі Майстра звітів

- Шаг 1. Виберіть для створення режим Майстер звітів і таблицю для якої будуватиметься звіт.
- Шаг 2. Вибираються поля заданої таблиці, які будуть поміщені в звіт.
- Шаг 3. Указуються поля по яких буде проводиться угруповання
- Шаг 4. Вибір сортування полів
- Шаг 5. Вибирається вид макету для звіту.
- Шаг 6. Вибирається стиль відображення звіту
- Шаг 7. Указується ім'я звіту і подальші дії з ним.

Після проходження всіх кроків Майстра буде створений паперовий макет звіту, готовий до друку. Якщо необхідно відкоректувати його, то вікно з макетом закривається і потім звіт відкривається в режимі Конструктора.

5.8.2 Створення звітів в режимі Конструктора

Як і форми, звіти складаються з розділів, а розділи можуть містити елементи управління. Але, на відміну від форм, розділів в звітах стільки ж, а елементів управління, менше.

Структура звіту складається з п'яти розділів: заголовка звіту, верхнього колонтитулу, області даних, нижнього колонтитулу і примітки звіту.

Розділ заголовка служить для друку загального заголовка звіту.

Розділ верхнього колонтитулу можна використовувати для друку підзаголовків, якщо звіт має складну структуру і займає багато сторінок. Тут можна також поміщати і колонцифри (номери сторінок), якщо це не зроблено в нижньому колонтитулі.

В області даних розміщують елементи управління, пов'язані з вмістом полів таблиць бази. У ці елементи управління видаються дані з таблиць для друку на принтері. Порядок розміщення і вирівнювання елементів управління той же, що і при створенні структури форм.

Розділ нижнього колонтитулу використовують для тих же цілей, що і розділ верхнього колонтитулу.

Розділ примітки використовують для розміщення додаткової інформації.

Звіт по таблиці Клієнти в режимі Конструктора виглядатиме таким чином:

Рис. 5.8.2 Звіт в режимі Конструктора

У нижній (верхній) колонтитул можна додати наступні вирази:

- Виведення номера сторінки – ="Страница" & Page
- Число сторінок – =Pages
- Сторінка N з M – = " Страница " & Page & " из " & Pages
- Поточна дата – = Date()
- Поточна дата і час – = Now()
- Поточний користувач – = CurrentUser()

Контрольні питання:

1. Що таке звіт? Для чого він використовується?
2. Які режими створення звітів ви знаєте?
3. З яких п'яти розділів складається структура звіту у режимі Конструктора?
4. Для чого призначен кожен розділ?

6. Комп'ютерні мережі

6.1. Введене

Комп'ютерна мережа (обчислювальна мережа, мережа передачі даних) - система зв'язку комп'ютерів і/або комп'ютерного устаткування (сервери, маршрутизатори і інше устаткування, канали зв'язку). Для передачі інформації можуть бути використані різні фізичні явища, як правило - різні види електричних, світлових сигналів або електромагнітного випромінювання.

Розрізняють два поняття мережі : **комунікаційна мережа і інформаційна мережа.**

Комунікаційна мережа призначена для передачі даних, також вона виконує завдання, пов'язані з перетворенням даних. Комунікаційні мережі розрізняються за типом використовуваних фізичних засобів з'єднання.

Інформаційна мережа призначена для зберігання інформації і складається з інформаційних систем. На базі комунікаційної мережі може бути побудована група інформаційних мереж.

Під інформаційною системою слід розуміти систему, яка є постачальником або споживачем інформації. Іншими словами - об'єкт, здатний здійснювати зберігання, обробку або передачу інформації. До складу інформаційної системи входять: комп'ютери, програми, користувачі і інші складові, призначені для процесу обробки і передачі даних. Надалі інформаційна система, призначена для вирішення завдань користувача, називатиметься - робоча станція (client). Робоча станція в мережі відрізняється від звичайного персонального комп'ютера (ПК) наявністю мережевої карти (мережевого адаптера), каналу для передачі даних і мережевого програмного забезпечення.

Під каналом зв'язку слід розуміти шлях або засіб, по якому передаються сигнали.

Канали зв'язку (data link) створюються по лініях зв'язку за допомогою мережевого устаткування і фізичних засобів зв'язку. Фізичні засоби зв'язку побудовані на основі витих пар, коаксіальних кабелів, оптичних каналів або ефіру. Між взаємодіючими інформаційними системами через фізичні канали комунікаційної мережі і вузли комутації встановлюються логічні канали.

Логічний канал - це шлях для передачі даних від однієї системи до іншої. Логічний канал прокладається по маршруту в одному або декількох фізичних каналах. Логічний канал можна охарактеризувати, як маршрут, прокладений через фізичні канали і вузли комутації.

Інформація в мережі передається блоками даних по процедурах обміну між об'єктами. Ці процедури називають протоколами передачі даних. Протокол - це сукупність правил, що встановлюють формат і процедури обміну інформацією між двома або декількома пристроями. Завантаження мережі характеризується параметром, званим трафіком. Трафік (traffic) - це потік повідомлень в мережі передачі даних. Під ним розуміють кількісний вимір у вибраних точках мережі числа блоків даних і їх довжини, що проходять, виражене в бітах в секунду.

Істотний вплив на характеристику мережі чинить метод доступу. Метод доступу - це спосіб визначення того, яка з робочих станцій зможе наступною використовувати канал зв'язку і як управляти доступом до каналу зв'язку (кабелю).

У мережі усі робочі станції фізично сполучені між собою каналами зв'язку по певній структурі, званою топологією. Топологія - це опис фізичних з'єднань в мережі, що вказує які робочі станції можуть зв'язуватися між собою. Тип топології визначає продуктивність, працездатність і надійність експлуатації робочих станцій, а також час звернення до файлового сервера. Залежно від топології мережі використовується той або інший метод доступу.

Склад основних елементів в мережі залежить від її архітектури. Архітектура - це концепція, що визначає взаємозв'язок, структуру і функції взаємодії робочих станцій в мережі. Вона передбачає логічну, функціональну і фізичну організацію технічних і програмних засобів мережі. Архітектура визначає принципи побудови і функціонування апаратного і програмного забезпечення елементів мережі.

6.2. Класифікації мереж

Для класифікації комп'ютерних мереж використовуються різні ознаки, вибір яких полягає в тому, щоб виділити з існуючого різноманіття такі, які дозволили б забезпечити цій класифікаційній схемі наступні обов'язкові якості, :

- можливість класифікації усіх, як існуючих, так і перспективних, комп'ютерних мереж;
- диференціацію істотно різних мереж;
- однозначність класифікації будь-якої комп'ютерної мережі;
- наочність, простоту і практичну доцільність класифікаційної схеми.

Певну невідповідність цих вимог один з одним робить завдання вибору раціональної схеми класифікації комп'ютерної мережі досить складною і такою, що не має до теперішнього часу однозначного рішення. В основному комп'ютерні мережі класифікують за ознаками структурної і функціональної організації.

6.2.1. Класифікація за розміром охопленої території :

Мережа, що використовує людське тіло (HAN, Human Area Network). Human Area Network - комп'ютерна мережа де як середовище передачі даних використовується людське тіло. Як носій інформації використовується слабе електричне поле на поверхні людського тіла. Передавач у визначеному місці в малій мірі змінює його, приймач реєструє ці зміни на поверхні тіла.

Персональна мережа (PAN, Personal Area Network). Personal Area Network - це мережа, побудована "навколо" людини. Ці мережі покликані об'єднувати усі персональні електронні пристрої користувача (персональні настільні комп'ютери, ноутбуки, телефони, кишенькові персональні комп'ютери, смартфони, гарнітури и т.п.). Прикладом таких мереж є сісти на основі технології Bluetooth. Параметри PAN :

- Мале число абонентів
- Некритичність до напруження на відмову.
- Усі пристрої входять в PAN -сеть можна контролювати.
- Вузький радіус дії (30 метрів).
- Мережа повинна підтримувати до 8 учасників.
- Немає арбітражу середовища, тобто хто і як може працювати з цією мережею ніяк не контролюється, немає централізованого управління такою мережею.

Локальна мережа (LAN, Local Area Network) Локальна обчислювальна мережа (ЛВС) - комп'ютерна мережа, що покриває зазвичай відносно невелику територію або невелику групу будівель (будинок, офіс, фірму, інститут). Також існують локальні мережі, вузли яких рознесені географічно на відстані більше 12 500 км (космічні станції і орбітальні центри). Незважаючи на такі відстані, подібні мережі все одно відносять до локальних.

Також до цієї категорії мереж можна віднести HomePNA (англ. Home Phoneline Networking Alliance, HPNA). HPNA - об'єднана асоціація некомерційних промислових компаній, які просувають і стандартизують технології домашніх мереж за допомогою існуючих в будинках коаксіальних кабелів і телефонних ліній. Серед компаній-покровителів HPNA, які встановлюють курс організації, можна виділити AT&T, 2Wire, Motorola, CooperGate, Scientific Atlanta і K - Micro. HPNA створює промислові специфікації, які потім стандартизуються Міжнародним Союзом Телекомунікацій (International Telecommunication Union ITU), провідною світовою організацією стандартизації в області телі і радіо-комунікацій. HPNA також просуває технології, тестує і сертифікує членські продукти як схвалені HomePNA. HomePNA 3.1 один з нового покоління стандартів домашніх мереж, розроблений для нових "розважальних" застосувань, таких як IPTV (інтернет-телебачення), які припускають наявність високої і стійкої продуктивності в цілому будинку.

Технологія цього типу забезпечує додаткові можливості, такі як гарантована якість обслуговування (Quality of Service QoS) і використовується більшістю провайдерів (організацією, що надають доступ до таких мереж і займаються їх обслуговуванням) для забезпечення комерційного сервісу "triple play" (відео, звук і інформація). HomePNA 3.1

використовує частоти вище за тих, що використовуються технологіями ADSL, ISDN (технології передачі даних по телефонних лініях) і телефонними дзвінками на лінії і нижче за тих, що використовуються для телетрансляції і супутникової телетрансляції DVB - S по коаксіальному (телевізійному) кабелю, тому HomePNA 3.1 може співіснувати з цими сервісами в одних дротах. HomePNA 3.1 був розроблений як для збільшення функціональності в коаксіальних дротах і розширення їх мережевих можливостей, так і для подолання деяких обмежень телефонних мереж.

Вимоги для HomePNA 3.1:

- Стандартний телефонний або коаксіальний кабель.
- Устаткування, сертифіковане HomePNA.

Переваги HomePNA 3.1:

- Не вимагається проведення нових кабелів в будинок.
- Робота існуючих сервісів - телефону, факсу, DSL, супутникового телебачення не порушиться, завдяки тому, що HomePNA працює з різними частотами на одному коаксіальному або телефонному кабелі.
- Новітня продукція пропонує швидкість передачі даних до 320Мб/з, забезпечуючи можливість підтримки високочіткого телесигналу (High Definition TV HDTV) і стандартного телевізійного сигналу (Standart Definition TV SDTV).
- Гарантована якість обслуговування QoS, усуває мережеві "колізії". Це дозволяє потокам інформації в реальному часі, таким як IPTV, бути доставленими до клієнта без переривань.
- Максимальна кількість пристроїв, що підключаються, - 64.
- Пристрої можуть бути розташовані на відстані 300м. один від одного на телефонній лінії і на відстані більше кілометра один від одного на коаксіальному кабелі. Для будинків це більш ніж достатньо.
- Використовуються стандартні драйвера Ethernet, що дозволяє легко додавати будь-яку продукцію з Ethernet -портом, не торкаючись операційної системи.
- Необхідне устаткування має невисоку вартість.
- Розробляються нові технології, такі як 802.11 Wi - Fi, для створення змішаних дротяних/безпроводних домашніх мереж.
- Провайдери можуть надавати послуги телефону, інтернету і цифрового телебачення одним пакетом, за допомогою устаткування, сертифікованого HomePNA.
- Готельна індустрія розглядає HomePNA як ефективну дорогу опцію.
- Технологія працює в багатоквартирних будинках, надаючи сервіс "triple play" в квартири.

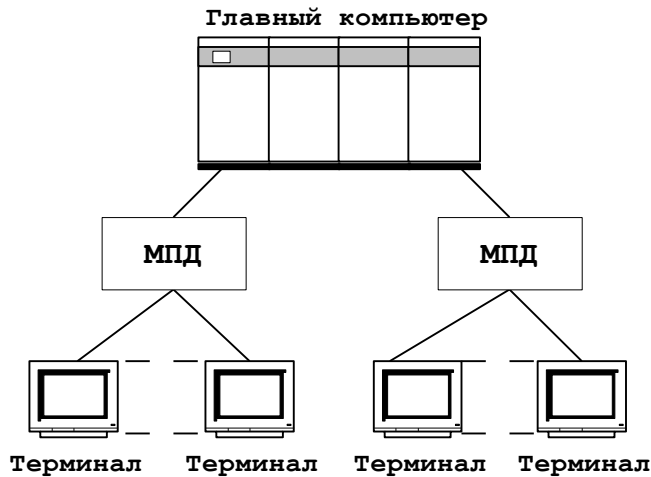
Глобальна обчислювальна мережа, ГВС (англ. Wide Area Network, WAN) є комп'ютерною мережею, що охоплює великі території і що включає десятки і сотні тисяч комп'ютерів. ГВС служать для об'єднання розрізнених мереж так, щоб користувачі і комп'ютери, де б вони не знаходилися, могли взаємодіяти з усіма іншими учасниками глобальної мережі. Деякі ГВС побудовані виключно для приватних організацій, інші є засобом комунікації корпоративних ЛВС з глобальною мережею Інтернет або за допомогою Інтернет з видаленими мережами, що входять до складу корпоративних. Частіше усього ГВС спирається на виділені лінії, на одному кінці яких маршрутизатор підключається до ЛВС, а на іншому концентратор зв'язується з іншими частинами ГВС.

Глобальні мережі відрізняються від локальних тим, що розраховані на необмежене число абонентів і використовують, як правило, не занадто якісні канали зв'язку і порівняно низьку швидкість передачі, а механізм управління обміном, у них в принципі не може бути гарантовано швидким. У глобальних мережах не набагато важливіша якість зв'язку, а сам факт її існування. Правда, зараз вже не можна провести чітку і однозначну межу між локальними і глобальними мережами. Більшість локальних мереж мають вихід в глобальну мережу, але характер переданої інформації, принципи організації обміну, режими доступу, до ресурсів усередині локальної мережі, як правило, сильно відрізняються від тих, що прийнято в глобальній мережі. І хоча усі комп'ютери локальної мережі в даному випадку включені також і

в глобальну мережу, специфіку локальної мережі це не відмінняє. Можливість виходу в глобальну мережу залишається усього лише одним з ресурсів, поділені користувачами локальної мережі.

6.2.2. Класифікація за типом функціональної взаємодії

Архітектура термінал - головний комп'ютер (terminal - host computer architecture) - це концепція інформаційної мережі, в якій уся обробка даних здійснюється одним або групою головних комп'ютерів.



Мал. 6.1 Архітектура термінал - головний комп'ютер

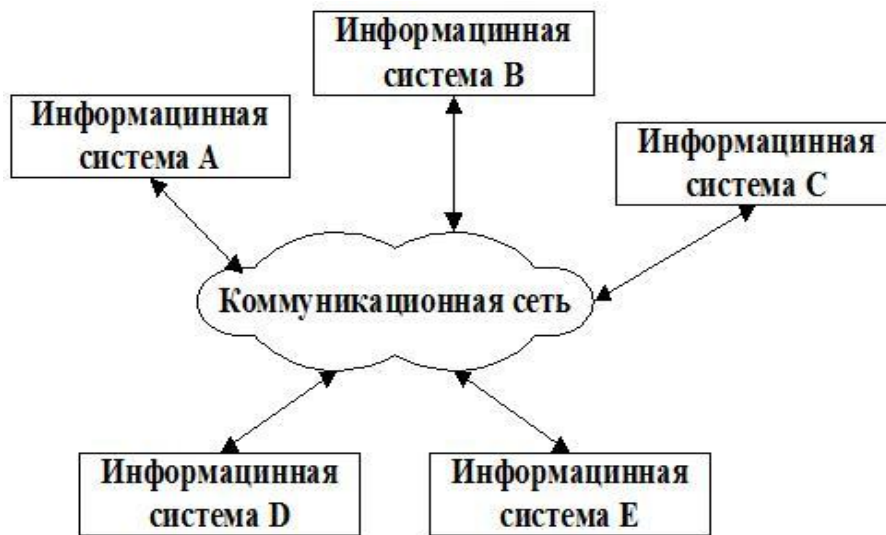
Дана архітектура припускає два типи устаткування :

- Головний комп'ютер, де здійснюється управління мережею, зберігання і обробка даних.
- Термінали, призначені для передачі головному комп'ютеру команд на організацію сеансів і виконання завдань, введення даних для виконання завдань і отримання результатів.

Головний комп'ютер через мультиплексори передачі даних (мультиплексор передачі даних (МПД) - пристрій, який один фізичний канал представляє у вигляді декількох незалежних один від одного логічних каналів) взаємодіють з терміналами, як представлено на мал. 6.1.

Однорангова архітектура (peer - to - peer architecture) - це концепція інформаційної мережі, в якій її ресурси розосереджені по усіх системах. Ця архітектура характеризується тим, що в ній усі системи рівноправні.

До однорангових мереж відносяться малі мережі, де будь-яка робоча станція може виконувати одночасно функції файлового сервера і робочої станції. У однорангових ЛВС дисковий простір і файли на будь-якому комп'ютері можуть бути загальними. Щоб ресурс став загальним, його необхідно віддати в загальне користування, використовуючи служби видаленого доступу мережесхем однорангових операційних систем. Залежно від того, як буде встановлений захист даних, інші користувачі зможуть користуватися файлами відразу ж після їх створення. Однорангові ЛВС досить хороші тільки для невеликих робочих груп.



Мал. 6.2 Однорангова архітектура

Однорангові ЛВС є найбільш легким і дешевим типом мереж для установки. Вони на комп'ютері вимагають, окрім мережевої карти і мережевого носія, тільки операційної системи (наприклад Windows XP). При з'єднанні комп'ютерів, користувачі можуть надавати ресурси і інформацію в спільне користування.

Однорангові мережі мають наступні переваги:

- вони легкі в установці і налаштуванні;
- окремі ПК не залежать від виділеного сервера;
- користувачі в змозі контролювати свої ресурси;
- мала вартість і легка експлуатація;
- мінімум устаткування і програмного забезпечення;
- немає необхідності в адміністраторові;
- добре підходять для мереж з кількістю користувачів, що не перевищує десяти.

Проблемою однорангової архітектури є ситуація, коли комп'ютери відключаються від мережі. У цих випадках з мережі зникають види сервісу, які вони надавали. Мережеву безпеку одночасно можна застосувати тільки до одного ресурсу, і користувач повинен пам'ятати стільки паролів, скільки мережевих ресурсів. При діставанні доступу до ресурсу, що розділяється, відчувається падіння продуктивності комп'ютера. Істотним недоліком однорангових мереж є відсутність централізованого адміністрування.

Використання однорангової архітектури не виключає застосування в тій же мережі також архітектура "термінал - головний комп'ютер" або архітектура "клієнт - сервер".

Архітектура клієнт - сервер (client - server architecture) - це концепція інформаційної мережі, в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів (мал. 1.5). Дана архітектура визначає два типи компонентів : сервери і клієнти.

Сервер - це об'єкт, що надає сервіс іншим об'єктам мережі по їх запитах. Сервіс - це процес обслуговування клієнтів.



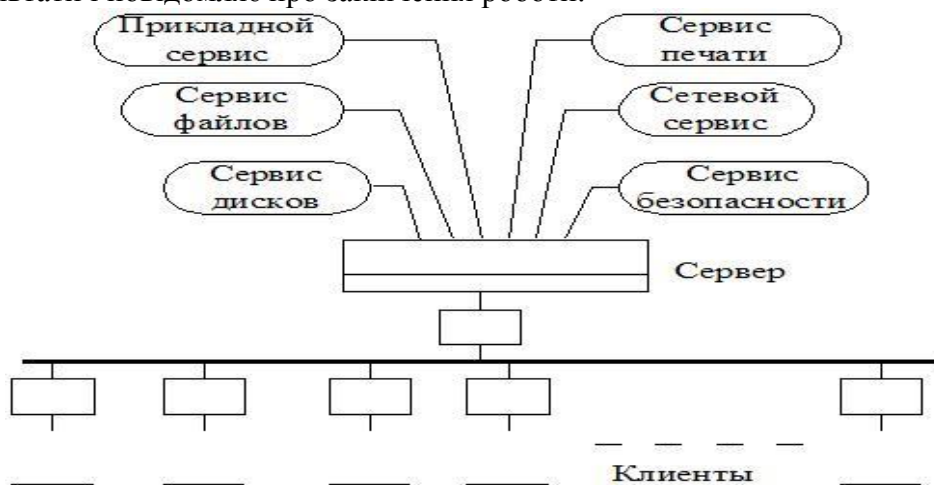
Мал. 6.3 Архітектура клієнт - сервер

Сервер працює по завданнях клієнтів і управляє виконанням їх завдань. Після виконання кожного завдання сервер посилає отримані результати клієнтові, що послав це завдання.

Сервісна функція в архітектурі клієнт - сервер описується комплексом прикладних програм, відповідно до якого виконуються різноманітні прикладні процеси.

Процес, який викликає сервісну функцію за допомогою певних операцій, називається клієнтом. Їм може бути програма або користувач. На мал. 6.4 приведений перелік сервісів, які можуть бути присутніми в архітектурі клієнт - сервер. Клієнтами можуть виступати робочі станції, які використовують ресурси сервера і надають зручні інтерфейси користувача. Інтерфейси користувача це процедури взаємодії користувача з системою або мережею.

Клієнт є ініціатором і використовує електронну пошту або інші сервіси сервера. У цьому процесі клієнт запрошує вид обслуговування, встановлює сеанс, отримує потрібні йому результати і повідомляє про закінчення роботи.



Мал. 6.4 Модель клієнт-сервер

У мережах з виділеним файловим сервером на виділеному комп'ютері встановлюється серверна мережева операційна система. Цей ПК стає сервером. Програмне забезпечення (ПО), встановлене на робочій станції, дозволяє їй обмінюватися даними з сервером. Приклад мережових операційних систем : операційні системи сімейства Windows Server.

Окрім мережевої операційної системи потрібні мережеві прикладні програми, що реалізують переваги, що надаються мережею.

Мережі на базі серверів мають кращі характеристики і підвищену надійність. Сервер володіє головними ресурсами мережі, до яких звертаються інші робочі станції.

У сучасній клієнт - серверній архітектурі виділяється чотири групи об'єктів : клієнти, сервери, ці і мережеві служби. Клієнти розташовуються в системах на робочих місцях користувачів. Дані в основному зберігаються в серверах. Мережеві служби є спільно використовуваними серверами і даними. Крім того служби управляють процедурами обробки даних.

У міру ускладнення функцій, що покладаються на сервери, і збільшення числа обслуговуваних ними клієнтів відбувається все більша спеціалізація серверів. Існує безліч типів серверів.

- Первинний контролер домена, сервер, на якому зберігається база бюджетів користувачів і підтримується політика захисту.

- Вторинний контролер домена, сервер, на якому зберігається резервна копія бази бюджетів користувачів і політики захисту.

- Універсальний сервер, призначений для виконання нескладного набору різних завдань обробки даних в локальній мережі.

- Сервер бази даних, що виконує обробку запитів, що направляються базі даних.

- Проху сервер, що підключає локальну мережу до мережі Internet.

- Web -сервер, призначений для роботи з web -інформацією.

- Файловий сервер, що забезпечує функціонування розподілених ресурсів, включаючи файли, програмне забезпечення.

- Сервер застосувань, призначений для виконання прикладних процесів. З одного боку, взаємодіє з клієнтами, отримуючи завдання, а з іншого боку, працює з базами даних, підбираючи дані, необхідні для обробки.

- Сервер видаленого доступу, що забезпечує співробітникам, торговельним агентам, що працюють удома, службовцям філій, особам, що знаходяться у відрядженнях, можливість роботи з даними мережі.

- Телефонний сервер, призначений для організації в локальній мережі служби телефонії. Цей сервер виконує функції мовної пошти, автоматичного розподілу викликів, облік вартості телефонних розмов, інтерфейсу із зовнішньою телефонною мережею. Разом з телефонією сервер може також передавати зображення і повідомлення факсимільного зв'язку.

Поштовий сервер, що надає сервіс у відповідь на запити, прислані по електронній пошті.

Сервер доступу, що дає можливість колективного використання ресурсів, користувачами, що опинилися поза своїми ятерами (наприклад, користувачами, які знаходяться у відрядженнях і хочуть працювати зі своїми ятерами). Для цього користувачі через комунікаційні мережі з'єднуються з сервером доступу і останній надає потрібні ресурси, наявні в мережі.

- Термінальний сервер, що об'єднує групу терміналів, спрощує перемикання при їх переміщенні.

- Комунікаційний сервер, що виконує функції термінального сервера, але що здійснює також маршрутизацію даних.

- Відеосервер, який найбільшою мірою пристосований до обробки зображень, забезпечує користувачів відеоматеріалами, повчальними програмами, відеоіграми, забезпечує електронний маркетинг. Має високу продуктивність і велику пам'ять.

- Факс-сервер, що забезпечує передачу і прийом повідомлень в стандартах факсимільного зв'язку.

- Сервер захисту даних, оснащений широким набором засобів забезпечення безпеки даних і, в першу чергу, ідентифікації паролів.

- Мережі клієнт - серверної архітектури мають наступні переваги:

- дозволяють організовувати мережі з великою кількістю робочих станцій;

- забезпечують централізоване управління обліковими записами користувачів, безпекою і доступом, що спрощує мережеве адміністрування;

- ефективний доступ до мережевих ресурсів;
- користувачеві потрібний один пароль для входу в мережу і для дістання доступу до усіх ресурсів, на які поширюються права користувача.
- Разом з перевагами мережі клієнт - серверної архітектури мають і ряд недоліків :
- несправність сервера може зробити мережу непридатною, як мінімум втрату мережевих ресурсів;
- вимагають кваліфікованого персоналу для адміністрування;
- мають вищу вартість мереж і мережевого устаткування.
- Вибір архітектури мережі залежить від призначення мережі, кількості робочих станцій і від виконуваних на ній дій.
- Слід вибрати однорангову мережу, якщо:
 - кількість користувачів не перевищує десяти;
 - усі машини знаходяться близько один від одного;
 - мають місце невеликі фінансові можливості;
 - немає необхідності в спеціалізованому сервері, такому як сервер БД, факс-сервер або який-небудь інший;
 - немає можливості або необхідності в централізованому адмініструванні.
- Слід вибрати клієнт серверну мережу, якщо:
 - кількість користувачів перевищує десяти;
 - вимагається централізоване управління, безпека, управління ресурсами або резервне копіювання;
 - потрібний спеціалізований сервер;
 - потрібний доступ до глобальної мережі;
 - вимагається розділяти ресурси на рівні користувачів.

6.2.3. Класифікація за типом мережевої топології.

Під топологією мережі розуміється опис її фізичного розташування, тобто те, як комп'ютери сполучені в мережі один з одним і за допомогою яких пристроїв входять у фізичну топологію.

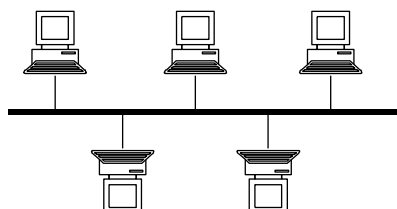
Існує чотири основні топології:

- Bus (шина);
- Ring (кільце);
- Star (зірка);
- Mesh (осередок).

Шина.

Фізична топологія шина, що іменується також лінійною шиною, складається з єдиного кабелю, до якого приєднані усі комп'ютери сегменту (мал. 6.5).

Повідомлення посилаються по лінії усім підключеним станціям незалежно від того, хто є одержувачем. Кожен комп'ютер перевіряє кожен пакет в дроті, щоб визначити одержувача пакету. Якщо пакет призначений для іншої станції, то комп'ютер відкидає його. Якщо пакет призначений цьому комп'ютеру, то він отримає і обробить його.



Малюнок 6.5 - Топологія "шина"

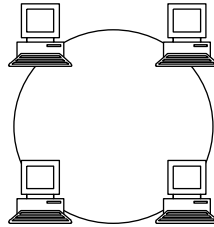
Головний кабель шини, відомий як магістраль, має на обох кінцях заглушки (термінатори) для запобігання віддзеркаленню сигналу.

Недоліки:

- важко ізолювати неполадки станції або іншого мережевого компонента;
- неполадки в магістральному кабелі можуть привести до виходу з ладу усієї мережі.

Кільце.

У фізичній топології "кільце" лінії передачі даних фактично утворюють логічне кільце, до якого підключені усі комп'ютери мережі (мал. 6.6).



Малюнок 6.6 - Топологія "кільце"

Доступ до носія в кільці здійснюється за допомогою маркерів (token), які пускаються по колу від станції до станції, даючи їм можливість переслати пакет, якщо це треба. Комп'ютер може посилати дані

тільки тоді, коли володіє маркером.

Оскільки кожен комп'ютер при цій топології є частиною кільця, він має можливість пересилати будь-які отримані ним пакети даних, адресовані іншій станції.

Недоліки:

- неполадки на одній станції можуть привести до відмови усієї мережі;
- при переконфігурації будь-якої частини мережі необхідно тимчасово відключати усю мережу.

Зірка.

У топології Star (зірка) усі комп'ютери в мережі сполучені один з одним за допомогою центрального концентратора (мал. 6.7).

Усі дані, які посилає станція, прямують прямо на концентратор, який пересилає пакет у напрямі одержувача.

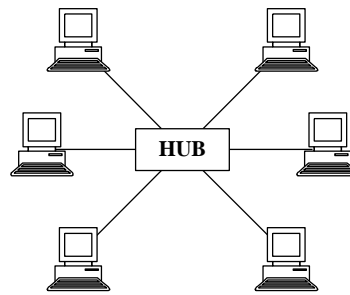
У цій топології тільки один комп'ютер може посилати дані в конкретний момент часу. При одночасній спробі двох і більше комп'ютерів переслати дані, усі вони дістануть відмову і будуть вимушені чекати випадковий

інтервал часу, щоб спробувати ще раз.

Ці мережі краще масштабуються, чим інші мережі. Неполадки на одній станції не виводять з ладу усю мережу. Наявність центрального концентратора полегшує додавання нового комп'ютера.

Недоліки:

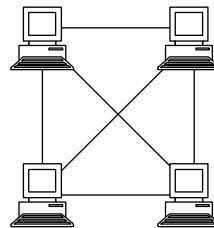
- вимагає більше кабелю, чим інші топології;
- вихід з ладу концентратора виведе з ладу увесь сегмент мережі.



Малюнок 6.7 - Топологія "зівка"

Комірка.

Топологія Mesh (комірка) сполучає усі комп'ютери попарно (мал. 6.8).



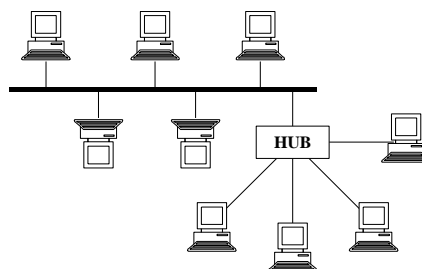
Малюнок 6.8 - Топологія "комірка"

Мережі Mesh використовують значно більшу кількість кабелю, чим інші топології. Ці мережі значно важче встановлювати. Але ці мережі стійкі до збоїв (здатні працювати за наявності ушкоджень).

Змішані топології.

На практиці існує безліч комбінацій головних мережевих топологій. Розглянемо основні з них.

Змішана топологія Star Bus (зівка на шині) об'єднує топології Шина і Зівка (мал. 6.9).



Малюнок 6.9 - Топологія "зівка на шині"

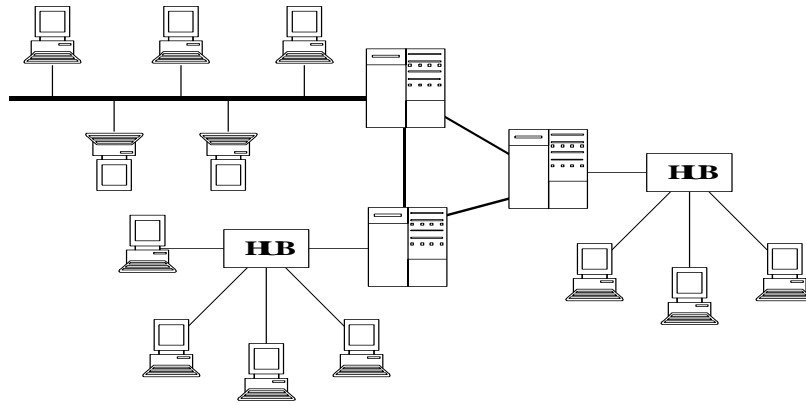
Топологія Star Ring (зівка на кільці) відома також під назвою Star - wired Ring, оскільки сам концентратор виконаний як кільце.

Ця мережа ідентична топології "зівка", але насправді концентратор сполучений дротами як логічне кільце.

Також як і у фізичному кільці, в цій мережі посилюються маркери для визначення порядку передачі даних комп'ютерами.

Топологія Hybrid Mesh (гібридний осередок). Оскільки реалізація справжньої топології Mesh у великих мережах може бути дорогою, мережа топології Hybrid Mesh може надати деякі з істотних переваг справжньої мережі Mesh.

В основному застосовується для з'єднання серверів, що зберігають критично важливі дані (мал. 6.10).



Малюнок 6.10 - Топологія "гібридна комірка"

Питання:

1. Чим відрізняється комунікаційна мережа від інформаційної мережі?
2. Як розділяються мережі за територіальною ознакою?
3. Що таке інформаційна система?
4. Що таке канали зв'язку?
5. Дати визначення фізичного каналу зв'язку.
6. Дати визначення логічного каналу зв'язку.
7. Як називається сукупність правил обміну інформацією між двома або декількома пристроями?
8. Яким параметром характеризується завантаження мережі?
9. Що таке метод доступу?
10. Що таке сукупність правил, що встановлюють процедури і формат обміну інформацією?
11. Які елементи входять до складу мережі?
12. Як називається опис фізичних з'єднань в мережі?
13. Що таке архітектура мережі?
14. Чим відрізняється однорангова архітектура від клієнт серверної архітектури?
15. У якому випадку використовується однорангова архітектура?
16. Які переваги великомасштабної мережі з виділеним сервером?
17. Що таке сервер? Що таке клієнт?
18. Які сервіси надає клієнт серверна архітектура?
19. Перерахуйте види мережових топологій.

6.3. Огляд мережевого устаткування.

6.3.1 Крайове устаткування лінії зв'язку.

Для підключення комп'ютера або терміналу до мережі потрібне так зване крайове устаткування лінії зв'язку (DCE - англ. Data Circuit - terminating Equipment або Data Communication Equipment або Data Carrier Equipment) - устаткування, що перетворює дані, комп'ютером або терміналом в сигнал для передачі по лінії зв'язку і здійснює зворотне перетворення. Основними видами такого обладнання є мережеві адаптери і модеми.

Мережевий адаптер (Network Interface Card, NIC) - це периферійний пристрій комп'ютера, що безпосередньо взаємодіє з середовищем передачі даних, яка прямо або через інше комунікаційне устаткування зв'язує його з іншими комп'ютерами. Цей пристрій вирішує завдання надійного обміну двійковими даними, представленими відповідними електромагнітними сигналами, по зовнішніх лініях зв'язку. Як і будь-який контролер комп'ютера, мережевий адаптер працює під управлінням драйвера операційної системи.

У більшості сучасних стандартів для локальних мереж передбачається, що між мережевими адаптерами взаємодіючих комп'ютерів встановлюється спеціальний комунікаційний пристрій (см далі), який бере на себе деякі функції по управлінню потоком даних.

Мережевий адаптер зазвичай виконує наступні функції:

- Оформлення передаваної інформації у вигляді кадру певного формату. Кадр включає декілька службових полів, серед яких є адреса комп'ютера призначення і контрольна сума кадру.

- Дістання доступу до середовища передачі даних. У локальних мережах в основному застосовуються канали зв'язку (загальна шина, кільце), що розділяються між групою комп'ютерів, доступ до яких надається по спеціальному алгоритму (найчастіше застосовуються метод випадкового доступу або метод з передачею маркера доступу по кільцю).

- Кодування послідовності біт кадру послідовністю електричних сигналів при передачі даних і декодування при їх прийомі. Кодування повинне забезпечити передачу початковою інформацією по лініях зв'язку з певною смугою пропускання і певним рівнем перешкод так, щоб приймаюча сторона змогла розпізнати з високою мірою вірогідності послану інформацію.

- Перетворення інформації з паралельної форми в послідовну і назад. Ця операція пов'язана з тим, що в обчислювальних мережах інформація передається в послідовній формі, біт за бітом, а не побайтно, як усередині комп'ютера.

- Синхронізація бітів, байтів і кадрів. Для стійкого прийому передаваної інформації потрібна підтримка постійного синхронізму приймача і передавача інформації.

Мережеві адаптери розрізняються за типом і розрядністю використовуваної в комп'ютері внутрішньої шини даних - ISA, EISA, PCI, MCA.

Мережеві адаптери розрізняються також за типом прийнятої в мережі мережевої технології і тому подібне. Як правило, конкретна модель мережевого адаптера працює за певною мережевою технологією.

У зв'язку з тим, що для кожної технології зараз є можливість використання різних середовищ передачі, мережевий адаптер може підтримувати як одну, так і одночасно декілька середовищ. У разі, коли мережевий адаптер підтримує тільки одне середовище передачі даних, а необхідно використовувати іншу, застосовуються трансивери і конвертори.

Трансивер (приймач, transmitter+receiver) - це частина мережевого адаптера, його крайовий пристрій, що виходить на кабель. У деяких варіантах виявилось зручним випускати мережеві адаптери, до яких можна приєднати трансивер для необхідного середовища.

Замість підбору відповідного трансивера можна використовувати конвертор, який може погоджувати вихід приймача, призначеного для одного середовища, з іншим середовищем передачі даних.

Модем (аббревіатура, складена із слів модулятор-демодулятор) - пристрій, що застосовується в системах зв'язку і виконує функцію модуляції і демодуляції. Модулятор здійснює модуляцію сигналу, що несе, тобто змінює його характеристики відповідно до змін вхідного інформаційного сигналу, демодулятор здійснює зворотний процес. Часткою випадком модему є широко вживаний периферійний пристрій для комп'ютера, що дозволяє йому зв'язуватися з іншим комп'ютером, обладнаним модемом, через телефонну мережу (телефонний модем) або кабельну мережу (кабельний модем).

Модеми діляться на види по виконанню:

зовнішні - підключаються через COM, USB порт або стандартний роз'єм в мережевій карті RJ - 45 зазвичай мають зовнішній блок живлення (існують USB -модеми, що живляться від USB і LPT -модеми).

внутрішні - встановлюються всередину комп'ютера в слот ISA, PCI, PCI - E, PCMCIA, AMR, CNR.

вбудовані - є внутрішньою частиною пристрою, наприклад ноутбука або док-станції.

За принципом роботи:

апаратні - усі операції перетворення сигналу, підтримка фізичних протоколів обміну, проводяться вбудованим в модем обчислювачем (наприклад з використанням DSP, контроллера). Так само в апаратному модемі присутній ПЗП, в якому записана мікропрограма, що управляє модемом,

винмодемы - апаратні модеми, позбавлені ПЗП з мікропрограмою. Мікропрограма такого модему зберігається в пам'яті комп'ютера, до якого підключений модем. Працездатний тільки за наявності драйверів, які зазвичай писалися виключно під операційні системи сімейства MS Windows,

напівпрограмні (Controller based soft - modem) - модеми, в яких частина функцій модему виконує комп'ютер, до якого підключений модем,

програмні (Host based soft - modem) - усі операції по кодуванню сигналу, перевірки на помилки і управління протоколами реалізовані програмно і проводяться центральним процесором комп'ютера. При цьому в модемі знаходиться аналогова схема і перетворювачі : АЦП, ЦАП, контроллер інтерфейсу (наприклад USB).

По виду з'єднання:

ISDN - модеми для цифрових комутованих телефонних ліній

DSL - використовуються для організації виділених (некомутованих) ліній використовуючи звичайну телефонну мережу. Відрізняються від комутованих модемів тим, що використовують інший частотний діапазон, а також тим, що по телефонних лініях сигнал передається тільки до АТС. Зазвичай дозволяють одночасно з обміном даними здійснювати використання телефонної лінії в звичайному порядку.

Кабельні - використовуються для обміну даними по спеціалізованих кабелях - приміром, через кабель колективного телебачення.

Стільникові - працюють по протоколах стільникового зв'язку - GPRS, EDGE, 3G, 4G і тому подібне. Часто мають виконання у вигляді USB -брелока. Як такі модеми також часто використовують термінали мобільного зв'язку (мобільні телефони).

PLC - використовують технологію передачі даних по дротах побутової електричної мережі.

6.3.2 Проміжне мережеве устаткування.

Повторителі і концентратори

Основна функція **повторителя (repeater)**, як це витікає з його назви, - повторення сигналів, що поступають на його порт. Повторитель покращує електричні характеристики сигналів і їх синхронність, і за рахунок цього з'являється можливість збільшувати загальну довжину кабелю між самими видаленими в мережі вузлами.

Багатоportовий повторитель часто називають концентратором (concentrator) або хабом (hub), що відбиває той факт, що цей пристрій реалізує не лише функцію повторення сигналів, але і концентрує в одному центральному пристрої функції об'єднання комп'ютерів в мережу. Практично в усіх сучасних мережевих стандартах концентратор є необхідним елементом мережі, що сполучає окремі комп'ютери в мережу.

Відрізки кабелю, сполучаючи два комп'ютери або які або два інших мережевих пристрої, називаються фізичними сегментами, тому концентратори і повторителі, які використовуються для додавання нових фізичних сегментів, є засобом фізичної структуризації мережі.

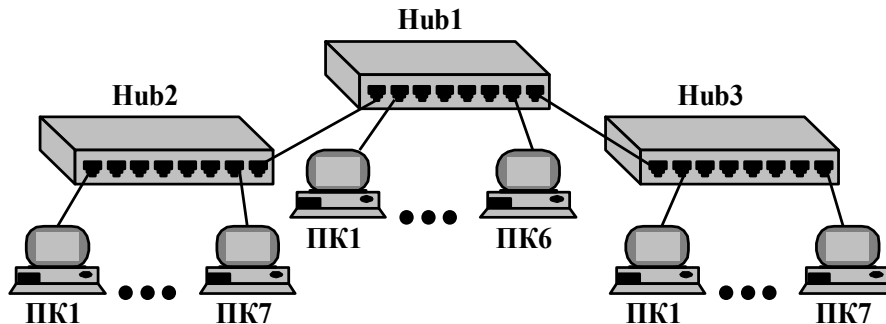
Концентратор - пристрій, у якого сумарна пропускна спроможність вхідних каналів вища за пропускну спроможність вихідного каналу. Оскільки потоки вхідних даних в концентраторі більше вихідного потоку, то головним його завданням є концентрація даних. При цьому можливі ситуації, коли число блоків даних, що поступає на входи концентратора, перевищує його можливості. Тоді концентратор ліквідує частину цих блоків.

Ядром концентратора є процесор. Функції, що виконуються концентратором, близькі до завдань, покладених на мультиплексор. Нарощувані (модульні) концентратори дозволяють вибирати їх компоненти, не думаючи про сумісність із вже використовуваними. Сучасні концентратори мають порти для підключення до різноманітних локальних мереж.

Концентратор є активним устаткуванням. Концентратор служить центром (шиною) зіркоподібної конфігурації мережі і забезпечує підключення мережевих пристроїв. У концентраторі для кожного вузла (ПК, принтери, сервери доступу, телефони і ін.) має бути передбачений окремий порт.

Нарощувані концентратори є окремими модулями, які об'єднуються за допомогою швидкодіючої системи зв'язку. Такі концентратори надають зручний спосіб поетапного розширення можливостей і потужності мережі.

Концентратор здійснює електричну розв'язку відрізків кабелю до кожного вузла, тому коротке замикання на одному з відрізків не виведе з ладу усю мережу.



Мал. 6.11 Логічний сегмент, побудований з використанням концентраторів

Концентратори утворюють з окремих фізичних відрізків кабелю загальне середовище передачі даних - логічний сегмент. Логічний сегмент також називають доменом колізій, оскільки при спробі одночасної передачі даних будь-яких двох комп'ютерів цього сегменту, що хоч би і належать різним фізичним сегментам, виникає блокування передавального середовища. Слід особливо підкреслити, що, яку б складну структуру не утворювали концентратори, наприклад шляхом ієрархічного з'єднання (мал. 6.11), усі комп'ютери, підключені до них, утворюють єдиний логічний сегмент, в якому будь-яка пара взаємодіючих комп'ютерів повністю блокує можливість обміну даними для інших комп'ютерів. Концентратори підтримують технологію "plug and play" і не вимагають якої-небудь установки параметрів. Необхідно просто спланувати свою мережу і вставити роз'єми в порти хаба і комп'ютерів.

Планування мережі з хабом

При виборі місця для установки концентратора беруть до уваги наступні аспекти: місце розташування;

відстані;
живлення.

Вибір місця установки концентратора є найбільш важливим етапом планування невеликої мережі. Хаб розумно розташувати поблизу геометричного центру мережі (на однаковій відстані від усіх комп'ютерів). Таке розташування дозволить мінімізувати витрату кабелю. Довжина кабелю від концентратора до будь-якого з комп'ютерів, що підключаються до мережі, або периферійних пристроїв не повинна перевищувати 100м

При плануванні мережі є можливість нарощування (каскадують) хабов.

Переваги концентратора

Концентратори мають багато переваг. По-перше, в мережі використовується топологія зірка, при якій з'єднання з комп'ютерами утворюють промені, а хаб є центром зірки. Така топологія спрощує установку і управління мережі. Будь-які переміщення комп'ютерів або додавання в мережу нових вузлів при такій топології дуже нескладно виконати. Крім того, ця топологія значно надійніша, оскільки при будь-якому ушкодженні кабельної системи мережа зберігає працездатність (перестає працювати лише пошкоджений промінь). Світлодіодні індикатори хаба дозволяють контролювати стан мережі і легко виявляти неполадки.

Різні виробники концентраторів реалізують у своїх пристроях різні набори допоміжних функцій, але найчастіше зустрічаються наступні:

- об'єднання сегментів з різними фізичними середовищами (наприклад, з різними видами кабелів);
- автосегментація портів - автоматичне відключення порту при його некоректній поведінці (ушкодження кабелю, інтенсивна генерація пакетів помилкової довжини і тому подібне);
- підтримка між концентраторами резервних зв'язків, які використовуються при відмові основних;
- захист передаваних по мережі даних від несанкціонованого доступу (наприклад, шляхом спотворення поля даних в кадрах, повторюваних на портах, що не містять комп'ютера з адресою призначення);

Мости і комутатори.

Міст (bridge) - ретрансляційна система, що сполучає канали передачі даних, об'єднує різнотипні канали передачі даних в один загальний.

Міст (bridge), а також його швидкодіючий аналог - комутатор (switching hub), ділять загальне середовище передачі даних на логічні сегменти. Логічний сегмент утворюється шляхом об'єднання декількох фізичних сегментів (відрізків кабелю) за допомогою одного або декількох концентраторів. Кожен логічний сегмент підключається до окремого порту моста/комутатора. При вступі кадру на який-небудь з портів міст/комутатор повторює цей кадр, але не на усіх портах, як це робить концентратор, а тільки на тому порту, до якого підключений сегмент, комп'ютер-адресат, що містить.

Мости можуть сполучати сегменти, що використовують різні типи носіїв і сполучати мережі з різними методами доступу до каналу.

Відмінність між мостом і комутатором

Різниця між мостом і комутатором полягає в тому, що міст в кожен момент часу може здійснювати передачу кадрів тільки між однією парою портів, а комутатор одночасно підтримує потоки даних між усіма своїми портами. Іншими словами, міст передає кадри послідовно, а комутатор паралельно.

Мости використовуються тільки для зв'язку локальних мереж з глобальними, тобто як засоби видаленого доступу, оскільки в цьому випадку необхідність в паралельній передачі між декількома парами портів просто не виникає.

Коли з'явилися перші пристрої, що дозволяють роз'єднувати мережу на декілька сегментів вони були двопортовими і дістали назву мостів (bridge). У міру розвитку цього типу устаткування, вони стали багатопортовими і дістали назву комутаторів (switch). Деякий час обидва поняття існували одночасно, а пізніше замість терміну "міст" стали застосовувати "комутатор". Далі в цій темі використовуватиметься термін "комутатор" для позначення цих обох різновидів пристроїв, оскільки усе сказане нижче в рівній мірі відноситься і до мостів, і до комутаторів. Слід зазначити, що останнім часом локальні мости повністю витиснені комутаторами.

Часто мости наділяються додатковими функціями. Такі мости мають певний інтелект (інтелектом в мережах називають дії, що виконуються пристроями) і фільтрують крізь себе блоки даних, адресовані абонентським системам, розташованим в тій же мережі. Для цього в пам'яті кожного моста є адреси систем, включених в кожну з мереж. Блоки, що проходять через інтелектуальний міст, двічі перевіряються, на вході і виході. Це дозволяє запобігати появі помилок усередині моста.

Часто мости наділяються додатковими функціями. Такі мости мають певний інтелект (інтелектом в мережах називають дії, що виконуються пристроями) і фільтрують крізь себе блоки даних, адресовані абонентським системам, розташованим в тій же мережі. Для цього в пам'яті кожного моста є адреси систем, включених в кожну з мереж. Блоки, що проходять через інтелектуальний міст, двічі перевіряються, на вході і виході. Це дозволяє запобігати появі помилок усередині моста.

Мости не мають механізмів управління потоками блоків даних. Тому може виявитися, що вхідний потік блоків виявиться більшим, ніж вихідний. В цьому випадку міст не впорається з обробкою вхідного потоку, і його буфери можуть переповнюватися. Щоб цього не сталося,

надлишкові блоки викидаються. Специфічні функції виконує міст в радіомережі. Тут він забезпечує взаємодію двох радіоканалів, що працюють на різних частотах. Його іменують **ретранслятором**.

Мости (bridges) оперують даними на високому рівні і мають абсолютно певне призначення. По-перше, вони призначені для з'єднання мережевих сегментів, що мають різні фізичні середовища, наприклад для з'єднання сегменту з оптоволоконним кабелем і сегменту з коаксіальним кабелем. Мости також можуть бути використані для зв'язку сегментів, що мають різні мережеві протоколи.

6.3.3 Комутатор

Комутатор (switch) - пристрій, що здійснює вибір одного з можливих варіантів напряму передачі даних. У комунікаційній мережі комутатор є ретрансляційною системою (система, призначена для передачі даних або перетворення протоколів), прозорості (тобто комутація здійснюється тут без якої-небудь обробки даних), що має властивість. Комутатор не має буферів і не може накопичувати дані. Тому при використанні комутатора швидкості передачі сигналів в каналах передачі даних, що сполучаються, мають бути однаковими. На відміну від інших видів ретрансляційних систем, тут, як правило, не використовується програмне забезпечення.

Спочатку комутатори використовувалися лише в територіальних мережах. Потім вони з'явилися і в локальних мережах, наприклад, приватні установські комутатори. Пізніше з'явилися комутовані локальні мережі. Їх ядром стали комутатори локальних мереж.

Комутатор (Switch) може сполучати сервери і служити основою для об'єднання декількох робітників груп. Він направляє пакети даних між вузлами мережі. Кожен комутований сегмент дістає доступ до каналу передачі даних без конкуренції і бачить тільки той трафік, який прямує в його сегмент. Комутатор повинен надавати кожному порту можливість з'єднання з максимальною швидкістю без конкуренції з боку інших портів (на відміну від спільно використовуваного концентратора). Зазвичай в комутаторах є один або два високошвидкісні порти, а також хороші інструментальні засоби управління. Комутатором можна замінити маршрутизатор, доповнити їм нарощуваний маршрутизатор або використовувати комутатор як основа для з'єднання декількох концентраторів. Комутатор може служити відмінним пристроєм для напряму трафіку між концентраторами мережі робочої групи і завантаженими файл-серверами.

Комутатор локальної мережі

Комутатор локальної мережі (local - area network switch) - пристрій, що забезпечує взаємодію сегментів однієї або групи локальних мереж.

Комутатор локальної мережі, як і звичайний комутатор, забезпечує взаємодію підключених до нього локальних мереж. Але на додаток до цього він здійснює перетворення інтерфейсів, якщо з'єднуються різні типи сегментів локальної мережі. У перелік функцій, що виконуються комутатором локальної мережі, входять забезпечення крізної комутації, наявність засобів маршрутизації, підтримка простого протоколу управління мережею, імітація моста або маршрутизатора, організація віртуальних мереж, швидкісна ретрансляція блоків даних.

6.3.4 Маршрутизатор

Маршрутизатор (router) - ретрансляційна система, що сполучає дві комунікаційні мережі або їх частини. З'єднання пар комунікаційних мереж здійснюється через маршрутизатори, які здійснюють необхідне перетворення вказаних протоколів. Маршрутизатор працює з декількома каналами, направляючи в який-небудь з них черговий блок даних. Маршрутизатори обмінюються інформацією про зміни структури мереж, трафік і їх стан. Завдяки цьому, вибирається оптимальний маршрут дотримання блоку даних в різних мережах від абонентської системи-відправника до системи-одержувача. Маршрутизатори забезпечують також з'єднання адміністративно незалежних комунікаційних мереж. Маршрутизатором може

бути як спеціальний електронний пристрій, так і спеціалізований комп'ютер, підключений до декількох мережевих сегментів за допомогою декількох мережевих карт.

Відмінність між маршрутизаторами і мостами

Маршрутизатори перевершують мости своєю здатністю фільтрувати і направляти пакети даних на мережі. Маршрутизатори можуть сполучати мережі, що використовують різну мережеву архітектуру, методи доступу до каналів зв'язку і протоколи. Маршрутизатори не мають такої здатності до аналізу повідомлень як мости, та зате можуть приймати рішення про вибір оптимального шляху для даних між двома мережевими сегментами. Мости приймають рішення з приводу адресації кожного з пакетів даних, що поступили, переправляти його через міст або немає залежно від адреси призначення. Маршрутизатори ж вибирають з таблиці маршрутів найкращий для цього пакету. У полі зору маршрутизаторів знаходяться тільки пакети, адресовані до них попередніми маршрутизаторами, тоді як мости повинні обробляти усі пакети повідомлень в сегменті мережі, до якого вони підключені. Тип топології або протоколу рівня доступу до мережі не має значення для маршрутизаторів. Маршрутизатори часто використовуються для зв'язку між сегментами з однаковими протоколами. Необхідно запам'ятати, що для роботи маршрутизаторів вимагається один і той же протокол в усіх сегментах, з якими він пов'язаний. При зв'язуванні мереж з різними протоколами краще використовувати мости. Для управління завантаженістю трафіку сегменту мережі також можна використовувати мости.

6.3.5 Шлюзи

Шлюз (gateway) - ретрансляційна система, що забезпечує взаємодію інформаційних мереж. Шлюз є найбільш складною ретрансляційною системою, що забезпечує взаємодію мереж з різними наборами протоколів. У свою чергу, набори протоколів можуть спиратися на різні типи фізичних засобів з'єднання. У тих випадках, коли з'єднуються інформаційні мережі, то в них частина рівнів може мати одні і ті ж протоколи. Тоді мережі з'єднуються не за допомогою шлюзу, а на основі простіших ретрансляційних систем, наприклад маршрутизаторами і мостами. Необхідність в мережевих шлюзах виникає при об'єднанні двох систем, що мають різну архітектуру. Як шлюз зазвичай використовується виділений комп'ютер, на якому запущено програмне забезпечення шлюзу і проводяться перетворення, що дозволяють взаємодіяти декільком системам в мережі. Іншою функцією шлюзів є перетворення протоколів. Шлюзи складні в установці і налаштуванні. Шлюзи працюють повільніше, ніж маршрутизатори.

Кабелі зв'язку, лінії зв'язку, канали зв'язку

Для організації зв'язку в мережах використовуються наступні поняття:

- кабелі зв'язку;
- лінії зв'язку;
- канали зв'язку.

Кабель зв'язку - це довгомірний виріб електротехнічної промисловості. З кабелів зв'язку і інших елементів (монтаж, кріплення, кожухи і так далі) будують лінії зв'язку. Прокладення лінії усередині будівлі завдання досить серйозне. Довжина ліній зв'язку коливається від десятків метрів до десятків тисяч кілометрів. У будь-яку більш-менш серйозну лінію зв'язку окрім кабелів входять: траншеї, колодязі, муфти, переходи через річки, море і океани, а також грозозахиста (так само як і інші види захисту) ліній. Дуже складні охорона, експлуатація, ремонт ліній зв'язку; зміст кабелів зв'язку під надлишковим тиском, профілактика (у сніг, дощ, на вітрі, в траншеї і в колодязі, в річці і на дні моря). Великою складністю є юридичні питання, що включають узгодження прокладення ліній зв'язку, особливо в місті. По вже побудованим лініям організовують канали зв'язку. Причому якщо лінію, як правило, будують і здають відразу усю, то канали зв'язку вводять поступово. Вже по лінії можна дати зв'язок, але таке використання украй дорогих споруд дуже неефективно. Тому застосовують апаратуру

каналообразовання (чи, як раніше говорили, ущільнення лінії). По кожному електричному ланцюгу, що складається з двох дротів, забезпечують зв'язок не одній парі абонентів (чи комп'ютерів), а сотням або тисячам : по одній коаксіальній парі в міжміському кабелі може бути утворені до 10800 каналів тональної частоти (0,3 - 3,4 КГц) або майже стільки ж цифрових, з пропускною спроможністю 64 Кбит/с.

За наявності кабелів зв'язку створюються лінії зв'язку, а вже по лініях зв'язку створюються канали зв'язку. Лінії зв'язку і канали зв'язку заводяться на вузли зв'язку. Лінії, канали і вузли утворюють первинні мережі зв'язку.

- Можливість легкого розширення мережі. Структурована кабельна система є модульною, тому її легко нарощувати, дозволяючи легко і ціною малих витрат переходити на досконаліше устаткування, що задовольняє зростаючим вимогам до систем комунікацій.

- Забезпечення ефективнішого обслуговування.

Структурована кабельна система полегшує обслуговування і пошук несправностей.

- Надійність. Структурована кабельна система має підвищену надійність, оскільки звичайне виробництво усіх її компонентів і технічний супровід здійснюється однією фірмою-виробником.

6.3.6 Кабельні системи

Виділяють два великі класи кабелів : електричні і оптичні, які принципово розрізняються за способом передачі по них сигналу.

Відмітна особливість оптоволоконних систем - висока вартість як самого кабелю (в порівнянні з мідним), так і спеціалізованих настановних елементів (розеток, роз'ємів, з'єднувачів і тому подібне). Правда, головний вклад до вартості мережі вносить ціна активного мережевого устаткування для оптоволоконних мереж.

Оптоволоконні мережі застосовуються для горизонтальних високошвидкісних каналів, а також все частіше стали застосовуватися для вертикальних каналів зв'язку (межетажних з'єднань).

Оптоволоконні кабелі в майбутньому зможуть скласти реальну конкуренцію мідним високочастотним, оскільки вартість виробництва мідних кабелів знижуватися не буде, адже для нього потрібна дуже чиста мідь, запасів якої на землі значно менше, чим кварцевого піску, з якого проводять оптоволокно.

Типи кабелів

Існує декілька різних типів кабелів, використовуваних в сучасних мережах. Нижче приведені найбільш часто використовувані типи кабелів. Безліч різновидів мідних кабелів складають клас електричних кабелів, використовуваних як для прокладення телефонних мереж, так і для інсталяції комп'ютерних мереж. По внутрішній будові розрізняють кабелі на витій парі і коаксіальні кабелі.

Кабель типу "вита пара" (twisted pair)

Витою парою називається кабель, в якому ізольована пара провідників скручена з невеликим числом витків на одиницю довжини. Скручування дротів зменшує електричні перешкоди ззовні при поширенні сигналів по кабелю, а екрановані виті пари ще більш збільшують міру завадозахищеності сигналів.

Кабель типу "вита пара" використовується в багатьох мережевих технологіях.

Кабелі на витій парі підрозділяються на: неекрановані (UTP - Unshielded Twisted Pair) і екрановані мідні кабелі. Останні підрозділяються на два різновиди: з екрануванням кожної пари і загальним екраном (STP - Shielded Twisted Pair) і з одним тільки загальним екраном (FTP - Foiled Twisted Pair). Наявність або відсутність екрану у кабелю зовсім не означає наявності або відсутності захисту передаваних даних, а говорить лише про різні підходи до пригнічення перешкод. Відсутність екрану робить неекрановані кабелі гнучкішими і стійкішими до зламів. Крім того, вони не вимагають дорогого контура заземлення для експлуатації в нормальному режимі, як екрановані. Неекрановані кабелі ідеально підходять для прокладення в приміщеннях усередині офісів, а екрановані краще використовувати для установки в місцях з особливими

умовами експлуатації, наприклад, поряд з дуже сильними джерелами електромагнітних випромінювань, яких в офісах зазвичай немає.

Кабелі класифікуються по категорії, вказаним в таблиці:

Категорія	Частота передаваного сигналу (МГц)
3	16
4	20
5	100
5+	300
6	200
7	600

Основою для віднесення кабелю до однієї з категорій служить максимальна частота передаваного по ньому сигналу.

Коаксіальні кабелі використовуються в радіо і телевізійній апаратурі. Коаксіальні кабелі можуть передавати дані із швидкістю 10 Мбіт/з на максимальну відстань від 185 до 500 метрів. Вони розділяються на товстих і тонких залежно від товщини. Типи коаксіальних кабелів приведені в таблиці:

Тип	Назва, значення опору
RG - 8 і RG - 11	Thicknet, 50 Ом
RG - 58/U	Thinnet, 50 Ом, суцільний центральний мідний провідник
RG - 58 A/U	Thinnet, 50 Ом, центральний багатожилий провідник
RG - 59	Broadband/Cable television (широкомовне і кабельне телебачення), 75 Ом
RG - 59 /U	Broadband/Cable television (широкомовне і кабельне телебачення), 50 Ом
RG - 62	ARCNet, 93 Ом

Кабель Thinnet, відомий як кабель RG, - 58, є найбільш широко використовуваним фізичним носієм даних. Мережі при цьому не вимагають додаткового устаткування і є простими і недорогими. Хоча тонкий коаксіальний кабель (Thin Ethernet) дозволяє передачу на меншу відстань, чим товстий, але для з'єднань з тонким кабелем застосовуються стандартні байонетні роз'єми BNC типу CP- 50 і зважаючи на його невелику вартість він стає фактично стандартним для офісних мереж.

Товстий коаксіальний кабель (Thick Ethernet) має велику міру заводо захищеності, велику механічну міцність, але вимагає спеціального пристосування для проколювання кабелю, щоб створити відгалуження для підключення до ЛВС. Він дорожчий і менш гнучкіший, чим тонкий.

Оптоволоконний кабель

Оптоволоконний кабель (Fiber Optic Cable) забезпечує високу швидкість передачі даних на великій відстані. Вони також несприйнятливі до інтерференції і підслуховування. У оптоволоконному кабелі для передачі сигналів використовується світло. Волокно, вживане як світлопровід, дозволяє передачу сигналів на великі відстані з величезною швидкістю, але воно дороге, і з ним важко працювати.

Для установки роз'ємів, створення відгалужень, пошуку несправностей в оптоволоконному кабелі потрібні спеціальні пристосування і висока кваліфікація. Оптоволоконний кабель складається з центральної скляної нитки завтовшки в декілька мікрон, покритою суцільною скляною оболонкою. Усе це, у свою чергу, заховано в зовнішню захисну оболонку.

Оптоволоконні лінії дуже чутливі до поганих з'єднань в роз'ємах. Як джерело світла в таких кабелях застосовуються світлодіоди (LED - Light Emitting Diode), а інформація кодується шляхом зміни інтенсивності світла. На приймальному кінці кабелю детектор перетворює світлові імпульси в електричні сигнали.

Існують два типи оптоволоконних кабелів - одномодові і багатомодові. Одномодові кабелі мають менший діаметр, велику вартість і дозволяють передачу інформації на великі відстані. Оскільки світлові імпульси можуть рухатися в одному напрямі, системи на базі оптоволоконних кабелів повинні мати кабель, що входить, і витікаючий кабель для кожного сегменту. Оптоволоконний кабель

вимагає спеціальних коннекторів і висококваліфікованої установки.

6.3.7 Безпроводні технології

Методи безпроводної технології передачі даних є зручним, а іноді незамінним засобом зв'язку. Безпроводні технології розрізняються по типах сигналу, частоті (велика частота означає велику швидкість передачі) і відстані передачі. Велике значення мають перешкоди і вартість. Можна виділити три основні типи безпроводної технології :

- радіозв'язок;
- зв'язок в мікрохвильовому діапазоні;
- інфрачервоний зв'язок.

Радіозв'язок

Технології радіозв'язку пересилають дані на радіочастотах і практично не мають обмежень по дальності. Вона використовується для з'єднання локальних мереж на великих географічних відстанях. Радіопередача в цілому має високу вартість і чутлива до електронного і атмосферного накладення, а також схильна до перехоплень, тому вимагає шифрування для забезпечення рівня безпеки.

Зв'язок в мікрохвильовому діапазоні

Передача даних в мікрохвильовому діапазоні (Microwaves) використовує високі частоти і застосовується як на коротких, так і на великих відстанях. Головне обмеження полягає в тому, щоб передавач і приймач були в зоні прямої видимості. Використовується в місцях, де використання фізичного носія ускладнене. Передача даних в мікрохвильовому діапазоні при використанні супутників може бути дуже дорогою.

Інфрачервоний зв'язок

Інфрачервоні технології (Infrared transmission), функціонують на дуже високих частотах, що наближаються до частот видимого світла. Вони можуть бути використані для встановлення двосторонньої або ширококомовної передачі на близьких відстанях. При інфрачервоному зв'язку зазвичай використовують світлодіоди (LED - Light Emitting Diode) для передачі інфрачервоних хвиль приймачу. Інфрачервона передача обмежена малою відстанню в прямій зоні видимості і може бути використана в офісних будівлях.

Питання:

1. Що таке мережевий адаптер? Які функції він виконує?
2. Що таке модем? Які функції він виконує?
3. Перерахуйте типи модемів.
4. Що таке повторитель?
5. Що таке концентратор?
6. Що таке комутатор?
7. Що таке маршрутизатор?
8. Що таке шлюз?
9. Що таке міст?
10. Чим відрізняється комутатор від моста?
11. Чим відрізняється маршрутизатор від моста?
12. Перерахуйте основні види кабелів?
13. Що таке структурована кабельна система?
14. Перерахуйте основні види безпроводних технологій?

6.4.Вимоги, що пред'являються до мереж

При організації і експлуатації мережі важливими вимогами при роботі являються наступні:

- продуктивність;
- надійність і безпека;
- розширюваність і масштабованість;
- прозорість;
- підтримка різних видів трафіку;
- керованість;
- сумісність.

Продуктивність

Продуктивність - це характеристика мережі, що дозволяє оцінити, наскільки швидко інформація передавальної робочої станції досягне до приймальної робочої станції.

На продуктивність мережі впливають наступні характеристики мережі :

- конфігурація;
- швидкість передачі даних;
- метод доступу до каналу;
- топологія мережі;
- технологія.

Якщо продуктивність мережі перестає відповідати вимогам, що пред'являються до неї, то адміністратор мережі може вдаватися до різних прийомів:

- змінити конфігурацію мережі так, щоб структура мережі більш відповідала структурі інформаційних потоків;
- перейти до іншої моделі побудови розподілених застосувань, яка дозволила б зменшити мережевий трафік;
- замінити мости швидкіснішими комутаторами.

Але найрадикальнішим рішенням в такій ситуації є перехід на швидкіснішу технологію.

Надійність і безпека

Найважливішою характеристикою обчислювальних мереж є надійність. Підвищення надійності засноване на принципі запобігання несправностям шляхом зниження інтенсивності відмов і збоїв за рахунок застосування електронних схем і компонентів з високою і надвисокою мірою інтеграції, зниження рівня перешкод, полегшених режимів роботи схем, забезпечення теплових режимів їх роботи, а також за рахунок вдосконалення методів зборки апаратури.

Відмовостійкість - це така властивість обчислювальної системи, яка забезпечує їй як логічній машині можливість продовження дій, заданих програмою, після виникнення несправностей. Введення відмовостійкості вимагає надлишкового апаратного і програмного забезпечення. Напрями, пов'язані із запобіганням несправностям і відмовостійкістю, основні в проблемі надійності. На паралельних обчислювальних системах досягається як найбільш висока продуктивність, так і, у багатьох випадках, дуже висока надійність. Наявні ресурси надмірності в паралельних системах можуть гнучко використовуватися як для підвищення продуктивності, так і для підвищення надійності.

Слід пам'ятати, що поняття надійності включає не лише апаратні засоби, але і програмне забезпечення. Головною метою підвищення надійності систем є цілісність що зберігаються в них даних.

Безпека - одне з основних завдань, що вирішуються будь-якою нормальною комп'ютерною мережею. Проблема безпеки можна розглядати з різних сторін - зловмисне псування даних, конфіденційність інформації, несанкціонований доступ, розкрадання і тому подібне

Забезпечити захист інформації в умовах локальної мережі завжди легше, ніж за наявності на фірмі десятка автономно працюючих комп'ютерів. Практично у вашому розпорядженні один інструмент - резервне копіювання (backup). Для простоти давайте називати цей процес резервуванням. Суть його полягає в створенні в безпечному місці повної копії даних, що оновлюється регулярно і як можна частіше. Для персонального комп'ютера більш менш безпечним носієм служать дискети. Можливе використання стримера, але це вже додаткові витрати на апаратуру.

Найлегше забезпечити захист даних від самих різних прикращів у разі мережі з виділеним файловим сервером. На сервері зосереджені усі найбільш важливі файли, а уберегти одну машину куди простіше, ніж десять. Концентрованість даних полегшує і резервування, оскільки не вимагається їх збирати по усій мережі.



Мал. 6.12 Завдання забезпечення безпеки даних

Екрановані лінії дозволяють підвищити безпеку і надійність мережі. Екрановані системи набагато стійкіші до зовнішніх радіочастотних полів.

Прозорість

Прозорість - це такий стан мережі, коли користувач, працюючи в мережі, не бачить її.

Комунікаційна мережа є прозорою відносно інформації, що проходить крізь неї, якщо вихідний потік бітів, в точності повторює вхідний потік. Але мережа може бути непрозорою в часі, якщо із-за розмірів черг блоків даних, що міняються, змінюється і час проходження різних блоків через вузли комутації. Прозорість мережі за швидкістю передачі даних вказує, що дані можна передавати з будь-якою потрібною швидкістю.

Якщо в мережі по одних і тих же маршрутах передаються інформаційні сигнали, що управляють (що синхронізують), то говорять, що мережа прозора по відношенню до типів сигналів.

Якщо передавана інформація може кодуватися будь-яким способом, то це означає, що мережа прозора для будь-яких методів кодувань.

Прозора мережа є простим рішенням, в якому для взаємодії локальних мереж, розташованих на значній відстані один від одного, використовується принцип Plug - and - play (підключися і працюй).

Служба прозорих локальних мереж забезпечує крізне (end - to - end) з'єднання, що зв'язує між собою видалені локальні мережі. Привабливість цього рішення полягає в тому, що ця служба об'єднує видалені один від одного на значну відстань вузли як частини локальної мережі. Тому не треба вкладати кошти у вивчення нових технологій і створення територіально розподілених мереж (Wide - Area Network - WAN). Користувачам вимагається тільки підтримувати локальне з'єднання, а провайдер служби прозорих мереж забезпечить

безперешкодну взаємодію вузлів через мережу масштабу міста (Metropolitan - Area Network - MAN) або мережу WAN. Служби Прозорої локальної мережі мають багато переваг. Наприклад, користувач може швидко і безпечно передавати великі об'єми даних на значні відстані, не обтяжуючи себе складнощами, пов'язаними з роботою в мережах WAN.

Підтримка різних видів трафіку

Трафік в мережі складається випадковим чином, проте в ній відбиті і деякі закономірності. Як правило, деякі користувачі, що працюють над загальним завданням, (наприклад, співробітники одного відділу), найчастіше поводяться із запитами або один до одного, або до загального сервера, і тільки іноді вони випробовують необхідність доступу до ресурсів комп'ютерів іншого відділу. Бажано, щоб структура мережі відповідала структурі інформаційних потоків. Залежно від мережевого трафіку комп'ютери в мережі можуть бути розділені на групи (сегменти

мережі). Комп'ютери об'єднуються в групу, якщо велика частина породжуваних ними повідомлень, адресована комп'ютерам цієї ж групи.

Для розподілу мережі на сегменти використовуються мости і комутатори. Вони екранують локальний трафік усередині сегменту, не передаючи за його межі ніяких кадрів, окрім тих, які адресовані комп'ютерам, що знаходяться в інших сегментах. Таким чином, мережа розпадається на окремі підмережі. Це дозволяє раціональніше вибирати пропускну спроможність наявних ліній зв'язку, враховуючи інтенсивність трафіку усередині кожної групи, а також активність обміну даними між групами.

Проте локалізація трафіку засобами мостів і комутаторів має істотні обмеження. З іншого боку, використання механізму віртуальних сегментів, реалізованого в комутаторах локальних мереж, призводить до повної локалізації трафіку; такі сегменти повністю ізольовані один від одного, навіть відносно широкомовних кадрів. Тому в мережах, побудованих тільки на мостах і комутаторах, комп'ютери, що належать різним віртуальним сегментам, не утворюють єдиної мережі.

Для того, щоб ефективно консолідувати різні види трафіку в мережі АТМ, вимагається спеціальна попередня підготовка (адаптація) даних, таких, що мають різний характер: кадри - для цифрових даних, сигнали імпульсно-кодової модуляції - для голосу, потоки бітів - для відео. Ефективна консолідація трафіку вимагає також обліку і використання статистичних варіацій інтенсивності різних типів трафіку.

Керованість

ISO (International Standardization Organization - Міжнародна організація по стандартизації) внесла великий вклад до стандартизації мереж. Модель управління мережі є основним засобом для розуміння головних функцій систем управління мережі. Ця модель складається з 5 концептуальних областей:

- управління ефективністю;
- управління конфігурацією;
- управління обліком використання ресурсів;
- управління несправностями;
- управління захистом даних.

Управління ефективністю

Мета управління ефективністю - вимір і забезпечення різних аспектів ефективності мережі для того, щоб міжмережева ефективність могла підтримуватися на прийнятному рівні. Прикладами змінних ефективності, які могли б бути забезпечені, є пропускну спроможність мережі, час реакції користувачів і коефіцієнт використання лінії.

Управління ефективністю включає декілька етапів:

- збір інформації про ефективність по тих змінних, які представляють інтерес для адміністраторів мережі;
- визначення відповідних порогів ефективності для кожної важливої змінної таким чином, що перевищення цих порогів вказує на наявність проблеми в мережі, гідній уваги.

Управління конфігурацією

Мета управління конфігурацією - контроль інформації про мережеву і системну конфігурацію для того, щоб можна було відстежувати і управляти дією на роботу мережі різних версій апаратних і програмних елементів. Оскільки усі апаратні і програмні елементи мають експлуатаційні відхилення, погрішності (чи те і інше разом), які можуть впливати на роботу мережі, така інформація важлива для підтримки гладкої роботи мережі.

Кожен пристрій мережі має в розпорядженні різноманітну інформацію про версії, що асоціюються з ним. Щоб забезпечити легкий доступ, підсистеми управління конфігурацією зберігають цю інформацію в базі даних. Коли виникає яка-небудь проблема, в цій базі даних може бути проведений пошук ключів, які могли б допомогти розв'язати цю проблему.

Управління обліком використання ресурсів

Мета управління обліком використання ресурсів - вимір параметрів використання мережі, щоб можна було відповідним чином регулювати її використання індивідуальними або груповими користувачами. Таке регулювання мінімізує число проблем в мережі (оскільки ресурси мережі можуть бути поділені виходячи з можливостей джерела) і максимізував рівнодоступність до мережі для усіх користувачів.

Управління несправностями

Мета управління несправностями - виявити, зафіксувати, повідомити користувачів і (в межах можливого) автоматично усунути проблеми в мережі, з тим щоб ефективно підтримувати роботу мережі. Оскільки несправності можуть привести до простоїв або неприпустимої деградації мережі, управління несправностями, ймовірно, є найбільш широко використовуваним елементом моделі управління мережі ISO.

Управління несправностями включає декілька кроків:

- визначення симптомів проблеми;
- ізолювання проблеми;
- усунення проблеми;
- перевірка усунення несправності на усіх важливих підсистемах;
- реєстрація виявлення проблеми і її рішення.

Управління захистом даних

Мета управління захистом даних - контроль доступу до мережевих ресурсів відповідно до місцевих керівних принципів, щоб зробити неможливими саботаж мережі і доступ до чутливої інформації особам, що не мають відповідного дозволу. Наприклад, одна з підсистем управління захистом даних може контролювати реєстрацію користувачів ресурсу мережі, відмовляючи в доступі тим, хто вводить коди доступу, не відповідні встановленим.

Підсистеми управління захистом даних працюють шляхом розподілу джерел на санкціоновані і несанкціоновані області. Для деяких користувачів доступ до будь-якого джерела мережі є невідповідним.

Підсистеми управління захистом даних виконують наступні функції:

- ідентифікують чутливі ресурси мережі (включаючи системи, файли і інші об'єкти);
- визначають відображення у вигляді карт між чутливими джерелами мережі і набором користувачів;
- контролюють точки доступу до чутливих ресурсів мережі;
 - реєструють невідповідний доступ до чутливих ресурсів мережі.

Сумісність

Сумісність і мобільність програмного забезпечення. Концепція програмної сумісності уперше в широких масштабах була застосована розробниками системи IBM/360. Основне завдання при проектуванні усього ряду моделей цієї системи полягало в створенні такої архітектури, яка була б однаковою з точки зору користувача для усіх моделей системи незалежно від ціни і продуктивності кожної з них. Величезні переваги такого підходу, що дозволяє зберігати існуючий заділ програмного забезпечення при переході на нові (як правило, продуктивніші) моделі, були швидко оцінені як виробниками комп'ютерів, так і користувачами, і починаючи з цього часу практично усі фірми-постачальники комп'ютерного устаткування узяли на озброєння ці принципи, поставляючи серії сумісних комп'ютерів. Слід помітити проте,

що з часом навіть сама передова архітектура неминуче застаріває і виникає потреба внесення радикальних змін в архітектуру і способи організації обчислювальних систем.

Нині одним з найбільш важливих чинників, що визначають сучасні тенденції в розвитку інформаційних технологій, є орієнтація компаній-постачальників комп'ютерного устаткування на ринок прикладних програмних засобів.

Цей перехід висунув ряд нових вимог. Передусім, таке обчислювальне середовище повинне дозволяти гнучко міняти кількість і склад апаратних засобів і програмного забезпечення відповідно до вимог вирішуваних завдань, що міняються. По-друге, вона повинна забезпечувати можливість запуску одних і тих же програмних систем на різних апаратних платформах, тобто забезпечувати мобільність програмного забезпечення. По-третє, це середовище повинне гарантувати можливість застосування одних і тих же людино-машинних інтерфейсів на усіх комп'ютерах, що входять в неоднорідну мережу. В умовах жорсткої конкуренції виробників апаратних платформ і програмного забезпечення сформувалася концепція відкритих систем, сукупність стандартів, що є, на різні компоненти обчислювального середовища, призначених для забезпечення мобільності програмних засобів у рамках неоднорідної, розподіленої обчислювальної системи.

Питання

1. Які основні вимоги пред'являються до мереж?
2. Що таке продуктивність мережі?
3. Які характеристики впливають на продуктивність мережі?
4. Які є способи підвищення продуктивності мереж?
5. Як забезпечити високошвидкісну пересилку трафіку?
6. Чим забезпечується надійність мережі?
7. Що таке відмовостійкість?
8. Перерахувати завдання безпеки даних в мережі.
9. Для якої мети використовується резервне копіювання?
10. Чим забезпечується безпека мереж в клієнт-серверній архітектурі?
11. Для якої мети встановлюються екрановані лінії в мережі?
12. Що таке прозорість мереж?
13. У якому випадку лінія прозора по відношенню до типів сигналів?
14. Що таке прозоре з'єднання?
15. Що використовується для розподілу мережі на сегменти?
16. Яким чином можна зменшити трафік в мережі?
17. Дати визначення керованості мереж і перерахувати основні функції управління мережами.
18. Що включається в управління ефективністю?
19. Для якої мети використовується управління несправностями?
20. Для чого потрібне управління конфігурацією?
21. Яка мета управління захистом даних?
22. Які функції підсистеми управління захистом даних?
23. Дати визначення поняття сумісності мереж.

6.5. Глобальна мережа Internet (інтернет). WWW.

Історія виникнення Інтернету. Поняття Інтернету. Стек протоколів TCP/IP.

В середині 60-х років Міністерство оборони США побажало управляти мережею, яка могла б пережити навіть ядерну війну. Звичайні мережі, сполучені телефонними дротами, були визнані занадто уразливими, оскільки втрата однієї лінії або комутатора перервала б зв'язок, для якого ця лінія або комутатор використовувалися, а можливо, навіть розділила б мережу на окремі ділянки. У цих мережах використовувалася комутація каналів (коли на час передачі інформації пари комп'ютерів з'єднуються "один-с-одним" і в період з'єднання

відбувається передача усього об'єму інформації). Для вирішення проблеми Міністерство оборони звернулося до управління перспективного планування науково-дослідних робіт ARPA (Advanced Research Projects Agency). Приблизно в цей же час Теодором Хольме Нельсоном була придумана ідея гіпертексту (сам термін уперше був вимовлений в 1965 році).

У грудні 1969 року вдалося запустити експериментальну мережу (ARPANET), що складається з 4 вузлів. Ця мережа підтримувала комутацію пакетів (коли необхідні для передачі дані розбиваються на частини, а до кожної частини приєднується заголовок, що містить повну інформацію про доставку пакету за призначенням). В процесі експлуатації з'ясувалося, що наявні протоколи мережі не придатні для роботи з об'єднаними мережами. В результаті були проведені додаткові дослідження в області протоколів, що завершилися в середині 70-х винаходом моделі і протоколів TCP/IP. TCP/IP був спеціально розроблений для управління обміном даними по інтермережах, що ставало усе більш важливим у міру підключення усе нових мереж до ARPANET. У 1980 році були представлені основні принципи об'єднання мереж :

- мережі взаємодіють між собою по протоколу TCP/IP;
- об'єднання мереж проводиться через спеціальні шлюзи (gateway);
- усі комп'ютери, що підключаються, мають єдині методи адресації.

З 1983 року протокол TCP/IP став обов'язковим на усіх комп'ютерах ARPANET. Розвиток мережі відбувався досить швидкими темпами. До 1984 року число хост-систем (крайових систем) перевищило 1000. У цьому ж році вводиться принцип системи найменування доменів (Domain Name System, DNS), що дозволяє замінити цифрові адреси комп'ютерів символьними іменами.

Сучасна історія Інтернету почалася в 1989 році, коли Тім Бернерс-Лі (Tim Berners - Lee) в Європейській лабораторії фізики елементарних часток (CERN) розробив технологію гіпертекстових документів - World Wide Web (WWW), що дозволяє користувачам мати доступ до будь-якої інформації, що знаходиться в мережі Інтернет на комп'ютерах по всьому світу. Саме ця розробка зумовила кінець кінцем вибухове зростання числа

користувачів Інтернету. За минулий час WWW пройшов декілька фаз розвитку.

Символьний гіпертекст

Спочатку Web була текстовою гіперпосилальною системою. Перша програма-браузер (браузер - програма для перегляду html -документов) LineMode, створена в 1991 році, не підтримувала графіку і "мишу". Можна лише було переглядати статичні гіпертекстові сторінки, вводючи номери сторінок, що переглядаються.

Графічно-орієнтовані статичні HTML -документи

Ця фаза почалася в лютому 1993 року з випуском браузеру NSCA Mosaic. Браузер розроблявся для Національного центру по застосуванню супер-ЕОМ (NSCA). Це була невелика програма (9000 рядків коду), що працює тільки під X, - windows (графічна оболонка для операційних систем сімейства unix). Але в ній було головне - графіка стала повноправною частиною інтерфейсу, а миша - єдиним засобом роботи. Розробники Mosaic створили повністю новий інтерфейс для Інтернет. Інтерфейс виявився настільки приємний на погляд і в роботі, що підкорював кожного, хто починав з ним працювати просто з цікавості. Браузер не вимагав від користувачів знань про внутрішній устрій мережі. Інтернет відкрив простори мільйонам користувачів.

Через декілька місяців комп'ютерні компанії стали цілеспрямовано проникати в Web, яка сама по собі залишалася статичною. Вміст включав текстові або графічні документи і ряд інших елементів. Можливо, сторінка містила аудіо- або відеофайли, які доводилося спочатку завантажувати, а потім програвати, використовуючи зовнішні застосування.

Динамічні HTML -документи

Протягом перших двох фаз Web -сторінки створювалися за допомогою текстового редактора HTML і поміщалися на сервер. Вони залишалися незмінними до тих пір, поки автор не змінював їх. Для динамічної генерації стали використовувати CGI -сценарії (сценарії

інтерфейсу загального шлюзу), які дозволяли на сервері генерувати HTML -сторониці. З таким удосконаленням Web могла служити середовищем прикладного програмування, де уся обробка, проте проводилася на сервері.

Активні HTML -документи

Четверта фаза почалася в 1995 році з появою модулів, що підключалися, в Netscape Navigator і появою мови Java. Що спочатку розроблявся для потреб виробників побутової електроніки, мова Java виявилася засобом, що дозволяв перейти в Web на технологію клієнт-сервер, в якій клієнт мав певну незалежність від сервера. Інтернет-програмування перестало бути програмуванням тільки для сервера. А трохи пізніше була розроблена мова сценаріїв JavaScript, повністю інтегрований в гіпертекстову розмітку документів. У тому ж 1995 року Web стає найпопулярнішим сервісом Інтернету. Поступово інші сервіси стають в WWW вторинними, оскільки багато хто користується Web -сайтами як шлюзами для доступу до інших служб.

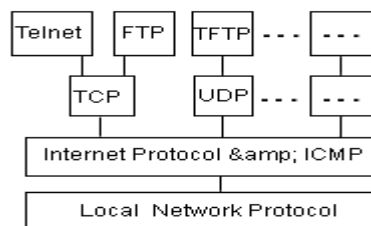
Для більшості користувачів глобальної мережі Інтернет асоціюється з одним високорівневим сервісом - World Wide Web (WWW), що використовує протокол HTTP (Hypertext Transfer Protocol - протокол передачі гіпертексту). Зрозуміло, існує безліч інших протоколів - електронна пошта, канали новин, FTP (File Transfer Protocol - протокол передачі файлів) та ін.

Що ж таке Інтернет? Визначення цьому терміну було дане в 1995 році Федеральною мережевою радою (FNC). Інтернет - це глобальна інформаційна система, яка, :

- логічно взаємозв'язана простором глобальних унікальних адрес, заснованих на Інтернет-протоколі (IP) або наступних розширеннях або наступниках IP;
- здатна підтримувати комунікації з використанням сімейства Протоколу управління передачею/Інтернет-протоколу (TCP/IP) або його наступних розширень/наступників і/або інших IP -совместимых протоколів;
- забезпечує, використовує або робить доступною, на громадській або приватній основі, високорівневі сервіси, надбудовані над описаною тут комунікаційною і іншою пов'язаною тут інфраструктурою.

Отже, основа Інтернету - протокол IP. **Еталонну модель стека протоколів Інтернету можна зображена на малюнку 6.13.**

Протокол Інтернету (IP) є основою усієї архітектури. Мета протоколу Internet полягає в передачі даних (одиниця інформації в протоколі, блок інформації, посланий як пакет мережевого рівня через передавальне середовище без попереднього встановлення з'єднання і створення віртуального каналу; повідомлення, яке не вимагає підтвердження про прийом від приймаючої сторони) через набір об'єднаних комп'ютерних мереж. Таким чином, на мережевому рівні Інтернет можна розглядати як набір підмереж або автономних систем, сполучених один з одним. Це здійснюється за допомогою передачі даних від одного модуля Internet до іншого до тих пір, поки не буде досягнутий одержувач. Дві основні функції протоколу IP - адресація і фрагментація.



верхний уровень (прикладной уровень)

TCP (транспортный уровень)

протокол Internet (межсетевой уровень)

коммуникационная сеть (от хоста сети)

Мал. 6.13 Еталонна модель стека протоколів Інтернету

Модулі Internet знаходяться на **хостах** (машинах, призначених для виконання програм користувача) і **шлюзах** (машинах, що забезпечують з'єднання різних, часто несумісних мереж) мереж системи Internet. Дейтаграмми прямують з одного модуля Internet на інший через конкретні комп'ютерні мережі, засновані на інтерпретації Internet -адресов. Таким чином, одним з важливих механізмів протоколу Internet є Internet -адрес. Принципи адресації в Інтернеті і структура IP -адреса буде розглянуті нижче.

При передачі повідомлень з одного Internet -модуля на іншій дейтаграмми можуть потребувати проходження через мережі, для яких максимальний розмір пакету менший, ніж розмір дейтаграмми. Щоб здолати цю складність, в протокол Internet включений механізм фрагментації.

Протокол IP обробляє кожну Internet -дейтаграмму як незалежну одиницю, що не має зв'язку ні з якими іншими дейтаграммами Internet. Протокол не має справи ні із з'єднаннями, ні з логічними ланцюжками (віртуальними або якими-небудь іншими). Протокол Internet не забезпечує надійності комунікації. Немає механізму підтверджень ні між відправником і одержувачем, ні між хост-комп'ютерами. Немає контролю помилок для поля даних, тільки контрольна сума для заголовка. Не підтримується повторна передача, немає управління потоком.

Протоколи транспортного рівня дозволяють відправляти і отримувати сегменти інформації змінної довжини, поміщені в оболонку Internet дейтаграмм. Основна функція транспортного рівня - прийняти інформацію від протоколу верхнього рівня, при необхідності розбити дані на фрагменти і гарантувати, що ці частини в правильному виді прибудуть за призначенням. На цьому рівні використовуються два протоколи.

Протокол TCP (Transmission Control Protocol - протокол управління передачею) є надійним протоколом на основі з'єднань, що дозволяє без помилок доставляти байтовий потік з однієї машини на будь-яку іншу машину об'єднаної мережі. Він розбиває вхідний потік байт на окремі повідомлення і передає їх міжмережевому рівню (протоколу IP). На пункті призначення одержуючий TCP -процес відновлює з отриманих повідомлень вихідний потік. Крім того, TCP здійснює управління потоком, щоб швидкий відправник не завалив інформацією повільного одержувача.

Щоб ідентифікувати окремі потоки даних, підтримувані протоколом TCP, останній визначає ідентифікатори портів (16-бітова адреса). Номери портів нижче 1024, звані **популярними портами**, зарезервовані за стандартними службами. Наприклад, служба передачі файлів використовує порт 21, передачі гіпертексту - 80 порт. Оскільки ідентифікатори портів вибираються кожною програмою протоколу TCP незалежно, то вони не будуть унікальні. Щоб забезпечити унікальність адрес для кожної програми протоколу TCP, ми об'єднуємо той, що ідентифікує цю програму Internet адреса і ідентифікатор порту. В результаті отримуємо сокет, який буде унікальний в усіх локальних мережах, об'єднаних в єдине ціле. З'єднання повністю визначається парою сокетів на своїх кінцях.

Протокол UDP (User Data Protocol - призначений для користувача протокол даних) є ненадійним протоколом без встановлення з'єднання. Цей протокол надає прикладній програмі процедуру для посилки повідомлень іншим програмам, причому механізм протоколу мінімальний. Протокол UDP орієнтований на транзакції, отримання дейтаграмм і захист від дублювання не гарантовані.

Завдання протоколів верхнього рівня - надання високорівневого сервісу. До них відносяться протоколи FTP (File Transfer Protocol - протокол передачі файлів), HTTP (Hypertext Transfer Protocol - протокол передачі гіпертексту), SMTP (Simple Mail Transfer Protocol - поштовий протокол) і інші.

Інтернет-протокол. Адресація в Інтернеті. Служба DNS.

Як було вказано вище, Інтернет-протокол IP взаємодіє, з одного боку, з протоколом більш високого рівня, від якого отримує дані для передачі по мережі або якому передає отримані дані, і з протоколами локальної комп'ютерної мережі, яким передаються упаковані дані для передачі або від яких приходять пакети даних.

Схему дій для передачі дейтаграмми від однієї прикладної програми до іншої можна проілюструвати таким чином (см. рис. 6.14).

Припустимо, що перенесення включатиме проходження одного проміжного шлюзу. Відправляюча прикладна програма готує свої дані і викликає свій локальний Internet -модуль для відправки цих даних як дейтаграмми, а як аргументи цього виклику передає адресу одержувача і інші параметри.

Модуль Internet готує заголовок дейтаграмми і стикає з ним дані. У заголовку дейтаграмми вказуються - версія протоколу IP, тип сервісу (висока достовірність, висока пропускна спроможність, мала затримка), час життя пакету, протокол верхнього рівня, що використовує передавані дані, адресу відправника, адреса одержувача та ін. параметри. Модуль Internet визначає локальну мережеву адресу, відповідну цій адресі Internet. В даному випадку це адреса шлюзу.

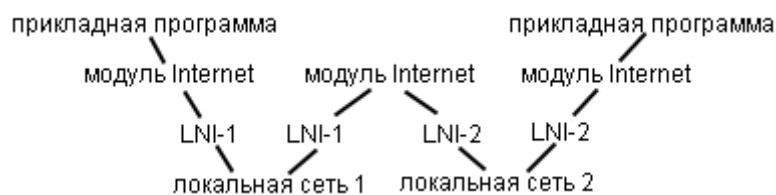
Модуль передає цю дейтаграмму і адресу в локальній мережі в розпорядження інтерфейсу локальної мережі.

Інтерфейс локальної мережі створює відповідний цій мережі заголовок і сполучає з ним дейтаграмму. Потім він передає по локальній мережі отриманий таким чином результат.

Дейтаграмма достигает хост-компьютер, играющий роль шлюза и расположенный в вершине сети. Интерфейс локальной сети отделяет этот заголовок и передает дейтаграмму на модуль Internet. Модуль Internet определяет из Internet адреса, что дейтаграмма должна быть направлена на хост-компьютер во второй сети. Модуль Internet определяет адрес хоста-получателя в локальной сети. Он обращается к интерфейсу локальной сети с тем, чтобы она переслала данную дейтаграмму по назначению.

Інтерфейс створює заголовок локальної мережі і сполучає з ним дейтаграмму, а потім результат направляє на хост-получатель. При необхідності (якщо максимальний розмір пакету в мережі менший, ніж розмір дейтаграмми) дейтаграмма може бути розбита на частини. Кожна частина забезпечується ідентифікатором, що дозволяє потім зібрати їх в єдине ціле. На хосте-получателе інтерфейс локальної мережі видаляє заголовок локальної мережі і передає те, що залишилося на Internet модуль.

Модуль Internet визначає, що дейтаграмма, що розглядається вище, призначена для прикладної програми на цей хосте. Модуль передає дані прикладній програмі у відповідь на системний виклик. Як результат цього виклику передаються адреса одержувача і інші параметри.



Мал. 6.14 Шлях передачі дейтаграмми

З вищесказаного ясно, що одним з найважливіших понять IP -протокола являється Інтернет-адреса (IP -адрес).

Кожен комп'ютер в мережі TCP/IP має IP -адрес, що складається з номера мережі і номера вузла. Усе IP -адреса мають довжину 32 біта і використовуються в полях **Адреса одержувача і Адреса відправника IP** -пакетов (дейтаграмм). IP -адрес зазвичай записується у вигляді чотирьох чисел (октетів), що представляють значення кожного байта в десятковій формі, і розділених точками, наприклад, 192.10.24.100

Номер мережі може бути вибраний адміністратором довільно, або призначений за рекомендацією спеціального підрозділу Internet (Network Information Center, NIC), якщо мережа повинна працювати як складова частина Internet. Зазвичай провайдери послуг Internet отримують діапазони адрес у підрозділів NIC, а потім розподіляють їх між своїми абонентами.

Вузол може входити в декілька IP -сетей. В цьому випадку вузол повинен мати декілька IP -адресов, по числу мережових зв'язків. Таким чином, IP -адрес характеризує не окремий комп'ютер або маршрутизатор, а одне мережеве з'єднання.

Ділення IP -адреса на полі номера мережі і номера вузла - гнучке, і межа між цими полями може встановлюватися дуже довільно. Яка частина адреси відноситься до номера мережі, а яка - до номера вузла, визначається значеннями перших бітів адреси.

	1 байт		2 байт	3 байт	4 байт
Класс А		№ сети	№ узла		
Класс В	10	№ сети		№ узла	
Класс С	110	№ сети			№ узла
Класс D	1110	Адрес группы широковещания (multicast)			
Класс E	11110	Зарезервировано для будущего использования			

Структура IP -адресів.

З таблиці видно:

- Мережа класу А має номери в діапазоні 1 - 126. Під адресу мережі відводиться 1 байт, під адресу вузла - 3 байти.

- Мережа класу В має номери в діапазоні 128 - 191. Під адресу мережі відводиться 2 байти, під адресу вузла - 2 байти.

- Мережа класу С має номери в діапазоні 192 - 223. Під адресу мережі відводиться 3 байти, під адресу вузла - 1 байт.

- Адреса класу D означає особливу, групову адресу. Якщо в пакеті як адреса призначення вказана адреса класу D (перший октет має діапазон 224 - 239), то такий пакет повинні отримати усі вузли, яким присвоєна ця адреса.

- Адреси з першим октетом в діапазоні 240 - 247 зарезервовані для майбутнього застосування.

У протоколі IP існує декілька угод про особливу інтерпретацію IP -адресов:

- Адреса 0.0.0.0 визначає адреса вузла, який згенерував пакет.
- IP -адрес з нульовим номером мережі означають поточну мережу. Ці адреси дозволяють спілкуватися хостам власної мережі, не знаючи номера мережі, а знаючи тільки клас мережі.

- Адреса 255.255.255.255 використовується для широкомовної передачі в межах власної мережі (limited broadcast).

- IP -адрес з певним номером мережі і одиницями в номері вузла використовується для широкомовної передачі в межах певної мережі (broadcast).

- Адреса 127.0.0.1 зарезервований для організації зворотного зв'язку при тестуванні роботи програмного забезпечення вузла без реальної відправки пакету по мережі. Ця адреса має назву loopback.

Окрім IP -адресов, хосты ідентифікуються доменними (символьними) іменами, легшими для запам'ятовування і відбиваючими логічну структуру мережі і, часто, функціональне призначення того або іншого хоста. Домен - це безліч хостов, об'єднаних в логічну групу. Доменне ім'я складається з символьних полів, розділених точками.

Для відображення доменного імені на IP -адрес використовується DNS (Domain Name System) - доменна служба імен. DNS - це розподілена база даних, підтримувальна ієрархічну систему імен для ідентифікації вузлів в мережі Internet. Служба DNS призначена для автоматичного пошуку IP -адреса по відомому символьному імені вузла. Специфікація DNS визначається стандартами RFC 1034 і 1035. DNS вимагає статичної конфігурації своїх таблиць, що відображують імена комп'ютерів в IP -адрес.

Протокол DNS є службовим протоколом прикладного рівня. Цей протокол несиметричний - в ньому визначені DNS -сервери і DNS -клиенти. DNS -сервери зберігають частину розподіленої

бази даних про відповідність символьних імен і IP -адресов. Ця база даних розподілена по адміністративних доменах мережі Internet. Клієнти сервера DNS знають IP -адрес сервера DNS свого адміністративного домена і по протоколу UDP передають запит, в якому повідомляють відоме символьне ім'я і просять повернути відповідний йому IP -адрес.

- Якщо дані про запитану відповідність зберігаються в базі цього DNS -сервера, то він відразу посилає відповідь клієнтові, якщо ж немає - те він посилає запит DNS -серверу іншого домена, який може сам обробити запит, або передати його іншому DNS -серверу. Усе DNS -сервери сполучені ієрархічно, відповідно до ієрархії доменів мережі Internet. Клієнт опитує ці сервери імен, поки не знайде потрібні відображення. Цей процес прискорюється через те, що сервери імен постійно кешують інформацію, що надається по запитах. Клієнтські комп'ютери можуть використовувати у своїй роботі IP -адреса декількох DNS -серверів, для підвищення надійності своєї роботи.

- База даних DNS має структуру дерева, званого доменним простором імен, в якому кожен домен (вузол дерева) має ім'я і може містити піддомени. Ім'я домена ідентифікує його положення в цій базі даних по відношенню до батьківського домена, причому точки в імені відділяють частини, відповідні вузлам домена.

- Корінь бази даних DNS управляється центром Internet Network Information Center. Домени верхнього рівня розділяються на дві групи - родові домени і домени держав. Для позначення країн використовуються трьохбуквені і двохбуквені аббревіатури, а для різних типів організацій (родових імен) використовуються наступні аббревіатури:

- com - комерційні організації (наприклад, microsoft.com);
- edu - освітні (наприклад, mit.edu);
- gov - урядові організації (наприклад, nsf.gov);
- org - некомерційні організації (наприклад, fidonet.org);
- net - організації, підтримувальні мережі (наприклад, nsf.net).

Повне ім'я домена подібно до повного імені файлу, складається з шляху від цього домена до вершини дерева. Крайнє праве поле означає домен верхнього рівня, далі, справа наліво, слідує піддомени в порядку ієрархічної вкладеності, крайнє ліве поле означає ім'я хоста. Наприклад, ім'я abcdef.hotmail.ru означає - хост abcdef знаходиться в домені hotmail, який у свою чергу знаходиться в домені ru.

Протокол TCP.

Як вже було сказано, протокол TCP є протоколом транспортного рівня. Він призначений для встановлення надійних прямих з'єднань між процесами в багатомережевій системі. Як було вказано раніше, протокол IP забезпечує передачу незалежних пакетів даних (дейтаграмм). При цьому в самому протоколі відсутні засоби забезпечення надійності з'єднань. Таким чином, з використанням Інтернет-протоколу неможливо відстежити втрату якої-небудь дейтаграмми. Це - завдання протоколу TCP.

Основні дії протоколу TCP :

Базова передача даних

Протокол TCP здатний передавати безперервні потоки октетів між своїми клієнтами в обох напрямках, пакуючи деяку кількість октетів в сегменти для передачі через системи Internet. У загальному випадку протоколи TCP вирішують на власний розсуд, коли проводити блокування і передачу даних.

Іноді користувачам буває необхідно переконатися в тому, що усі дані, передані ними протоколу TCP, вже відправлені. Для цієї мети визначена функція проштовхування (push). Щоб переконатися в тому, що дані, відправлені протоколу TCP, дійсно передані, відправник вказує, що їх слід проштовхнути до одержувача.

Простовхування призводить до того, що програми протоколу TCP відразу здійснюють відправлення і, відповідно, отримання даних, що залишаються. Правильно здійснене проштовхування може бути невидимо для одержувача, а сама функція проштовхування може не мати маркера межі запису.

Достовірність

Протокол TCP повинен мати захист від руйнування даних, втрати, дублювання і порушення черговості отримання, що викликаються комунікаційною системою Internet. Це досягається привласненням чергового номера кожному передаваному сегменту, а також вимогою підтвердження (ACK) від програми TCP, що приймає дані. Якщо підтвердження не отримано протягом контрольного інтервалу часу, то дані посилаються повторно. З боку одержувача номера черги використовуються для відновлення черговості сегментів, які можуть бути отримані в неправильному порядку, а також для обмеження можливості появи дублікатів.

Ушкодження фіксуються за допомогою додавання до кожного передаваного сегменту контрольної суми, перевірки її при отриманні і наступній ліквідації дефектних сегментів.

До тих пір, поки програми протоколу TCP продовжують функціонувати коректно, а система Internet не розвалилася повністю на складові частини, помилки пересилки не впливатимуть на правильне отримання даних. Протокол TCP захищає від помилок комунікаційної системи Internet.

Управління потоком

Протокол TCP дає засоби одержувачеві управляти кількістю даних, посланих йому відправником. Це досягається поверненням так званого "вікна" (window) разом з кожним підтвердженням, яке вказує діапазон прийнятних номерів, що йдуть за номером останнього успішно прийнятого сегменту. Вікно визначає кількість октетів, яку відправник може послати до отримання подальших вказівок.

Розподіл каналів

Щоб дозволити на окремо взятому комп'ютері багатьом процесам одночасно використовувати комунікаційні можливості рівня TCP, протокол TCP надає на кожному хост-комп'ютері набір адрес або портів. Разом з адресами мереж і хост-комп'ютерів на комунікаційному рівні Internet вони утворюють сокет (socket - роз'єм).

Кожне з'єднання унікальним чином ідентифікується парою сокетів. Таким чином, будь-який сокет може одночасно використовуватися в багатьох з'єднаннях.

Співвідношення портів і процесів здійснюється кожним хост-комп'ютером самостійно. Проте часто використовувані процеси зв'язуються з фіксованими документованими сокетами (HTTP - порт 80, FTP - 21 і так далі).

Цей сервіс можна згодом використовувати через відомі адреси. Установка і налаштування адрес портів для інших процесів може включати динамічніші механізми.

Робота із з'єднаннями

Механізми управління потоком і забезпечення достовірності, описані вище, вимагають, щоб програми протоколу TCP ініціалізували і підтримували певну інформацію про стан кожного потоку даних. Набір такої інформації, що включає сокети, номери черги, розміри вікон, називається з'єднанням. Кожне з'єднання унікальним чином ідентифікується парою сокетів на двох кінцях.

Якщо два процеси бажають обмінюватися інформацією, відповідні програми протоколу TCP повинні спершу встановити з'єднання (на кожній стороні ініціалізувати інформацію про статус). Після закінчення обміну інформацією з'єднання має бути розірване або закрито, щоб звільнити ресурси для надання іншим користувачам.

Оскільки з'єднання повинні встановлюватися між ненадійними хост-комп'ютерами і через ненадійну комунікаційну систему Internet, то щоб уникнути помилкової ініціалізації з'єднань використовується механізм підтвердження зв'язку з хронометрованими номерами черги.

Пріоритет і безпека

Користувачі протоколу TCP можуть зажадати для свого з'єднання пріоритет і безпеку. Передбачені характеристики з'єднань, що приймаються за умовчанням, коли такі параметри не вимагаються.

Для забезпечення усіх цих можливостей на рівні TCP до даних, переданих від протоколу верхнього рівня, або від прикладної програми, додається заголовок, в якому вказується порт відправника, порт одержувача (по 16 біт), номер посланого пакету, номер підтвердженого пакету, параметри, що управляють, і так далі. Сформований пакет передається Інтернет-

протоколу, який додає свою інформацію, що управляє. При прийомі даних інформація витягається в зворотному порядку.

Протокол HTTP.

Протокол HTTP (Hypertext transfer protocol - протокол передачі гіпертексту) є представником протоколів Інтернет прикладного рівня. Цей протокол працює відповідно до моделі клієнт/сервер. При виконанні роботи два мережеві процеси працюють разом, пересилаючи один одному інформацію. При цьому один процес, що є ініціатором діалогу, є **клієнтом**. Процес, очікуючий запиту і оброблювальний запити клієнтів, є **сервером**.

Протокол HTTP - це протокол запитів/відповідей. Клієнт посилає запит серверу, який його обробляє і посилає відповідь. HTTP -соединение зазвичай відбувається шляхом TCP/IP -соединений. Заданий за умовчанням порт - 80, хоча може використовуватися і інші порти.

Таким чином, відповідно до моделі стека протоколів Інтернету, прикладна програма, що використовує протокол HTTP, повинна за певними правилами (вони будуть розглянуті нижче) сформулювати запит, по протоколу TCP встановити з'єднання, передати серверу сформований запит, отримати по протоколу TCP відповідь, оформлену за правилами протоколу HTTP.

Оскільки протокол HTTP і інші протоколи прикладного рівня використовують поняття URI (Uniform Resource Identifier - універсальний ідентифікатор ресурсу), розглянемо детальніше, як він визначається. Заздалегідь помітимо, що нині поняття URI практично співпадає з поняттям URL (Universal Resource Locator - універсальний адреса ресурсу).



URI складається з наступних частин: тип, ім'я вузла (домена), номер порту, шлях, рядок запити і ідентифікатор фрагмента.

Тип - є протокол (це може бути також ftp, https).

Вузол - ідентифікує машину. Це може бути або доменне ім'я, або IP -адрес.

Порт - номер порту, по якому проходить робота протоколу. Може не вказуватися. У такому разі мається на увазі стандартний порт, використовуваний протоколом (80 для HTTP).

Шлях - є місце розташування запитаного ресурсу. Залежно від налаштувань цей шлях може відповідати деякому шляху на сервері, але може і не відповідати.

Запит - у разі використання сценаріїв в рядку запити можуть передаватися параметри в сценарій. Є набір пар ім'я/значення. Пари розділяються знаком &, а ім'я і значення розділяються знаком =.

Фрагмент - вказує на певний розділ ресурсу.

URI може не містити рядка запиту і ідентифікатора фрагмента.

Для URI не має значення регістр букв (нечутливий до регістра).

URI може бути абсолютним і відносним. Абсолютний URI завжди починається з типу протоколу. Приклад абсолютного URI показаний вище. Відносний URI, без імені протоколу або номера порту, може містити повні шляхи (завжди вказуються з кореневого каталогу і починаються зі знаку слеша /) або відносні шляхи (починаються з символа, отличного от /).

Огляд клієнтських технологій Інтернет. Поняття про мови розмітки. SGML. HTML як застосування SGML. Структура HTML -документа.

Клієнтські застосування для Інтернету можуть бути написані на будь-якій мові програмування з використанням протоколів Інтернету. Ці застосування встановлюються на машинах користувачів і призначені для роботи на певній платформі. Проте існує клас застосувань, що виконуються на будь-якому комп'ютері незалежно від використовуваної платформи і операційної системи. Огляд клієнтських технологій Інтернет. Поняття про мови

розмітки. SGML. HTML як застосування SGML. Структура HTML -документа.

Клієнтська частина середовища розробки Web -приложений складається в основному з 4 частин:

- **браузери** - служать призначеним для користувача інтерфейсом для застосування.
- **HTML** - мова гіпертекстової розмітки. Документи у форматі HTML поступають браузеру для відображення.

- **Клієнтські розширення.** У міру зростання потреб користувачів в активних web - документах з'явилася потреба працювати з виконуваним вмістом (executable content) в браузері. Для самої технології не обов'язково прив'язуватися до роботи конкретного браузера, хоча він і підтримує її. Прикладом клієнтських розширень можуть служити Java -апплеты, елементи управління ActiveX.

- **Клієнтські мови сценаріїв.** На сьогодні основною такою мовою є JavaScript, тісно інтегрований з HTML.

Узагальнена структурована мова розмітки (Structured Generalized Markup Language - SGML) - це система визначення мов розмітки. SGML стандартизований в 1986 році і широко поширений скрізь, де вимагаються великі об'єми структурованого тексту, представленого в легкодоступних форматах. Мова дозволяє розмічати документи, уявляючи інформацію про структуру, представлення і семантику в одному документі. SGML має великий спектр підтримки, оскільки у нього є величезна кількість властивостей, спеціально орієнтованих на роботу з текстовими застосуваннями. Проте ця мова є досить складною (специфікація мови займає більше 500 сторінок) і, отже, важким для реалізації.

Мова SGML визначає базовий синтаксис і дає можливість описувати власні елементи (звідси і термін узагальнений). Кожна мова розмітки, визначена в SGML, називається застосуванням SGML. Застосування SGML характеризується:

- **Оголошенням SGML.** SGML Оголошення вказує, які символи і роздільники можуть відображатися в застосуванні.

- **Визначенням типу документу DTD.** DTD визначає синтаксис конструкцій розмітки. DTD може включати додаткові визначення, наприклад, посилання на комбінації символів.

- **Специфікацію, що описує семантику,** використовувану в розмітці. Ця специфікація також накладає синтаксичні обмеження, які неможливо виразити за допомогою DTD.

- **Екземпляри документу** містять дані (вміст) і розмітку. Кожен екземпляр утримує посилання на DTD, яке повинне використовуватися для інтерпретації.

HTML (Hypertext Markup Language - мова розмітки гіпертексту) є одним із застосувань SGML. HTML надає обмежений набір елементів для структуризації документу, тому його застосування є відносно простим.

Розглянемо, які конструкції SGML використовуються в HTML.

1. Елементи. SGML оголошує типи елементів, представляючі структури або бажану поведінку. HTML включає типи елементів, представляючі абзаци, гіпертекстові посилання, списки, таблиці, зображення і так далі

Кожне оголошення типу елементу зазвичай включає три частини: початковий тег, вміст і кінцевий тег.

Ім'я елементу відображується в початковому тегу (пишеться <имя-елемента>) і в кінцевому тегу (пишеться </имя-елемента>). Деякі елементи можуть не мати вмісту. Наприклад, елемент переходу на наступний рядок BR не має вмісту. У стандарті також визначено, що деякі елементи можуть не мати кінцевого тега або він може бути опущений. Наприклад, елемент абзацу P може не мати кінцевого тега.

2. Атрибути. З елементами можуть бути пов'язані властивості, звані атрибутами, які можуть мати значення (стандартні або встановлювані авторами або сценаріями). Пари атрибут/значення поміщаються перед закриваючою дужкою ">" початкового тега елементу. У початковому тегу елементу може бути будь-яке число (допустимих) пар атрибут/значення, розділених пропусками. Вони можуть вказуватися у будь-якому порядку.

Наприклад, в наступному прикладі для елементу H1 встановлений атрибут id :

<h1 id="section 1">Це заголовок</h1>

3. Посилання на символи. Посилання на символи - це числові або символні імена символів, які можуть бути включені в документ HTML. Вони зручні для звернення до рідко використовуваних символів або до символів, які важко або неможливо вводити в засобах розробки документів. Посилання на символи починаються зі знаку "&" і закінчуються крапкою з комою (;). Ось деякі приклади:

- "<" представляє знак <.
- ">" представляє знак >.
- """ представляє знак ".

4. Коментарі. Коментарі в HTML мають наступний синтаксис:

<!-- це коментар -->

<!-- це теж коментар, він займає декілька рядків -->

Проблеми між відкриваючим роздільником розмітки ("<") і відкриваючим роздільником коментаря ("--") недопустимі, але їх можна використовувати між закриваючим роздільником коментаря ("--") і закриваючим роздільником розмітки (">"). Поширеною помилкою є включення рядка символів перенесення ("---") в коментар. Слід уникати використання в коментарях двох або більш за символи перенесення.

Інформація в коментарях не має спеціального значення (наприклад, посилання на символи не інтерпретуються).

Документ у форматі HTML 4.0 складається з трьох частин:

- рядка, що містить інформацію про версію HTML
- розділу заголовків (визначуваного елементом HEAD)
- тіла, яке включає власне утримуване документу. Тіло може вводитися елементом BODY або елементом FRAMESET.

Перед кожним елементом або після кожного елементу може знаходитися порожній простір (пропуски, перехід на новий рядок, табуляції і коментарі). Розділи 2 і 3 повинні відділятися елементом HTML.

Вот пример простого документа HTML:

```
<!DOCTYPE "-//W3C//ДАТА ГРОМАДСЬКОСТІ HTML HTML
4.0//EN" "http://www.w3.org/КОНЦЕРН/строгий
REC-html40/.дата">
```

```
<HTML>
```

```
<ГОЛОВА>
```

```
<ЗАГОЛОВОК>Мой первый документ HTML</ЗАГОЛОВОК>
```

```
</ГОЛОВА>
```

```
<ТІЛО>
```

```
<P>Привет, мир!
```

```
</ТІЛО>
```

```
</HTML>
```

У документі HTML має бути оголошена використовувана в ній версія мови HTML. **Оголошення типу документу** вказує визначення типу документу (DTD), використовуване в цьому документі.

HTML 4.0 визначає три DTD, так що автори повинні включати у свої документи одне з наступних оголошень типів. Різниця між DTD полягає в підтримуваних ними елементах.

- HTML 4.0 Strict DTD (строге визначення) включає усі елементи і атрибути, що не є небажаними і що не використовуються в документах з кадрами. Для документів, що використовують це DTD, використовуйте таке оголошення типу документу :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
```

```
"http://www.w3.org/TR/REC - html40/strict.dtd">
```

- HTML 4.0 Transitional DTD (перехідне визначення) включає усе, що включено в строге DTD, а також небажані елементи і атрибути (більшість з яких відноситься до візуального

представлення). Для документів, що використовують це DTD, використовуйте таке оголошення типу документу :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
```

- HTML 4.0 Frameset DTD (визначення для кадрів) включає усе, що включено в перехідне DTD, а також кадри. Для документів, що використовують це DTD, використовуйте таке оголошення типу :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

Після оголошення типу документу інша частина документу HTML міститься в елементі HTML. Таким чином, типовий документ HTML має таку структуру:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
```

..Тут йдуть заголовок, тіло і т.д..

```
</HTML>
```

Основні елементи HTML. Елемент HTML. Атрибути lang, dir. Елементи HEAD, TITLE. Атрибут title. Елементи META, LINK, BODY. Атрибути id, class.

Елемент HTML. Ідентифікує документ як що містить елементи HTML. Початковий тег - не обов'язковий. Кінцевий тег - не обов'язковий. Після оголошення типу документу частина документу HTML, що залишилася, міститься в цьому елементі. Усередині цього елементу можуть

знаходитися елементи **HEAD і BODY** для ділення документу на частини.

Атрибути, визначені для елементу, :

- **lang.** Цей атрибут вказує основну мову значень атрибутів елементів і текстового вмісту. За умовчанням значення цього атрибуту не встановлене. Приклади можливих значень : ru (російський), en (англійський) і так далі
- **dir.** Цей атрибут задає основний напрям нейтрального в сенсі напрямку тексту і напрям таблиць. Можливі значення:

LTR: Зліва направо.

RTL: Справа наліво.

Ці атрибути використовуються в основному для допомоги пошуковим машинам, для допомоги програмам перевірки орфографії і тому подібне. Визначення цих атрибутів не є обов'язковим. Ці атрибути можуть бути встановлені і в інших елементах HTML.

Елемент HEAD. Містить інформацію про поточний документ, таку як заголовок, ключові слова, які можуть використовуватися пошуковими машинами, і інші дані, які не рахуються вмістом документу. Початковий тег - не обов'язковий. Кінцевий тег - не обов'язковий. Усередині цього елементу можуть знаходитися елементи TITLE, META, LINK, BASE і деякі інші.

Елемент TITLE. Означає назва документу, яка зазвичай відображується в заголовку вікна браузера. Початковий тег - обов'язковий. Кінцевий тег - обов'язковий. За стандартом застосування цього елементу є обов'язковим. На практиці ця вимога виконується не завжди. Якщо цей елемент присутній в документі, він повинен міститися в розділі HEAD. Заголовки можуть включати комбінації символів (для символів зі

знаком наголосу, спеціальних символів і так далі), але не можуть містити іншої розмітки.

Елемент **TITLE** може мати атрибут, званий title. Цей атрибут встановлюється для практично усіх елементів HTML. Цей атрибут пропонує інформацію про елемент, для якого він

встановлюється. Візуальні браузері часто відображають заголовок як підказку. Установка цього атрибуту для посилання дозволяє повідомити користувачів про природу пов'язаного ресурсу.

Вот образец заголовка документа:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
<HEAD>
<TITLE title="это заголовок">Интернет-технологии</TITLE>
... другие элементы заголовка...
</HEAD>
<BODY>
... тело документа...
</BODY>
</HTML>
```

Елемент META.

Надає різні типи інформації, що не відображується, і інструкцій для браузеру. Це, зокрема, інформація про зміст сторінки, ключові слова, про автора, команди, що управляють, пошуковим роботам і так далі. Початковий тег: обов'язковий, Кінцевий тег: заборонений. Усі елементи META повинні розташовуватися в розділі HEAD.

META -тэги мають два можливі атрибути

- <META HTTP - EQUIV="ім'я" CONTENT="вміст">
- <META NAME="ім'я" CONTENT="вміст">

Атрибут HTTP - EQUIV

META -тэги з атрибутом HTTP - EQUIV еквівалентні HTTP -заголовкам. Зазвичай вони управляють діями браузерів і можуть бути використані для вдосконалення інформації, що видається звичайними заголовками. Теги такої форми можуть дати такий же ефект, що і HTTP - заголовки, і на деяких серверах автоматично можуть бути переведені в справжні HTTP - заголовки.

Expires - дата застарівання. Управління кешуванням в HTTP/1.0. Якщо вказана дата пройшла, то черговий запит цього документу викликає повторний мережевий запит, а не підвантаження документу з кеша. Дата зі значенням "0" інтерпретується як "зараз". Таке значення примушує браузер кожного разу при запиті перевіряти - чи змінювався цей документ. Це, до речі відноситься і до прокси-агентам. Пошукові роботи можуть або зовсім не індексувати такий документ, або постійно перевіряти його.

<META HTTP - EQUIV="expires" CONTENT="Wed, 26 Feb 2012 08:21:57 GMT"> що еквівалентно HTTP -заголовку

Expires: Wed, 26 Feb 2012 08:21:57 GMT

Pragma - контроль кешування для HTTP/1.0. Значенням має бути "no-cache".

Content - Type - вказівка типу документу. Може бути розширено вказівкою кодування сторінки (charset).

<META HTTP - EQUIV="Content-type" CONTENT="text/html; charset=KOI - 8r">

Content - language - вказівка мови документу. Може використовуватися пошуковими машинами при індексуванні сторінок. Комбінація поля Accept - Language (посиланого браузером) з вмістом Content - language може бути умовою вибору сервером тієї або іншої мови.

<META HTTP - EQUIV="Content-language" CONTENT="en-GB">

Refresh - визначення затримки в секундах, після якої браузер автоматично оновлює документ. Додаткова можливість - автоматичне завантаження іншого документу.

<META HTTP - EQUIV="Refresh" Content="3, URL=http://www.name.com/page.html"> що еквівалентно HTTP -заголовку

Refresh: 3; URL=http://www.name.com/page.html

Cache - Control - визначає дії кеша по відношенню до цього документу. Можливі значення:

- **Public** - документ кэшируется в доступних для усіх кешах
- **Private** - тільки в приватному кеші
- **no - cache** - не може бути кэширован
- **no - store** - може бути кэширован, але не зберігається

Атрибут NAME

META -тэги з атрибутом NAME використовуються у випадках, коли поля не співвідносяться з HTTP -заголовками.

Robots - управління індексацією сторінки для пошукових роботів.

<META NAME="Robots" CONTENT="NOINDEX, FOLLOW">

Можливі значення:

- **index** - індексувати цю сторінку
- **follow** - індексувати сторінки, на які є посилання з цій сторінці
- **all** - еквівалентно двом попереднім через кому
- **noindex** - не індексувати сторінку, але йти по посиланнях
- **nofollow** - індексувати, але не йти по посиланнях
- **none** - еквівалентно двом попереднім через кому

За умовчанням CONTENT="index, follow".

Description - коротка анотація змісту документу. Використовується пошуковими системами для опису документу. Довжина тексту до 100 символів.

<META NAME="Description" CONTENT="Документ містить опис елементів HTML">

Keywords - використовується пошуковими системами для індексування документу. Довжина списку до 1000 символів. Не допускається використання одного і того ж ключового слова більше 7 разів, пошукові системи

просто ігноруватимуть це слово.

<META NAME="Keywords" CONTENT="елементи, теги, метадані ">

Document - state - управління індексацією сторінки для пошукових роботів. Визначає частоту індексації - або один раз індексувати, або реіндексувати документ регулярно.

<META NAME="Document-state" CONTENT="Static">

Можливі значення:

- **Static**
- **Dynamic**

Author - звичайне ім'я автора, формат довільний.

<META NAME=" Author " CONTENT="Іванов И. И., 2010 р. ">

Generator - звичайна назва і версія редактора, за допомогою якого створена ця сторінка.

Copyright - звичайний опис авторських прав на документ в довільному форматі

Елемент LINK.

Надає документу незалежний від середовища метод визначення відношення цього документу до інших документів і ресурсів Мережі. За допомогою елементу LINK можна:

- створювати в документі спеціальні навігаційні кнопки або меню
- управляти процесом відображення набору HTML файлів в друкарські документи
- прив'язувати такі асоційовані ресурси, як таблиці стилів і скрипти
- надавати альтернативні форми для цього документу

Початковий тег: обов'язковий, Кінцевий тег: заборонений. Усі елементи LINK повинні розташовуватися в розділі **HEAD**.

Серед атрибутів, визначених для цього елемента, найважливішими є:

- **rel** - визначає тип прямого зв'язку
- **rev** - визначає тип зворотного зв'язку
- **href** - визначає URI пов'язаного ресурсу
- **type** - рекомендований тип вмісту.

Атрибути rel і rev можуть набувати значень:

- **Alternate** - означає альтернативні версії документа, в якому знаходиться посилання.
- **Stylesheet** - означає зовнішню таблицю стилів.
- **Start** - означає перший документ в наборі. Цей тип посилання повідомляє пошукові машини про те, який документ автор вважає початком набору.
- **Next** - означає наступний документ в лінійній послідовності документів.
- **Prev** - означає попередній документ у впорядкованій серії документів.
- **Contents** - означає документ, що служить змістом.
- **Index** - означає документ, що є покажчиком поточного документа.
- **Glossary** - означає документ - глосарій термінів, що відносяться до поточного документа.
- **Copyright** - означає зауваження про авторське право для поточного документа.
- **Chapter** - означає документ, що є главою в наборі документів.
- **Section** - означає документ, що є розділом в наборі документів.
- **Subsection** - означає документ, що є підрозділом в наборі документів.
- **Appendix** - означає документ, що є застосуванням в наборі документів.
- **Help** - означає документ, що містить довідку (детальніша інформація, посилання на інші інформаційні ресурси і так далі)
- **Bookmark** - означає закладку.

Приклад використання цього елемента:

```
<!DOCTYPE "-//W3C//ДАТА ГРОМАДСЬКОСТІ HTML HTML 4.0//EN" "http://www.w
3.org/КОНЦЕРН/строгий REC-html40/.дата">
<HTML>
<ГОЛОВА>
  <ЗАГОЛОВОК>Глава 4</ЗАГОЛОВОК>
  <Заголовок="Початку" ЄДНАЛЬНОГО rel="Первая страница руководства",
type="текстовий/html" href="http://де-небудь.
початок com/керівництва/.html">
  <"Індекс" ЄДНАЛЬНОГО rel=, type="текстовий/html" href="http://де-небудь.індекс
com/керівництва/.html">
  <ЄДНАЛЬНИЙ rel=, "Наступний" type="текстовий/html" href="http://де-
небудь.com/керівництво/Chapter5.html">
  <ЄДНАЛЬНИЙ rel=, "Попередній" type="текстовий/html" href="http://де-
небудь.com/керівництво//Chapter3.html">
  <Заголовок=допомоги ЄДНАЛЬНОГО rel="Страница помощи", type="текстовий/html"
href="допомога.html"> <!-- описаны
ссылки на индекс, первую, следующую и предыдущую страницу, а также страницу
помощи -->
  <єднальний rel="stylesheet" type="текстовий/css" href="://де-небудь.керівництво
com/керівництва/.css">
  <!-- описано местонахождение таблицы стилей руководства -->
```

</ГОЛОВА> ..продолжение документа..

Елемент BODY.

Визначає зміст документа. Початковий тег: не обов'язковий, Кінцевий тег: не обов'язковий.

Визначення найважливіших атрибутів :

background = url. Небажаний. Значення цього атрибуту - URI, що вказує на зображення. Це зображення є фоном.

text = color. Цей атрибут встановлює колір тексту.

link color. Цей атрибут встановлює колір тексту гіпертекстових посилань, по яких перехід не був здійснений.

vlink color. Цей атрибут встановлює колір тексту посилань, по переходу був здійснений.

alink = color. Цей атрибут встановлює колір тексту посилань, коли вони вибрані користувачем.

Усі ці атрибути є небажаними, оскільки переважним є завдання представлення документа за допомогою таблиць стилів.

Все эти атрибуты являются нежелательными, так как предпочтительным является задание представления документа с помощью таблиц стилей.

Кроме этих атрибуты, элемент **BODY** и имеет множество атрибуты, общих для многих других элементов:

- **id, class** (идентификаторы в пределах документа)
- **style** (встроенная информация о стиле)
- **bgcolor** (цвет фона)

Рассмотрим атрибуты **id** и **class**.

Атрибут **id** назначает элементу уникальный идентификатор. Он имеет в HTML несколько ролей:

- Способ выбора таблицы стиля.
- Назначение цели (якорь) для гипертекстовых ссылок.
- Средство ссылки на определенный элемент сценария.
- Имя объявленного объекта OBJECT.
- В целях обработки агентами пользователей (например, для полей идентификации при извлечении данных из страниц HTML в базу данных, преобразовании документов HTML в другие форматы, в сценариях и т.д.).

Атрибут **class**, з іншого боку, призначає одне або декілька імен класів елементу; при цьому елемент може називатися таким, що належить до цих класів. Ім'я класу може використовуватися декількома екземплярами елементу. Атрибут **class** має в HTML декілька ролей:

- Спосіб вибору таблиці стилю (коли автор хоче призначити інформацію про стиль набору елементів).

- Для загальної обробки агентами користувачів

Наприклад: <BODY id="myBody" class="info">..

означає привласнення елементу BODY унікального ідентифікатора з ім'ям myBody. Цей ідентифікатор може потім використовуватися в сценаріях. Також вказується, що до документу має бути клас info з таблиці стилів (помітимо, що розташування таблиці стилів може бути вказане за допомогою елементу LINK).

Елементи форматування тексту. Списки. Таблиці.

Існує досить багато елементів, призначених для форматування тексту документа. Це елементи B, BLOCKQUOTE, BR, Hx, HR, I, P, SUB, SUP і інші.

Елементи Hx (H1, H2, H3, H4, H5, H6). Визначають заголовки тексту. Всього існує 6 рівнів заголовків - від H1 (найбільш важливий, і, отже, найбільший заголовок) до H6 (найменш важливий і найменший заголовок). Початковий тег - обов'язковий. Кінцевий тег - обов'язковий. Для елементу немає специфічних атрибутів, визначені тільки загальні для більшості елементів атрибути - id, class, lang, dir, title і деякі інші.

Елемент P. Визначає параграф (абзац) тексту. Початковий тег - обов'язковий. Кінцевий тег - не обов'язковий. Для елемента немає специфічних атрибутів.

Елемент BR. Призначений для примусового переходу тексту на інший рядок. Початковий тег - обов'язковий, кінцевий тег - заборонений.

При відкритті тексту в браузері можна переконатися, що візуальна дія елемента BR не дорівнює дії елемента P.

Мова HTML пропонує авторам декілька механізмів створення списків інформації. У кожному списку має бути один або декілька елементів списків. Списки можуть містити:

- нерегульовану інформацію.
- Впорядковану інформацію.
- Визначення.

Елемент UL призначений для створення нерегульованих списків, елемент OL - для впорядкованих списків. Ці елементи повинні містити один або декілька елементів LI, що описують елементи списку. Відмінність впорядкованих і нерегульованих списків - тільки у візуальному представленні. Елементи впорядкованого списку нумеруються, а елементи нерегульованого списку відзначаються деяким маркером, вид якого залежить від використовуваного стилю. Для **UL** і **OL** наявність початкового і кінцевого тега обов'язкова. Для **LI** початковий тег обов'язковий, кінцевий - не обов'язковий. Для тегів визначені загальні атрибути і деякий набір небажаних атрибутів :

type = інформація про стиль. Цей атрибут встановлює стиль елемента списку. Доступные в настоящее время значения:

Type	Стиль нумерации	
1	арабские цифры	1, 2, 3, ...
a	буквы нижнего регистра	a, b, c, ...
A	буквы верхнего регистра	A, B, C, ...
i	римские цифры в нижнем регистре	i, ii, iii, ...
I	римские цифры в верхнем регистре	I, II, III, ...

start = число. Тільки для OL. Цей атрибут задає початковий номер першого елемента у впорядкованому списку. За умовчанням початковий номер - "1". Хоча значенням цього атрибуту є ціле число, відповідна мітка може бути нецифрова. Якщо як стиль вибрані латинські букви верхнього регістра (A, B, C, ..), start=3 означає "C". Якщо як стиль вибрані римські цифри нижнього регістра, start=3 означає "iii" і так далі

value = число. Тільки для LI. Цей атрибут встановлює номер поточного елемента списку. Хоча значенням атрибуту є ціле число, відповідна мітка може бути нечислова (див. атрибут start).

compact. Якщо цей логічний атрибут встановлений, він повідомляє візуальних агентів користувачів про те, що генерувати список треба компактніше. Інтерпретація цього атрибуту залежить від агента користувача.

Описаний вище нерегульований список може бути представлений таким чином:

```
<UL>
<LI>Неврегульовану інформацію.
<LI>Впорядковану інформацію.
<LI>Визначення.
</UL>
```

Визначення трохи відрізняються від розглянутих вище списків і створюються з використанням елементів **DL**, **DT**, **DD**.

summary="В этой таблице приводится некоторая статистика о фруктовых мухах: средняя высота и вес, процент мух с красными глазами (особей мужского и женского пола).">

```
<CAPTION> Тестова таблиця із збідненими осередками </CAPTION>
<TR><TH rowspan="2">&nbsp;</TH><TH colspan="2">Средний </TH>
<TH rowspan="2">Красные<BR>глаза</TH></TR></TR>
<TR><TH>высота</TH><TH>вес</TH></TR>
<TR><TH>мужской пол</TH><TD>1.9</TD><TD>0.003</TD><TD>40%</TD></TR>
<TR><TH>женский пол</TH><TD>1.7</TD><TD>0.002</TD><TD>43%</TD></TR>
</TABLE>
```

Візуальне представлення цієї таблиці буде наступним:

Тестовая таблица с объединенными
ячейками

	Средний		Красные глаза
	высота	вес	
мужской пол	1.9	0.003	40%
женский пол	1.7	0.002	43%

Елемент **TABLE** створює таблицю. Початковий тег - обов'язковий. Кінцевий тег - обов'язковий. Усі інші елементи таблиці повинні знаходитися усередині цього елемента. Для цього елемента визначена безліч атрибутів. З них найважливішими є:

- **align** - вирівнювання таблиці відносно документу. Можливі значення: center, left, right
- **border** - товщина рамки в пікселях. Якщо атрибут не вказаний, то таблиця виводиться без видимої рамки
- **bordercolor** - колір рамки
- **cellspacing** - задає відстань між елементами таблиці
- **cellpadding** - задає відстань між вмістом осередку і її рамкою
- **rules** - описує рамки навколо таблиці. Може набувати наступних значень

<i>all</i>	Отображает все части рамки внутри таблицы
<i>cols</i>	Отображает все вертикальные рамки внутри таблицы
<i>groups</i>	Отображает горизонтальные части рамки между группами таблицы THEAD, TBODY, TFOOT
<i>none</i>	Удаляет все рамки вокруг таблицы
<i>rows</i>	Отображает все горизонтальные рамки внутри таблицы

▪ **summary** - коротка інформація про призначення і структуру таблиці для агентів користувачів, що виконують генерацію для невізуальних засобів

▪ **width** - ширина таблиці у відсотках або пікселях. Якщо значення вказане у відсотках, це означає долю у відсотках від доступного горизонтального простору.

Елемент **CAPTION** визначає заголовок таблиці. Він повинен розташовуватися безпосередньо після початкового тега елементу **TABLE** і може зустрічатися в таблиці тільки один раз. Початковий тег - обов'язковий. Кінцевий тег - обов'язковий.

Рядки таблиці можуть групуватися в заголовки, нижні заголовки і один або декілька розділів тіла таблиці за допомогою елементів **THEAD**, **TFOOT** і **TBODY** відповідно. Заголовки таблиці повинні містити інформацію про стовпці таблиці. Тіло таблиці повинне містити рядки даних таблиці.

Якщо елементи **THEAD**, **TFOOT** і **TBODY** присутні, кожен з них містить групу рядків. Кожна група рядків повинна містити принаймні один рядок, визначуваний елементом **TR**. Елементи **TR** служать контейнерами для рядка елементів таблиці.

У прикладі показаний порядок і структура заголовків таблиці, нижніх заголовків і тел.

```
<TABLE>
<THEAD>
  <TR> ...заголовок...
</THEAD>
<TFOOT>
  <TR> ...нижній заголовок...
</TFOOT>
<TBODY>
  <TR> ...первая строка данных блока 1...
  <TR> ...вторая строка данных блока 1...
</TBODY>
<TBODY>
  <TR> ...первая строка данных блока 2...
  <TR> ...вторая строка данных блока 2...
  <TR> ...третья строка данных блока 2...
</TBODY>
</TABLE>
```

Елемент **TFOOT** повинен розташовуватися до елементу **TBODY** у визначенні **TABLE**, щоб агенти користувачів могли генерувати нижній заголовок до отримання усіх (можливо, численних) рядків даних. Нижче наводиться зведення обов'язкових тегів і тегів, які можна опустити, :

- Початковий тег **TBODY** завжди обов'язковий, якщо тільки таблиця не містить єдиного тіла без верхніх і нижніх заголовків. Кінцевий тег **TBODY** завжди можна опускати.

- Початкові теги для елементів **THEAD** і **TFOOT** обов'язкові, якщо в таблиці присутні верхній і нижній заголовки, але відповідні кінцеві теги можна опускати.

Розділи **THEAD**, **TFOOT** і **TBODY** повинні містити однакове число стовпців.

Кожен рядок таблиці (описана елементом **TR**) містить одну або декілька елементів таблиці. Елементи таблиці описуються елементами **TH** і **TD**. При цьому **TH** використовується для опису осередків заголовка, а **TD** - для опису осередків даних, що знаходяться в тілі таблиці. Для цих елементів початковий тег - обов'язковий, кінцевий - не обов'язковий. Серед безлічі атрибутів, визначених для осередків, найбільш

важливими є: **rowspan** = число. Цей атрибут визначає число рядків, об'єднаних поточним осередком. За умовчанням використовується значення один ("1"). Значення нуль ("0") означає, що осередок об'єднує усі рядки від поточної до останнього рядка таблиці.

colspan = число. Цей атрибут визначає число стовпців, об'єднаних поточним осередком. За умовчанням використовується значення один ("1"). Значення нуль ("0") означає, що осередок об'єднує усі стовпці від поточного до останнього стовпця таблиці.

Наведемо приклад складної таблиці, в якій деякі осередки об'єднані, :

```
<TABLE border="1">
```

```

<TR><TD>1 </TD><TD rowspan="2">2,5 </TD><TD>3</TD></TR>
<TR><TD>4 </TD><TD>&nbsp;</TD></TR>
<TR><TD colspan="3">7-9</TD></TR>
</TABLE>

```

Зовнішній вигляд таблиці буде наступним:

1		3
4	2,5	
7-9		

Тут об'єднано два осередки по вертикалі. Для цього використовувався атрибут `rowspan`. Для об'єднання осередків в рядку (осередки 7 -9) використовувався атрибут `colspan`.

У цьому ж прикладі показано застосування символу нерозривного прикладу ` `, який часто застосовується, якщо вимагається зберегти форматування таблиці за відсутності вмісту в осередку.

Окрім цих атрибутів, часто використовуються атрибути вирівнювання вмісту осередків

- **align** - вирівнює текст в осередку
- **left** - по лівому краю (за умовчанням)
- **right** - по правому краю
- **center** - по центру
- **justify** - по ширині.
- **valign** - вирівнює текст в осередку по вертикалі
- **top** - по верхньому краю
- **middle** - по центру
- **bottom** - по нижньому краю
- **baseline** - по базовій лінії.

Ці ж атрибути можуть використовуватися в елементі `TR`. У такому разі вони діють на усі осередки рядка.

Посилання. Зображення. Об'єкти

Посилання зв'язують один ресурс Інтернету з іншим. Посилання має два кінці - званих якорями - і напрям. Посилання починається в "початковому" якорі (джерелі) і вказує на "цільовий" якор, який може бути будь-яким ресурсом Web (наприклад, зображенням, відеокліпом, звуковим файлом, програмою, документом HTML, елементом в документі HTML і так далі). За умовчанням з посиланням пов'язано завантаження іншого ресурсу Web. Ця поведінка досягається зазвичай шляхом вибору посилання (наприклад, за допомогою клацання миші, введення з клавіатури і так далі).

Для визначення якорів використовується елемент `A`. Початковий тег - обов'язковий. Кінцевий тег - обов'язковий. Посилання і якорі, визначувані елементом `A`, не можуть бути вкладеними; елемент `A` не повинен містити інших елементів `A`. Має безліч атрибутів, серед яких найважливішими є, :

name - дає якорю ім'я, так що він може служити метою іншого посилання. Значенням цього атрибуту має бути унікальне ім'я. Зона дії імені - поточний документ.

href - визначає місце розташування ресурсу Web, визначаючи зв'язок між поточним елементом (початковим якорем) і цільовим якорем, визначуваним цим атрибутом.

Кожен елемент `A` визначає якор:

- Вміст елемента `A` визначає положення якоря.
- Атрибут `name` задає ім'я якоря, так що він може служити пунктом призначення будь-якого числа посилань.

- Атрибут href призначає якір пунктом призначення рівно одного посилання.

У наступному HTML -фрагменті міститься два посилання, у однієї цільовим anchor являється документ HTML з ім'ям "chapter 2.html", а у другій цільовий anchor - зображення у форматі GIF, розташоване у файлі "forest.gif":

```
<BODY> ..якийсь текст..
```

```
<P>Детальніше за см в <A href="chapter 2.html">главі два</A>.
```

```
См також <A href="/images/forest.gif">карту лісу.</A>
```

```
</BODY>
```

Визначивши якір, ми можемо зв'язати його з цим же або з іншим документом. URI, що призначають якір, включають символ "#", за яким слідує ім'я якоря (ідентифікатор фрагмента). Ось декілька прикладів таких URI :

- Абсолютний URI: <http://www.mycompany.com/one.html#anchor> - one

- Відносний URI: [./one.html#anchor](#) - one або [one.html#anchor](#) - one

- Якщо посилання визначене в тому ж документі: [#anchor](#) - one

Ось приклад визначення і використання якорів усередині одного документу:

```
<H1>Зміст;</H1>
```

```
<P><A href="#section1">Вступ</A><BR>
```

```
<A href="#section2">Передісторія</A><BR> ..продовження змісту..
```

```
..тіло документу..
```

```
<H2><A name="section 1">Вступ</A></H2> ..розділ 1..
```

```
<H2><A name="section 2">Передісторія</A></H2> ..розділ 2..
```

Мова HTML дозволяє включати в сторінки зображення, аплети, мультимедійні об'єкти і тому подібне. Для цього використовуються наступні елементи:

Елемент IMG використовується для вставки зображень в документи. Для цього елементу початковий тег - обов'язковий, кінцевий тег - заборонений. Найважливішими атрибутами елементу являються:

src - задає місце розташування зображення.

alt - вказує альтернативний текст. Не є обов'язковим. Використовується текстовими браузерами - цей текст показується замість зображення. Звичайні браузери використовують цей текст як підказку.

width, height - визначають розмір зображення. Також не є обов'язковими, але можуть бути використані для точнішої розмітки документу.

align - визначають вирівнювання зображення відносно навколишнього тексту. Має значення:

- **bottom**: означає, що вікно об'єкту має бути вертикально вирівняне відносно поточної базової лінії. Це значення використовується за умовчанням.

- **middle**: означає, що центр об'єкту має бути вирівняний вертикально відносно поточної базової лінії.

- **top**: означає, що верх об'єкту має бути вертикально вирівняний відносно верху поточного текстового рядка.

- **left** і **right**, призводять до переміщення зображення до поточного лівого або правого поля.

Приклад використання елементу **IMG** :

```
<IMG src="589.jpg" alt="вид з мого вікна">
```

```
<IMG src="589.jpg" width="100" height="200">
```

Сучаснішим способом вставки зображень є використання елементу **OBJECT**. Він дозволяє вставляти не лише зображення, але і складніші об'єкти - документи Word, анімацію, Flash -фільми і так далі

Наприклад, зображення можна вставити таким чином:

```
<OBJECT data="589.jpg" type="image/jpeg" width="100" height="200"> вид з мого вікна
</OBJECT>
```

Документ Word може бути вставлений в HTML таким чином:

```
<OBJECT data="lecture 8.doc" type="application/msword"></OBJECT>
```

Відеокліп:

```
<OBJECT data="http://mysite.com/myvideo.avi" type="application/avi"></OBJECT>
```

Частина коду, створювана Flash для публікації ролика :

```
<OBJECT classid="clsid: D27CDB6E - AE6D - 11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
WIDTH="550"
HEIGHT="400" id="myflash" ALIGN="">
<PARAM NAME=movie VALUE=" myflash.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
</OBJECT>
```

Таким чином, елемент **OBJECT** надає загальний механізм включення об'єктів різної природи в сторінку. Для цього елементу наявність початкового і кінцевого тега є обов'язковою.

Найважливіші атрибути:

classid - використовується для вказівки місця розташування об'єкту за допомогою URI. Він може використовуватися разом з атрибутом data або як альтернатива йому, залежно від типу об'єкту.

codebase - визначає базовий шлях, використовуваний для дозволу відносних адрес URI, що задаються в атрибутах classid, data і archive. Якщо цей атрибут відсутній, значенням за умовчанням є базова адреса URI поточного документу.

codetype - визначає тип вмісту даних, отримання яких слід чекати при завантаженні об'єкту, що задається атрибутом classid. Цей атрибут не є обов'язковим, але рекомендується, якщо використовується атрибут classid, оскільки він дозволяє агенту користувача уникнути завантаження інформації для типу утримуваного, який він не підтримує. Якщо цей атрибут відсутній, за умовчанням використовується значення атрибуту type.

data - використовується для вказівки місця розташування даних об'єкту, наприклад, даних зображення для об'єктів, що визначають зображення. Якщо дається відносна адреса URI, він повинен інтерпретуватися відносно атрибуту codebase.

type - визначає тип вмісту для даних, що задаються атрибутом data. Цей атрибут не є обов'язковим, але рекомендується, якщо використовується атрибут data.

archive - використовується для визначення розділеного пропусками списку адрес URI архівів, що містять ресурси, що відносяться до об'єкту, який може включати ресурси, що задаються атрибутами classid і data. Попереднє завантаження архівів приведе до зменшення часу завантаження об'єкту. Архіви, вказані у вигляді відносних адрес URI, повинні інтерпретуватися відносно атрибуту codebase.

У більшості браузерів є вбудовані механізми для генерації основних типів даних, таких як текст, зображення, кольори, шрифти і ряд графічних елементів. Для генерації типів даних, які браузери не підтримують за умовчанням, вони зазвичай запускають зовнішні застосування.

Елемент **OBJECT** дозволяє авторам об'єктів управляти генерацією даних - задавати зовнішню генерацію або використання деякої визначуваної автором програми, що генерує дані в браузері.

У загальному випадку автор повинен визначити три типи інформації :

- Реалізація включеного об'єкту. Наприклад, якщо включений об'єкт - аплет, автор повинен вказати місце розташування виконуваного коду аплету.
- Генеровані дані. Наприклад, якщо включений об'єкт є програмою, що генерує дані шрифту, автор повинен вказати місце розташування цих даних.
- Додаткові значення, необхідні об'єкту. Наприклад, деяким аплетам можуть бути потрібні початкові значення для їх параметрів.

Елемент **PARAM** визначає набір значень, які можуть знадобитися об'єкту під час роботи. Для цього елементу початковий тег обов'язковий, кінцевий - заборонений. Набір елементів **PARAM** повинен розташовуватися усередині елементу **OBJECT**. Елемент має наступний найважливіші атрибути:

- **name** - визначає ім'я робочого параметра, яке має бути об'єкту, що зрозуміло вставляється. Чи враховує ім'я властивості реєстр, залежить від конкретної реалізації об'єкту.
- **value** - визначає значення робочого параметра, що задається атрибутом name. Значення властивостей не мають значення в HTML; їх значення визначається об'єктом.
- **valuetype** - визначає тип атрибуту value. Можливі значення:
- **data**: Це значення використовується за умовчанням. Воно означає, що значення, що задається атрибутом value, визначатиметься і передаватиметься в об'єкт у вигляді рядка.
- **ref**: Значення, що задається атрибутом value, є адресою URI ресурсу, де зберігаються робочі значення.
- **object**: Значення, що задається атрибутом value, є ідентифікатором, що посилається на оголошення **OBJECT** в цьому ж документі. Ідентифікатором має бути значення атрибуту id для оголошеного елементу **OBJECT**.
- **type** - задає тип утримуваного ресурсу, що призначається атрибутом value тільки у разі, якщо значенням атрибуту valuetype є "ref". Таким чином, цей атрибут визначає для агента користувача тип значень, які знаходитимуться за адресою URI, призначеною атрибутом value.

Елементи **DIV** і **SPAN**. Таблиці стилів.

HTML -документ може мати структуру, для якої недостатньо наявні елементи HTML. Елементи **DIV** і **SPAN** разом з атрибутами id і class забезпечують загальний механізм додавання структури в документи. Ці елементи визначають вбудовувану інформацію (**SPAN**) або інформацію рівня блоку (**DIV**), але не накладають ніяких інших виразів для представлення контексту. Припустимо, необхідно згенерувати документ у форматі HTML на основі бази даних інформації про клієнтів. Оскільки HTML не включає елементів для ідентифікації таких об'єктів як "клієнт", "номер телефону", "адреса електронної пошти" і так далі, використовуються елементи **DIV** і **SPAN** для досягнення потрібних ефектів структури і представлення. Для структуризації інформації може бути використаний елемент **TABLE** таким чином:

```
<DIV id="client-boyera" class="client">
<P><SPAN class="client-title">Інформація про клієнта: </SPAN>
<TABLE class="client-data">
<TR><TH>Прізвище:<TD>Boyera</TR>
<TR><TH>Ім'я:<TD>Stephane</TR>
<TR><TH>Тіл:<TD>(212) 555-1212</TR>
<TR><TH>Email:<TD>sb@foo.org</TR>
</TABLE>
</DIV>
```

Для елементів **DIV** і **SPAN** наявність початкового і кінцевого тега обов'язкова. Відмінність між цими елементами полягає у візуальному представленні - браузери ставлять переведення рядка до початку і після закінчення елементу **DIV**. У разі елементу **SPAN** це не відбувається.

Найчастіше ці елементи застосовуються у поєднанні з певними стилями, що дають можливість візуально представити структуру документу. Стель - це набір правил оформлення і форматування, який може бути застосований до різних елементів сторінки. Визначення стільового оформлення HTML -документа проводиться за допомогою каскадних таблиць стилів (CSS - Cascading Style Sheets).

Існує цілих чотири способи зв'язування документу і таблиці стилів :

- **Зв'язування** - дозволяє використовувати одну таблицю стилів для форматування багатьох сторінок HTML

- **Впровадження** - дозволяє задавати усі правила таблиці стилів безпосередньо в самому документі

- **Вбудовування** в теги документу - дозволяє змінювати форматування конкретних елементів сторінки

- **Імпортування** - дозволяє вбудовувати в документ таблицю стилів, розташовану на сервері

Зв'язування - проводиться за допомогою елементу LINK :

```
<LINK REL="stylesheet" TYPE="text/css" HREF="URL">
```

Впровадження - варіант, при якому опис стилів розташовується в документі, усередині елементу HEAD з використанням елементу STYLE :

```
<STYLE type="text/css">
```

```
<!-- приховуємо таблицю для старих браузерів
```

```
-->
```

```
</STYLE>.
```

Параметр type="text/css" є обов'язковим і служить для вказівки браузеру використовувати CSS.

Вбудовування в теги документу - производится за допомогою параметра style, використовуюваного в більшості елементів HTML. Цей метод небажаний, він призводить до втрати однієї з основних переваг CSS - можливості відділення інформації від опису оформлення інформації.

Імпортування - в елементі STYLE можна імпортувати зовнішню таблицю стилів за допомогою властивості @import таблиці стилів :

```
@import: url(mystyles.css);
```

Його слід задавати на початку стільового блоку або зв'язуваної таблиці стилів перед завданням інших правил. Значення властивості @import являється URL файлу таблиці стилів.

Розглянемо правила формування таблиць стилів. Усі оголошення CSS (селектори) записуються у фігурних дужках:

```
ЕЛЕМЕНТ {характеристика: величина}
```

Існує досить великий набір характеристик, визначених для кожного елементу. Ці характеристики визначають властивості шрифту, тексту, кольору і фону, властивості, пов'язані з розмірами і рамками, і так далі.

Деякі характеристики будуть розглянуті нижче.

Наприклад, щоб забарвити усі заголовки першого рівня в червоний колір, можна написати:

```
H1 {color: red} чи H1 {color: #FF0000}
```

Групування. Можна присвоїти відразу групу властивостей декільком елементам. Для цього елементи перераховуються через кому, а кожна властивість відділяється від іншого крапкою з комою:

```
H1, H2 {font - size: 14pt; font - family: Arial; text - decoration: underline}
```

Цей запис означає, що усі заголовки першого і другого рівня відображаються підкресленими з використанням шрифту Arial величиною 14 пікселів.

Спадкоємство. У документі HTML деякі елементи можуть міститися усередині інших елементів. Якщо для зовнішнього елемента визначені правила форматування, а для вкладеного - ні, вкладений елемент наслідуює правила форматування елемента-батька.

Деякі властивості не наслідують вкладеними елементами від своїх батьків, наприклад властивість `background`, але за умовчанням вкладені елементи відображатимуться з фоном батьківського елемента.

Спадкоємство корисне при завданні значень властивостей, що застосовуються до документу за умовчанням. Для цього досить задати усі властивості для елемента, що породжує усі інші елементи сторінки HTML. Таким елементом

є тіло документу, визначуване BODY:

```
BODY {color: black;
font - family: "Times New Roman";
font - size: 12 pt;
background: url(picture.gif) white;}
```

Приведені правила задають форматування документу за умовчанням: чорним шрифтом Times New Roman з фоном, графічним файлом `picture`, що задається `.gif`, або на білому фоні, якщо файл недоступний.

Використання **спадкоємства** дозволяє використовувати для форматування не абсолютні, а відносні величини. Наприклад, використовуючи правила форматування документу, описані вище, ми можемо встановити властивості для елемента SPAN таким чином:

```
SPAN {font - size: 120% }
```

Контекстні селектори. Спадкоємство дозволяє застосовувати стилі контекстного. Це означає, що селектор буде застосований до елемента тільки у тому випадку, якщо він знаходиться усередині певного елемента. Наприклад, визначення

`P SPAN {font - size: 120%}` означає, що розмір шрифту для елемента SPAN буде збільшений на 120% тільки у тому випадку, коли цей елемент знаходиться усередині елемента P. У усіх інших випадках використовуватимуться або значення за умовчанням (визначені в браузері), або визначені в таблиці стилів для цього елемента значення.

Каскади. Ідея каскадних таблиць припускає, що є можливість одночасного використання різних таблиць стилів для одного документу. У кожного браузера є свій стиль, визначений за умовчанням для представлення документів. Коли браузер завантажує HTML -документ, він показує його своїм стилем. Якщо ж документ посилається на таблицю стилів, будуть відображені особливості, задані в цій таблиці. Якій таблиці буде віддано перевагу, задається деякою системою правил :

- Визначається, чи немає протиріч в завданні параметрів якого-небудь елемента. Якщо є, використовуються батьківські параметри. Якщо немає батьківських параметрів, використовуються параметри за умовчанням.

- Якщо присутній конфлікт між авторським стилем і стилем, визначеним за умовчанням, перевага віддається авторським параметром.

- Якщо конфліктують два стилі і один застосовується тільки в цій ситуації, а інший - в усіх випадках, перевага віддається першому (пріоритет "по специфічності").

Така система пріоритетів дозволяє одному документу посилатися на декілька таблиць стилів.

Класи. CSS реалізує можливість привласнювати стилі не усім однаковим елементам сторінки, а вибірково - для цього використовується параметр `CLASS = "ім'я класу"`, що привласнюється будь-якому елементу сторінки. Клас дозволяє задати різні правила форматування для одного елемента певного типу або усіх елементів документу. Ім'я класу вказується в селекторі правила після імені тега і відділяється від нього точкою. Можна визначити декілька правил форматування для одного елемента і за допомогою параметра `CLASS` відповідного тега застосовувати різні правила форматування. Наприклад, можна визначити два класи для відображення заголовка першого рівня :

H1.red {color: red}

H1.blue (color: red; background - color: blue)

У тексті документу посилання на відповідний клас задається в параметрі CLASS :

<H1 CLASS="red">Червоний шрифт</H1>

<H1 CLASS="blue">Червоний шрифт на синьому фоні</H1>

У наведеному прикладі класи задавалися для різного відображення елементів одного типу. Якщо клас повинен застосовуватися до усіх елементів документу, то в селекторі задається ім'я класу з лідируючою точкою без вказівки конкретного елемента : .red {color: red} .blue (color: red; background - color: blue)

Тепер два класи red і blue можна застосовувати до будь-яких елементів документу :

<P CLASS="red">Червоний шрифт</P>

<P CLASS="blue">Червоний шрифт на синьому фоні</P>

Перший абзац відіб'ється червоним шрифтом, а другий - червоним шрифтом на синьому фоні. Розглянемо деякі характеристики, визначені в таблицях стилів.

Властивості шрифту

font-family	Використовується для указання шрифту та шрифтового сімейства, котрым будет відображатся елемент. P {font-family: Times New Roman, sans-serif;}
font-style	Задає спосіб зображення шрифту : normal - Нормальний (за умовчанням), italic - Курсив, oblique - Похилий. P {font-style: italic;}
font-weight	Визначає міра жирності шрифту за допомогою параметрів: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 B {font-weight: bolder;}
font-size	Встановлює розмір шрифту. Параметр може вказуватися як у відносній (відсотки), так і абсолютній величині (пункти, піксели, сантиметри) H1 {font-size: 200%;} H2 {font-size: 150px;}

Колір елемента та колір фону

color	Визначає колір елемента I {color: green;}
background-color	Встановлює колір фону для елемента. H4 {background-color: yellow;}
background-image	Встановлює URL фоновой картинки TABLE {background-image:url(background.gif);}

Властивості тексту

text-decoration	Встановлює ефекти оформлення шрифту, такі, як підкреслення або закреслений текст Прикраса тексту H4 { : underline;} - подчеркивание Прикраса тексту : ніщо;} - стандартный текст Я {прикраса тексту : безпересадочний для лінії;} - зачеркивание B {text-decoration:overline;} - надчеркивание
-----------------	---

text-transform	<p>Задає перетворення регістра тексту при відображенні</p> <p>H4 {text - transform: capitalize;} - Перша буква кожного слова перетвориться в заголовну</p> <p>A {text - transform: uppercase;} - Усі букви перетворюються в заголовні</p> <p>I {text - transform: lowercase;} - Усі букви перетворюються в рядкові</p> <p>B {text - decoration: none;} - Відміняє встановлені перетворення</p>
text-align	<p>Визначає вирівнювання елементу. Можливі значення: left, right, center, justify</p> <p>P {text-align: justify}</p> <p>H5 {text-align: center}</p>
text-indent	<p>Встановлює відступ першого рядка тексту. Найчастіше використовується для створення параграфів з табульованим першим рядком.</p> <p>P {text-indent: 50pt;}</p>
line-height	<p>Управляє інтервалами між рядками тексту.</p> <p>P {line-height: 50 %}</p>
word-spacing	<p>Встановлює інтервалом між словами. Можна використовувати негативні значення</p> <p>P {word-spacing: 50 %}</p>
letter-spacing	<p>Встановлює інтервалом між буквами</p> <p>P {letter-spacing: 50 pt}</p>
vertical-align	<p>Встановлює вертикальне положення елементу. Може набувати наступних значень: baseline middle sub super text-top text-bottom top bottom</p> <p>P {vertical-align: 50 pt}</p>

МЕЖІ. (Показані деякі загальні властивості. Крім того, є можливість встановити стиль кожної сторони рамки).

border-width	<p>Ширина межі. Може бути задана значенням в пікселях або за допомогою зарезервованих слів thin (тонка), medium (середня), thick (товста).</p> <p>table {border-width: 2px;}</p>
border-color	<p>Колір межі. Може набувати значення transparent для завдання невидимої, але такої, що має ширину, межі, що іноді корисно</p> <p>table {border-color: green;}</p>
border-style	<p>Задає стиль рисования границы. Может принимать следующие значения: none (по умолчанию), hidden, dotted, dashed, solid, double, groove, ridge, inset, outset</p> <p>table {border-style: dashed;}</p>

Курсори

	<p>Курсор із стрілками в усі чотири сторони і точкою в центрі, що показує на можливість скролінгу сторінки в будь-якому напрямі</p> <p>I {cursor: all-scroll;}</p>
auto	<p>За умовчанням. Браузер визначає самостійно, який курсор вимагається у даному контексті</p> <p>I {cursor: auto;}</p>

col-resize	Курсор із стрілками вліво-управо і вертикальною розділяючою смужкою. Використовується для індикації можливості зміни розмірів по горизонталі H4 {cursor: col-resize;}
crosshair	Курсор-хрест H4 {cursor: crosshair;}
default	Стандартний курсор, використовуваний системою H4 {cursor: default;}
hand	Рука з витягнутим вказівним пальцем. Використовується при гіперпосиланні H4 {cursor: hand;}
help	Стрілка зі знаком питання. H3 {cursor: help;}
move	Курсор з 4 стрілками, що показує можливість переміщення H2 {cursor: move;}
no-drop	Рука з перекресленим кружечком. Не можна скинути об'єкт в поточну позицію курсора TD {cursor: no-drop;}
not-allowed	Ist durchgekreuzt Kreis. Diese Operation nicht aufrechterhält TD {cursor: not-allowed;}
pointer	Ідентична стилю hand TD {cursor: pointer;}
progress	Пісочний годинник, що показує на продовження операції TD {cursor: progress;}
row-resize	Курсор із стрілками вгору-вниз і вертикальною розділяючою смужкою. Використовується для індикації можливості зміни розмірів по вертикалі TD {cursor: row-resize;}
text	Текстовий курсор-каретка TD {cursor: text;}
url(uri)	Ваш власний курсор. Підтримуються файли.cur и .ani TD {cursor:url(logo.cur);}
vertical-text	Горизонтальна текстова каретка для вертикального тексту TD {cursor: vertical-text;}
wait	Курсор, що показує, що система зайнята і вимагається почекаати TD {cursor: wait;}
*-resize	Курсори, що показують можливість потягнути за край вікна. Замість символу * використовуйте N, NE, NW, S, SE, SW, E, або W, що визначають напрям стрілок TD {cursor: n-resize;}

Природно, приведений вище список властивостей далеко не повний. Тут не показані властивості, що дозволяють управляти видимістю елементів і їх розмірами, порядком розташування їх на сторінці. Не показані властивості, що дозволяють міняти вид смуг прокрутки і так далі. Цю інформацію можна знайти в будь-якому довіднику по HTML.

HTML -редакторы.

Хоча HTML -документи можна створювати і редагувати в будь-якому текстовому редакторі (наприклад в "Блокноті" Windows), на ринку існують безліч спеціалізованих редакторів. Вони діляться на дві основні категорії: невізуальні і візуальні редактори.

Першими користуються в основному професіонали, які в змозі писати код в будь-якому текстовому редакторі. Невізуальні редактори html дозволяють створювати найчистіший програмний код, як мовиться, що не містить нічого зайвого. Використовуючи таку програму, завжди точно знаєш, що отримаєш в результаті певної дії. Невізуальні редактори html

дозволяють вставляти там, де це треба, цілі блоки коду, але при цьому, на відміну від візуальних редакторів, не додають в нього нічого непотрібних рядків на тему розмітки сторінки. Прикладами таких являються "HTML Pad" і "CSE HTML Validator Professional".

HTMLPad - один з лідерів серед редакторів в категорії "для професіоналів". Підтримує можливість редагування коду і швидкої вставки основних елементів розмітки для html, CSS, PHP, JavaScript, VBScript, ASP, SSI. Вбудоване підсвічування коду, створення і редагування власних меню користувача, можливість створення нових елементів для швидкої вставки, що дуже прискорює розробку після налаштування під потреби користувача.

CSE HTML Validator Professional - потужна утиліта для перевірки написаного html - кода. Знаходить помилки в коді сторіночки і виводить їх список користувачеві, містить вбудований редактор html коду, конвертор тегів, шаблони, майстри підказок. Дуже зручна річ для пошуку помилок перед публікацією сайту в інтернет.

Робота з візуальними редакторами (їх ще називають WYSIWYG -редакторами від заголовних букв вираження What You See Is What You Get - "що бачите, то і отримаєте") зводиться до редагування, при якому редагований матеріал в процесі редагування виглядає в точність так само, як і кінцевий результат. HTML -код при цьому генерується автоматично. Серед таких "Microsoft FrontPage", "Macromedia Dreamweaver", "Namo WebEditor"

Microsoft FrontPage - один з простих редакторів html для новачків. Містить величезний набір інструментів і дозволяє з успіхом використовувати програму навіть самому недосвідченому творцеві сайтів. Ідеально підходить початківцям для створення своєї власної домашньої сторіночки. Цей редактор html містить великий набір готових шаблонів сторінок, фонів, кнопок і тому подібне

Недоліки: цей редактор html створює дуже надлишковий код, що сильно впливає на вагу сторінки і збільшує швидкість її завантаження. Переваги: хороша сумісність з іншими продуктами від Microsoft.

Macromedia Dreamweaver - візуальний редактор html. Влаштує і початкуючого творця сайтів і досвідченішого вебмастера. У деяких вебстудіях є стандартом де-факто використання цієї програми, але на суб'єктивну думку автора - це не найзручніша річ для професійного розробника. Великим плюсом програми є відмінна сумісність програми з іншими продуктами фірми Macromedia.

Namo WebEditor - досить непоганий візуальний редактор html. Також підходить і початківцям, і фахівцям середньої кваліфікації. Містить велику кількість готових шаблонів і безліч окремих елементів для створення кнопок і банерів, підтримує Java, СУБД і динамічну навігацію.

Також в Інтернеті є безліч online -редакторів, що не вимагають установки на комп'ютер користувача, - досить зайти на відповідний сайт і почати редагування в браузері.

Інші служби мережі інтернет.

Термінальний режим

Історично однією з ранніх є служба видаленого управління комп'ютером Telnet. Підключившись до видаленого комп'ютера по протоколу цієї служби, можна управляти його роботою. Таке управління ще називають консольним або термінальним.

У минулому цю службу широко використовували для проведення складних математичних розрахунків на видалених обчислювальних центрах. Так, наприклад, якщо для дуже складних обчислень на персональному комп'ютері вимагалися тижні безперервної роботи, а на видаленій супер-ЕОМ всього декілька хвилин, то персональний комп'ютер застосовували для видаленого введення даних в ЕОМ і для прийому отриманих результатів.

В наші дні у зв'язку з швидким збільшенням потужності персональних комп'ютерів необхідність в подібній послугі скоротилася, але, проте, служби Telnet в Інтернеті продовжують існувати. Часто протоколи Telnet застосовують для дистанційного керування технічними об'єктами, наприклад телескопами, відеокамерами, промисловими роботами.

Кожен сервер, надаючий Telnet -услуги, зазвичай пропонує своє клієнтське застосування. Його потрібно отримати по мережі, встановити на своєму комп'ютері, підключитися до сервера

і працювати з видаленим устаткуванням. Простий клієнт Telnet входить до складу операційної системи Windows (файл telnet.exe).

Електронна пошта (E - Mail)

Ця служба також є однією з найбільш ранніх. Її забезпеченням в Інтернеті займаються спеціальні поштові сервери.

Поштові сервери отримують повідомлення від клієнтів і пересилають їх по ланцюжку до поштових серверів адресатів, де ці повідомлення накопичуються. При встановленні з'єднання між адресатом і його поштовим сервером відбувається автоматична передача повідомлень, що поступили, на комп'ютер адресата.

Поштова служба заснована на двох прикладних протоколах: SMTP і POP3. По першому відбувається відправка кореспонденції з комп'ютера на сервер, а по другому - прийом повідомлень, що поступили.

Існує велика різноманітність клієнтських поштових програм. До них відноситься, наприклад, програма "Microsoft Outlook Express", що входить до складу операційної системи Windows як стандартна. Потужніша програма, інтегруюча в собі окрім підтримки електронної пошти і інші засоби діловодства, "Microsoft Outlook", входить до складу відомого пакету "Microsoft Office". Із спеціалізованих поштових програм хорошу популярність мають програми "The Bat" і "Mozilla Thunderbird".

Списки розсилки (Mail list)

Звичайна електронна пошта припускає наявність двох партнерів по листуванню. Якщо ж партнерів немає, то досить великий потік поштової інформації у свою адресу можна забезпечити, підписавшись на списки розсилки. Це спеціальні тематичні сервери, що збирають інформацію по певних темах і переправляють її передплатникам у вигляді повідомлень електронної пошти.

Темами списків розсилки може бути що завгодно, наприклад питання, пов'язані з вивченням іноземних мов, науково-технічні огляди, презентація нових програмних і апаратних засобів обчислювальної техніки.

Більшість телекомпаній створюють списки розсилки на своїх вузлах, через які розсилають клієнтам анотовані огляди телепрограм.

Служба телеконференцій (Usenet)

Служба телеконференцій схожа на циркулярну розсилку електронної пошти, в ході якої одне повідомлення вирушає не одному кореспондентові, а великій групі (такі групи називаються телеконференціями або групами новин).

Повідомлення, спрямовані на сервер групи новин, вирушають з нього на усі сервери, з якими він пов'язаний, якщо на них цього повідомлення ще немає. Далі процес повторюється.

На кожному з серверів повідомлення, що поступило, зберігається обмежений час і усі охочі можуть протягом цього часу з ним ознайомитися. Поширюючись на всі боки, менш ніж за добу повідомлення можуть охопити усю земну кулю. Далі поширення затухає, оскільки на сервер, який вже має це повідомлення, повторна передача проводитися не може.

Щодня у світі створюється близько декількох мільйонів повідомлень для груп новин. Вибрати в цьому масиві дійсно корисну інформацію практично неможливо. Тому уся система телеконференцій розбита на тематичні групи. Сьогодні у світі налічують близько 50 000 тематичних груп новин. Вони охоплюють більшість тим, що цікавлять маси

Основний прийом використання груп новин полягає в тому, щоб поставити питання, звертаючись до всього світу, і отримати відповідь або раду від тих, хто з цим питанням вже розібрався. При цьому важливо стежити за тим, щоб зміст питання відповідав темі цієї телеконференції.

Багато кваліфікованих фахівців світу регулярно переглядають повідомлення телеконференцій, що проходять в групах, що стосуються їх сфери діяльності. Такий перегляд називається моніторингом інформації. Регулярний моніторинг дозволяє фахівцям точно знати, що нового відбувається у світі по їх спеціальності, які проблеми турбують великі маси людей і на що потрібно звернути особливу увагу у своїй роботі.

У сучасних промислових і проектно-конструкторських організаціях вважається хорошим тоном, якщо фахівці вищого ешелону періодично (один-два рази в місяць) відповідають через систему телеконференцій на типові питання користувачів своєї продукції.

При відправці повідомлень в телеконференції прийнято вказувати свою адресу електронної пошти для зворотного зв'язку.

Величезний об'єм повідомлень в групах новин значно утрудняє їх цілеспрямований моніторинг, тому в деяких групах проводиться попереднє "відсівання" даремної інформації (зокрема, рекламною), що не відноситься до теми конференції. Такі конференції називають модеруємыми. Модератором може виступати не лише людина, але і програма, що фільтрує повідомлення за певними ключовими словами. У останньому випадку говорять про автоматичну модерацію.

Для роботи із службою телеконференцій існують спеціальні клієнтські програми. Так, наприклад, застосування Microsoft Outlook Express, вказане вище як поштовий клієнт, дозволяє працювати також і із службою телеконференцій. Для початку роботи потрібно настроїти програму на взаємодію з сервером груп новин, оформити "підписку" на певні групи і періодично, як і електронну пошту, отримувати усі повідомлення, що проходять по темі цієї групи. В даному випадку слово "підписка" не припускає з боку клієнта ніяких зобов'язань або платежів - це просто вказівка серверу про те, що повідомлення по вказаних темах потрібно доставляти, а по інших - ні. Відмінити підписку або змінити її склад можна в будь-який зручний момент.

Служба передачі файлів (FTP)

Необхідність в передачі файлів виникає, наприклад, при прийомі файлів програм, при пересилці великих документів (наприклад, книг), а також при передачі архівних файлів, в яких запаковані великі об'єми інформації.

Служба FTP має свої сервери у світовій мережі, на яких зберігаються архіви даних. З боку клієнта для роботи з серверами FTP може бути встановлене спеціальне програмне забезпечення, хоча в більшості випадків браузері WWW мають вбудовані можливості для роботи і по протоколу FTP.

Протокол FTP працює одночасно з двома TCP -соединениями між сервером і клієнтом. По одному з'єднанню йде передача даних, а друге з'єднання використовується як що управляє.

Протокол FTP також надає серверу засобу для ідентифікації клієнта, що звернувся. Цим часто користуються комерційні сервери і сервери обмеженого доступу, що поставляють інформацію тільки зареєстрованим клієнтам, - вони видають запит на введення імені користувача і пов'язаного з ним пароля. Проте існують і десятки тисяч FTP -серверов з анонімним доступом для усіх охочих. В цьому випадку як ім'я користувача потрібно ввести слово: anonymous, а як пароль задати адресу електронної пошти. В більшості випадків програми-клієнти FTP роблять це автоматично.

Служба Internet Relay Chat

Служба IRC (Internet Relay Chat) призначена для прямого спілкування декількох чоловік в режимі реального часу. Іноді службу IRC називають чат-конференціями або просто чатом.

На відміну від системи телеконференцій, в якій спілкування між учасниками обговорення теми відкрите всьому світу, в системі IRC спілкування відбувається тільки в межах одного каналу, в роботі якого беруть участь зазвичай лише декілька чоловік. Кожен користувач може створити власний канал і запросити в нього учасників "бесіди" або приєднатися до одного з відкритих в даний момент каналів.

Існує декілька популярних клієнтських програм для роботи з серверами і мережами, підтримувальними сервіс IRC. Одна з найбільш популярних - програма mtRC.exe.

Служба ICQ

Ця служба призначена для пошуку мережевого IP -адреса людини, підключеної в даний момент до Інтернету.

Необхідність в подібній послугі пов'язана з тим, що більшість користувачів не мають постійного IP -адреса. Назва служби є акронимом вираження "I seek you" - я тебе шукаю.

Для користування цією службою потрібно реєструватися на її центральному сервері (<http://www.icq.com>) і отримати персональний ідентифікаційний номер UIN (Universal Internet Number). Цей номер можна повідомити партнерам по контактах, і тоді служба ICQ набуває характеру Інтернет-пейджера. Знаючи номер UIN партнера, але не знаючи його поточний IP -адрес, можна через центральний сервер служби відправити йому сполучення з пропозицією встановити з'єднання.

Кожен комп'ютер, підключений до Інтернету, повинен мати чотиризначний IP -адрес. Ця адреса може бути постійною або динамічно тимчасовою. Ті комп'ютери, які включені в Інтернет на постійній основі, мають постійні IP -адреса. Більшість же користувачів підключаються до Інтернету лише на час сеансу. Їм видається динамічний IP -адрес, діючий тільки протягом цього сеансу. Ця адреса видає той сервер, через який відбувається підключення. У різних сеансах динамічний IP -адрес може бути різним, причому заздалегідь невідомо яким.

При кожному підключенні до Інтернету програма ICQ, встановлена на комп'ютері, визначає поточний IP -адрес і повідомляє його центральну службу, яка, у свою чергу, оповіщає ваших партнерів по контактах. Далі ваші партнери (якщо вони теж є клієнтами цієї служби) можуть встановити з вами прямий зв'язок. Програма надає можливість вибору режиму зв'язку ("готовий до контакту"; "прошу не турбувати, але готовий прийняти термінове повідомлення"; "закритий для контакту" і тому подібне). Після встановлення контакту зв'язок відбувається в режимі, аналогічному сервісу IRC.

IP -телефонія.

VoIP (англ. Voice - over - IP - IP -телефонія) - система зв'язку, що забезпечує передачу мовного сигналу по мережі Інтернет або по будь-кому іншим IP -мережам. Сигнал по каналу зв'язку передається в цифровому виді і, як правило, перед передачею перетворюється (стискується) з тим, щоб видалити надмірність, властиву людській мові. При розмові, голосові сигнали (слова, які вимовляються) перетворюються в стислі пакети даних. Потім, ці пакети даних посилаються через Інтернет іншій стороні. Коли пакети даних досягають адресата, вони декодуються в голосові сигнали оригіналу.

IP -телефонія забезпечує голосовий зв'язок поверх мереж, що використовують Інтернет-протокол (IP). Технологія дозволяє об'єднати безліч розосереджених об'єктів організації, включаючи мобільних працівників, в єдину конвергентну мережу.

IP -телефонія дозволяє досягти економії витрат шляхом об'єднання функцій голосового зв'язку і передачі даних в одну мережу, технічна підтримка якої може здійснюватися централізовано, а також шляхом ліквідації витрат на міжміський і міжнародний зв'язок при дзвінках на видалені об'єкти.

Принцип дії IP -телефонії - це конвертація голосового зв'язку в пакети даних. Телефонні апарати підключені до портів передачі даних IP -сети. При цьому, телефонні функції можуть з легкістю виконуватися іншим пристроєм, вже підключеним до мережі. Таку функцію може виконувати ПК.

Телефонія типу "клієнт-сервер", LAN з функціями телефонії, чиста IP -телефонія, конвергентна телефонія і LAN -телефонія - усе це терміни, що означають одну і ту ж базову розподілену архітектуру IP -телефонії. В протилежність відомому підходу, коли окремих багатоканальний телефон розміщується у ПК на кожному робочому місці, IP -телефонія використовує програмне забезпечення, що встановлюється на ПК і виконує функції "програмованого" телефону. На відміну від телефонного устаткування, програмне забезпечення можна легко модернізувати і удосконалити. При цьому не виникає необхідності переривати

робочий процес, витратити засоби на устаткування і навіть підходити до кожного робочого місця.

Концепція передачі голосу поверх мережі даних дуже вигідна для компаній, які підтримують інтернет-зв'язок з усіма відділеннями і одночасно платять за використання контурів голосового зв'язку АТС в таких відділеннях. Економія витрат сама по собі є привабливим чинником, а параметри безпеки, надійності і якості зв'язку поза сумнівом спонукатимуть менеджерів мереж зробити вибір на користь IP -телефонії.

Способи ведення розмов можна розділити на декілька типів:

Комп'ютер-телефон - При цьому способі абонентам треба мати з одного боку комп'ютер із звуковою картою і підключеними до неї динаміками або навушниками і мікрофон (також може бути використаний IP -телефон) і з іншого боку звичайнісінький телефон, підключений до телефонної лінії.

Комп'ютер-комп'ютер - При цьому способі абонентам треба мати з обох сторін комп'ютер із звуковою картою і підключеними до неї динаміками або навушниками і мікрофон (також може бути використаний IP -телефон). Це найпростіший і дешевший спосіб інтернет телефонії, оскільки витрати обох співрозмовників зводяться до вартості їх інтернет з'єднання.

Телефон-телефон - При цьому способі одному з абонентів доведеться скористатися однією з численних компаній, що надають послуги IP -телефонії а з іншого боку звичайнісінький телефон, підключений до телефонної лінії.

Найбільш широко поширене програмне забезпечення для ip -телефонії - поширювана безкоштовно програма **Skype**.

Питання:

1. Що таке інтернет?
2. У чому полягають мета і функції протоколу IP?
3. Що таке і яка структура IP -адреса?
4. Що таке DNS?
5. Що таке HTTP?
6. Що таке URI і яка його структура?
7. Що таке HTML?
8. З яких частин складається HTML -документ?
9. Для чого призначений і як використовується тег HTML?
10. Для чого призначений і як використовується тег HEAD?
11. Для чого призначений і як використовується тег BODY?
12. Для чого призначений і як використовується тег TITLE?
13. Для чого призначений і як використовується тег META?
14. Для чого призначений і як використовується тег LINK?
15. Для чого призначений і як використовується тег LINK?
16. Наведіть приклади елементів форматування тексту.
17. За допомогою яких тегів створюються списки?
18. За допомогою яких тегів створюються таблиці?
19. Яким чином в HTML -документі створюються гіпертекстові посилання?
20. Яким чином в HTML -документ можна вставити зображення?
21. Яким чином в HTML -документ можна вставити об'єкт? Що це за об'єкти?
22. Для чого призначені теги DIV і SPAN?
23. Що таке стиль?
24. Які існують способи зв'язування документу з таблицями стилів?
25. Наведіть приклади декількох властивостей стилів.
26. Що таке WYSIWYG -редактор HTML -документів?
27. Які служби мережі INTERNET окрім WWW Ви знаєте?

7. ОСНОВИ АЛГОРИТМІЗАЦІЇ І ПРОГРАМУВАННЯ

7.1. Стадії рішення завдань на комп'ютері

Рішення завдань комп'ютером включає наступні основні стадії, частина, з якої, здійснюється без участі комп'ютера.

- **Постановка задачі:**
 - збір інформації про задачі;
 - формулювання умови задачі;
 - визначення кінцевої мети рішення задачі;
 - визначення форми видачі результатів;
 - опис даних (їх типів, діапазонів величин, структури і тому подібне).
- **Аналіз і дослідження задачі, моделі:**
 - аналіз існуючих аналогів;
 - аналіз технічних і програмних засобів;
 - розробка математичної моделі;
 - розробка структур даних.
- **Розробка алгоритму:**
 - вибір методу проектування алгоритму;
 - вибір форми запису алгоритму (блок-схеми, псевдокод і ін.);
 - вибір тестів і методу тестування;
 - проектування алгоритму.
- **Програмування:**
 - вибір мови програмування;
 - уточнення способів організації даних;
 - запис алгоритму на вибраній мові програмування.
- **Тестування і відладка:**
 - синтаксична відладка;
 - відладка семантики і логічної структури;
 - тестові розрахунки і аналіз результатів тестування;
 - вдосконалення програми.
- **Аналіз результатів рішення задачі і уточнення у разі потреби математичної моделі з повторним виконанням етапів 2 — 5.**
- **Супровід програми:**
 - доопрацювання програми для вирішення конкретних завдань;
 - складання документації до вирішеного завдання, до математичної моделі, до алгоритму, до програми, до набору тестів, до використання.

Математична модель — це система математичних співвідношень — формул, рівнянь, нерівностей і так далі, що відображають істотні властивості об'єкту або явища.

Щоб описати явище, необхідно виявити найістотніші його властивості, закономірності, внутрішні зв'язки, роль окремих характеристик явища. Виділивши найбільш важливі чинники, можна нехтувати менш істотними.

Звичайно, результати обчислювального експерименту можуть опинитися і не відповідними дійсності, якщо в моделі не будуть враховані якісь важливі сторони дійсності.

Отже, створюючи математичну модель для вирішення завдання, потрібно:

- виділити припущення, на яких ґрунтуватиметься математична модель;
- визначити, що вважати за початкові дані і результати;
- записати математичні співвідношення, що пов'язують результати з початковими даними.

Відладка програми — це процес пошуку і усунення помилок в програмі, вироблюваний за наслідками її прогону на комп'ютері.

Англійський термін *debugging* ("відладка") буквально означає "вилов жучків". Термін з'явився в 1945 р., коли один з перших комп'ютерів — "Марк-1" припинив роботу через те, що в його електричні ланцюги потрапив метелик і заблокував своїми останками одна з тисяч реле машини.

У сучасних програмних системах відладка здійснюється часто з використанням спеціальних програмних засобів, званих відладчиками. Ці засоби дозволяють досліджувати внутрішню поведінку програми.

Программа-отладчик зазвичай забезпечує наступні можливості:

- покрокового виконання програми із зупинкою після кожної команди (оператора);
- проглядання поточного значення будь-якою змінною або знаходження значення будь-якого виразу, зокрема, з використанням стандартних функцій; при необхідності можна встановити нове значення змінної;
- установку в програмі "Контрольних крапок", тобто крапок, в яких програма тимчасово припиняє своє виконання, так що можна оцінити проміжні результати, і ін.

При відладці програм важливо пам'ятати наступне:

- на початку процесу відладки треба використовувати прості тестові дані;
- виникаючі утруднення слід чітко розділяти і усувати строго по черзі;
- не потрібно вважати за причину помилок машину, оскільки сучасні машини і транслятори володіють надзвичайно високою надійністю.

Тестування (англ. test — випробування) — це випробування, перевірка правильності роботи програми в цілому, або її складових частин.

Як би не була ретельно відладжена програма, вирішальним етапом, що встановлює її придатність для роботи, є контроль програми за наслідками її виконання на системі тестів. Програму умовно можна вважати за правильну, якщо її запуск для вибраної системи тестових початкових даних у всіх випадках дає правильні результати. Але, як справедливо указував відомий теоретик програмування Е. Дейкстра, тестування може показати лише наявність помилок, але не їх відсутність. Нерідкі випадки, коли нові вхідні дані викликають "відмову" або отримання невірних результатів роботи програми, яка вважалася за повністю відладжену.

Для реалізації методу тестів мають бути виготовлені або заздалегідь відомі еталонні результати.

Тестові дані повинні забезпечити перевірку всіх можливих умов виникнення помилок:

- має бути випробувана кожна гілка алгоритму;
- черговий тестовий прогін повинен контролювати щось таке, що ще не було перевірене на попередніх прогонах;
- перший тест має бути максимально простий, щоб перевірити, чи працює програма взагалі;
- арифметичні операції в тестах повинні гранично спрощуватися для зменшення об'єму обчислень;
- кількості елементів послідовностей, точність для ітераційних обчислень, кількість проходів циклу в тестових прикладах повинні задаватися з міркувань скорочення об'єму обчислень;
- мінімізація обчислень не повинна знижувати надійності контролю;
- тестування має бути цілеспрямованим і систематизованим, оскільки випадковий вибір початкових даних привів би до труднощів у визначенні ручним способом очікуваних результатів; крім того, при випадковому виборі тестових даних можуть виявитися неперевіреними багато ситуацій;
- ускладнення тестових даних повинне відбуватися поступово.

Приклад. Система тестів для завдання знаходження коріння квадратного рівняння $ax^2 + bx + c = 0$:

Випадок, що перевіряється	Коефіцієнти			Результати
	a	b	c	
$d > 0$	1	1	-2	$x_1 = 1, x_2 = -2$
$d = 0$	1	2	1	Корені рівні: $x_1 = -1, x_2 = -1$
$d < 0$	2	1	2	Дійсних коренів нема
$a=0, b=0, c=0$	0	0	0	Всі коефіцієнти рівні нулю. x — будь-яке число.
$a=0, b=0, c \neq 0$	0	0	2	Невірне рівняння
$a=0, b \neq 0$	0	2	1	Лінійне рівняння. Один корінь: $x = -0,5$
$a \neq 0, b \neq 0, c = 0$	2	1	0	$x_1 = 0, x_2 = -0,5$

Відладка і тестування — це два різних і несхожих один на одного етапи:

- при відладці відбувається локалізація і усунення синтаксичних помилок і явних помилок кодування;
- у процесі ж тестування перевіряється працездатність програми, що не містить явних помилок.

Тестування встановлює факт наявності помилок, а відладка з'ясовує її причину.

7.2. Алгоритми.

Поняття алгоритму.

Поняття алгоритму таке ж основоположне для інформатики, як і поняття інформації. Саме тому важливо в нім розібратися.

Назва "**алгоритм**" відбулася від латинської форми імені найбільшого середньоазіатського математика Мухаммеда ібн Муса ал-Хорезмі (Alharithmi), що жив в 783-850 рр. У своїй книзі "Про індійський рахунок" він виклав правила запису натуральних чисел за допомогою арабських цифр і правила дій над ними "стовпчиком", знайомі тепер кожному школяру. У XII столітті ця книга була перекладена на латинь і набула широкого поширення в Європі.

Людина щодня зустрічається з необхідністю слідувати тим або іншим правилам, виконувати різні інструкції і вказівки. Наприклад, переходячи через дорогу на перехресті без світлофора треба спочатку подивитися направо. Якщо машин немає, то перейти півдороги, а якщо машини є, чекати, поки вони проїдуть, потім перейти півдороги. Після цього подивитися наліво і, якщо машин немає, то перейти дорогу до кінця, а якщо машини є, чекати, поки вони проїдуть, а потім перейти дорогу до кінця.

У математиці для вирішення типових завдань ми використовуємо певні правила, що описують послідовності дій. Наприклад, правила складання дробових чисел, вирішення квадратних рівнянь і так далі. Зазвичай будь-які інструкції і правила є послідовністю дій, які необхідно виконати в певному порядку. Для вирішення завдання треба знати, що дане, що слід отримати і які дії і в якому порядку слід для цього виконати. Розпорядження, що визначає порядок виконання дій над даними з метою отримання шуканих результатів, і є алгоритм.

Алгоритм — заздалегідь задане зрозуміле і точне розпорядження можливому виконавцеві зробити певну послідовність дій для отримання рішення задачі за кінцеве число кроків.

Це — не визначення в математичному сенсі слова, а, швидше, опис інтуїтивного поняття алгоритму, що розкриває його суть.

Порядок виконання алгоритму:

- Дії в алгоритмі виконуються в порядку їх запису
- Не можна міняти місцями ніякі дві дії алгоритму
- Не можна не закінчивши однієї дії переходити до наступної

У міру розвитку паралельності в роботі комп'ютерів слово «послідовність» почали замінювати більш загальним словом «порядок». Це пов'язано з тим, що якісь дії алгоритму мають бути виконані тільки один за одним, але якісь можуть бути і незалежними. Поняття алгоритму необов'язково відноситься до комп'ютерних програм, так, наприклад, чітко описаний рецепт приготування блюда також є алгоритмом, у такому разі виконавцем є людина. Проте найчастіше як виконавець виступає комп'ютер.

Існує багато тлумачень алгоритму. Наприклад:

«Алгоритм — це всяка система обчислень, що виконуються по строго заданим правилам, яка після якого-небудь числа кроків свідомо приводить до рішення поставленої задачі.» (А. Колмогоров)

«Алгоритм — це точне розпорядження, що визначає обчислювальний процес, що йде від варіюваних початкових даних до шуканого результату.» (А. Марков)

«Алгоритм — строго детермінована послідовність дій, що описує процес перетворення об'єкту з початкового стану в кінцевий, записана за допомогою зрозумілих виконавцві команд.» (Угріновіч Микола Дмитрович)

«Алгоритм — це послідовність дій, направлених на отримання певного результату за кінцеве число кроків.» (Roxanstudio)

«Алгоритм є формалізована послідовність дій (подій). Алгоритм може бути записаний словами і зображений схематично. Практично будь-яка не випадкова повторювана дія піддається опису через алгоритм.» ([grey_olli])

«Алгоритм — однозначно, доступно і коротко (умовні поняття — назви етапу) описана послідовність процедур для відтворення процесу з обумовленим завданням алгоритму результатом за заданих початкових умов. Універсальність (або спеціалізація) алгоритму визначається застосовністю і надійністю даного алгоритму для вирішення нестандартних завдань.»

«Алгоритм — це система операторів, узятих з безлічі операторів деякого виконавця, яка повністю визначає деякий клас алгоритмічних процесів, тобто процесів, які: дискретні; детерміновані; потенційно кінцеві; перетворюють деякі конструктивні об'єкти.

Виконавець алгоритму

Виконавець алгоритму — це деяка абстрактна або реальна (технічна, біологічна або біотехнічна) система, здатна виконати дії, що наказують алгоритмом.

Виконавця характеризують:

- середовище;
- елементарні дії;
- система команд;
- відмови.

Середовище — це "житло" виконавця. Наприклад, для виконавця Робот зі шкільного підручника середовище — це нескінченне клітинне поле. Стіни і закрашені клітки теж частина середовища. А їх розташування і положення самого Робота задають конкретний стан середовища.

Система команд. Кожен виконавець може виконувати команди тільки з деякого строго заданого списку — системи команд виконавця. Для кожної команди мають бути задані умови застосовності (у яких станах середовища може бути виконана команда) і описані результати виконання команди. Наприклад, команда Робота "вгору" може бути виконана, якщо вище за Робота немає стіни. Її результат — зсув Робота на одну клітку вгору.

Після виклику команди виконавець здійснює відповідну *елементарну дію*.

Відмови виконавця виникають, якщо команда викликається при неприпустимому для неї стані середовища.

Властивості алгоритмів

Основні властивості алгоритмів наступні:

1. **Зрозумілість** для виконавця — виконавець алгоритму повинен розуміти, як його виконувати. Іншими словами, маючи алгоритм і довільний варіант початкових даних, виконавець повинен знати, як треба діяти для виконання цього алгоритму.

2. **Дискретність** (переривчатість, нарізність) — алгоритм повинен представляти процес рішення задачі як послідовне виконання простих (або раніше визначених) кроків (етапів).

3. **Визначеність** — кожне правило алгоритму має бути чітким, однозначним і не залишати місця для свавілля. Завдяки цій властивості виконання алгоритму носить механічний характер і не вимагає ніяких додаткових вказівок або відомостей про вирішуване завдання.

4. **Результативність** (або кінцевість) полягає в тому, що за кінцеве число кроків алгоритм або повинен приводити до рішення задачі, або після кінцевого числа кроків зупинитися із-за неможливості отримати рішення з видачею відповідного повідомлення, або необмежено продовжуватися протягом часу, відведеного для виконання алгоритму, з видачею проміжних результатів.

5. **Масовість** означає, що алгоритм рішення задачі розробляється в загальному вигляді, тобто він має бути застосовний для деякого класу завдань, що розрізняються лише початковими даними. При цьому початкові дані можуть вибиратися з деякої області, яка називається областю застосовності алгоритму.

Способи запису алгоритмів.

На практиці найбільш поширені наступні форми представлення алгоритмів:

- **словесна** (запис на природній мові);
- **графічна** (зображення з графічних символів);
- **псевдокоди** (напівформалізовані описи алгоритмів на умовній алгоритмічній мові, що включають як елементи мови програмування, так і фрази природної мови, загальноприйняті математичні позначення і ін.);
- **програмна** (тексти на мовах програмування).

Словесним способом запису алгоритмів є опис послідовних етапів обробки даних. Алгоритм задається в довільному викладі на природній мові.

Наприклад. Записати алгоритм знаходження найбільшого загального дільника (НОД) двох натуральних чисел (алгоритм Евкліда).

Алгоритм може бути наступним:

- задати два числа;
- якщо числа рівні, то взяти будь-яке з них за відповідь і зупинитися, інакше продовжити виконання алгоритму;
- визначити більше з чисел;
- замінити більше з чисел різницею більшого і меншого з чисел;
- повторити алгоритм з кроку 2.

Описаний алгоритм застосовний до будь-яких натуральних чисел і повинен приводити до рішення поставленої задачі. Переконайтеся в цьому самостійно, визначивши за допомогою цього алгоритму найбільшого загального дільника чисел 125 і 75.

Словесний спосіб не має широкого розповсюдження, оскільки такі описи:

- строго не формалізуються;

- страждають багатослівністю записів;
- допускають неоднозначність тлумачення окремих розпоряджень.

Графічний спосіб представлення алгоритмів є компактнішим і наочнішим в порівнянні із словесним. Алгоритм зображається у вигляді послідовності зв'язаних між собою функціональних блоків, кожен з яких відповідає виконанню одного або декількох дій.

Таке графічне уявлення називається схемою алгоритму або блок-схемою. У блок-схемі кожному типу дій (введенню початкових даних, обчисленню значень виразів, перевірці умов, управлінню повторенням дій, закінченню обробки і тому подібне) відповідає геометрична фігура, представлена у вигляді блокового символу. Блокові символи з'єднуються лініями переходів, що визначають черговість виконання дій. У таблиці приведені символи, що найбільш часто вживаються.

Назва символу	Позначення і приклад заповнення	Пояснення
Процес (блок обчислень)		Обчислювальна дія або послідовність дій
Рішення (логічний блок)		Перевірка умов
Модифікація		Початок циклу
Зумовлений процес		Обчислення по підпрограмі, стандартна підпрограма
Введення-виведення		Уведення-виведення в загальному вигляді
Пуск-зупинка		Початок, кінець алгоритму, вхід і вихід в підпрограму
Документ		Виведення результатів на друк

Блок **"процес"** застосовується для позначення дії або послідовності дій, що змінюють значення, форму уявлення або розміщення даних. Для поліпшення наочності схеми декілька окремих блоків обробки можна об'єднувати в один блок. Представлення окремих операцій достатнє вільно.

Блок **"рішення"** використовується для позначення переходів управління по умові. У кожному блоці "рішення" мають бути вказані питання, умова або порівняння, які він визначає.

Блок **"модифікація"** використовується для організації циклічних конструкцій. (Слово модифікація означає видозміну, перетворення). У середині блоку записується параметр циклу, для якого указуються його початкове значення, гранична умова і крок зміни значення параметра для кожного повторення.

Блок **"зумовлений процес"** використовується для вказівки звернень до допоміжних алгоритмів, що існують автономно у вигляді деяких самостійних модулів, і для звернень до бібліотечних підпрограм.

Псевдокод є системою позначень і правил, призначеною для одноманітного запису алгоритмів.

Псевдокод займає проміжне місце між природною і формальною мовами. З одного боку, він близький до звичайної природної мови, тому алгоритми можуть на ній записуватися і читатися як звичайний текст. З іншого боку, в псевдокоді використовуються деякі формальні конструкції і математична символіка, що наближає запис алгоритму до загальноприйнятого математичного запису.

У псевдокодi не прийняті строгі синтаксичні правила для запису команд, властиві формальним мовам, що полегшує запис алгоритму на стадії його проектування і дає можливість використовувати ширший набір команд, розрахований на абстрактного виконавця.

Проте в псевдокодi зазвичай є деякі конструкції, властиві формальним мовам, що полегшує перехід від запису на псевдокодi до запису алгоритму на формальній мові. Зокрема, в псевдокодi, так само, як і у формальних мовах, є **службові слова**, сенс яких визначений раз і назавжди. Вони виділяються в друкарському тексті жирним шрифтом, а в рукописному тексті підкреслюються.

Єдиного або формального визначення псевдокоду не існує, тому можливі різні псевдокоди, що відрізняються набором службових слів і основних (базових) конструкцій.

Прикладом псевдокоду є шкільна алгоритмічна мова в російській нотації (шкільна АМ), описаний в підручнику А.Г. Кушніренко та ін. "Основи інформатики і обчислювальної техніки", 1991. Цю мову надалі ми називатимемо просто "алгоритмічна мова".

Основні службові слова

алг (алгоритм)	сим (символьний)	дано	для	так
арг (аргумент)	літ (літерний)	необхідно	від	ні
рез (результат)	лог (логічний)	якщо	до	при
поч (початок)	таб (таблиця)	то	знач	вибір
кін (кінець)	пц (початокциклу)	інакше	і	введення
ціл (цілий)	кц (кінецьциклу)	всі	або	виведення
дійс (дійсний)	довж (довжина)	поки	не	утв

Загальний вид алгоритму:

алг назва алгоритму (аргументи та результат)
дано умови застосування алгоритму
необхідно ціль виконання алгоритму
поч опис проміжних величин
 | послідовність команд (тіло алгоритму)
кін

Частина алгоритму від слова **алг** до слова **нач** називається заголовком, а частина, увязнена між словами **нач** і **кон** — **тілом** алгоритму.

В реченні **алг** після назви алгоритму в круглих скобках вказуються **характеристики** (**арг**, **рез**) і **тип значення** (**ціл**, **дійсн**, **сим**, **літ** або **лог**) всіх **вхідних** (аргументи) та **вихідних** (результати) змінних. При описі масивів (таблиць) використовується службове слово **таб**, доповнене **граничними парами** по кожному індексу елементів масиву.

Приклади речень алг:

алг Об'єм та площа циліндру (**арг** **дійсн** R, H, **рез** **дійсн** V, S)
алг Корені КвРів (**арг** **дійсн** a, b, c, **рез** **дійсн** x1, x2, **рез** **літ** t)
алг Виключити елемент (**арг** **ціл** N, **арг** **рез** **дійсн** **таб** A[1:N])
алг Діагональ (**арг** **ціл** N, **арг** **ціл** **таб** A[1:N, 1:N], **рез** **літ** Відповідь)

Речення **дано** і **необхідно** не обов'язкові. У них рекомендується записувати твердження, що описують стан середовища виконавця алгоритму, наприклад:

- **алг** Заміна (**арг** **літ** Str1, Str2, **арг** **рез** **літ** Text)
дано | довжина підстроки Str1 і Str2 співпадають
необхідно | повсюди в строці Text підстроку Str1 замінити на Str2
- **алг** Число максимумів (**арг** **ціл** N, **арг** **дійс** **таб** A[1:N], **рез** **ціл** K)
дано | N>0
необхідно | K — число максимальних елементів в таблиці A
- **алг** Опір (**арг** **дійс** R1, R2, **арг** **ціл** N, **рез** **дійс** R)

дано | $N > 5, R1 > 0, R2 > 0$

необхідно | R — Опір схеми

Тут в реченнях **дано** і **необхідно** після знаку "|" записані коментарі. Коментарі можна поміщати в кінці будь-якого рядка. Вони не обробляються транслятором, але істотно полегшують розуміння алгоритму.

Команди шкільного АМ

Команда **привласнення**. Служить для обчислення виразів і привласнення їх значень змінним. Загальний вигляд: $A := U$, де знак " $:=$ " означає команду замінити колишнє значення змінної, що стоїть в лівій частині, на обчислене значення виразу, що стоїть в правій частині.

Приклад, $a := (b+c) * \sin(\pi/4); i := i+1$.

Команди **введення** і **виводу**.

- **введення** імена змінних
- **вивід** імена змінних, вирази, тексти.

Команди **якщо** і **вибір**. Застосовують для організації розгалужень.

Команди **для** і **поки**. Застосовують для організації циклів.

Приклад запису алгоритму на шкільній АМ

алг Сума квадратів (**арг** ціл n, **рез** ціл S)

дано | $n > 0$

необхідно | $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

нач ціл i

введення n; $S := 0$

нц для i **от** 1 **до** n

$S := S + i*i$

кц

вивод "S = ", S

кін

Базові структури алгоритмів.

Логічна структура будь-якого алгоритму може бути представлена комбінацією трьох основних алгоритмічних структур (ОАС): **слідування**, **розгалуження**, **цикл**.

У складніших випадках використовуються суперпозиції (вкладення) ОАС.

Нижче приведені графічні позначення (позначення на блок-схемах) ОАС.

Структура "слідування"	Повнарозвилка	Неповнарозвилка
Цикл с передумовою(цикл ПОКИ)	Цикл с постумовою(цикл ДО)	Цикл с параметром

На схемах СЕРІЯ позначає один або декілька будь-яких операторів; УМОВА є логічний вираз (ЛВ) (якщо його значення ІСТИНА, перехід відбувається по гілці ТАК, інакше — по НІ). На схемі циклу з параметром використані позначення: ПЦ — параметр циклу, ПЗ — початкове значення параметра циклу, КЗ — кінцеве значення параметра циклу, К — крок зміни параметра циклу.

Початок і кінець алгоритму на блок-схемах позначають овалом, змінні, що вводяться і виводяться, записуються в паралелограми.

У прикладах ми використовуватимемо запис алгоритмів за допомогою блок-схем і словесний опис.

Лінійні алгоритми

Прості завдання мають лінійний алгоритм рішення. Це означає, що він не містить перевірок умов і повторень.

Приклад 1. Пішохід йшов по перетнутій місцевості. Його швидкість руху по рівнині v_1 км/ч, в гору — v_2 км/ч і під гору — v_3 км/ч. Час руху відповідно t_1 , t_2 і t_3 ч. Який шлях пройшов пішохід?

	<ol style="list-style-type: none"> 1. Ввести $v_1, v_2, v_3, t_1, t_2, t_3$. 2. $S_1 := v_1 * t_1$. 3. $S_2 := v_2 * t_2$. 4. $S_3 := v_3 * t_3$. 5. $S := S_1 + S_2 + S_3$. 6. Вивести значення S. 7. Кінець.
--	---

Для перевірки працездатності алгоритму необхідно задати значення вхідних змінних, обчислити кінцевий результат по алгоритму і порівняти з результатом ручного рахунку.

Приклад 2. Дано натуральне тризначне число n , в записі якого немає нулів. Скласти алгоритм, який повертає значення ІСТИНА, якщо вірно твердження: "число n кратне кожній своїй цифрі", і БРЕХНЯ — інакше.

	<ol style="list-style-type: none"> 1. Ввести число n 2. $A := n \bmod 10$ {разряд одиниць} 3. $B := n \div 100$ {разряд сотень} 4. $C := n \div 10 \bmod 10$ {десятки} 5. $L := (n \bmod A = 0) \text{ and } (n \bmod B = 0) \text{ and } (n \bmod C = 0)$ 6. Виведення L 7. Кінець
--	--

На приведеній вище схемі DIV і MOD відповідно операції ділення без остачі і отримання залишку від цілочисельного ділення. У фігурних дужках записані пояснення (коментарі) до операторів.

Розвилка

Досить часто та або інша дія має бути виконана залежно від значення логічного виразу, що виступає як умова. У таких випадках використовується розвилка.

Приклад 1. Обчислити значення функції

	1. Ввести x . 2. Якщо $x \leq 12$, $y := x^2$ 3. Якщо $x < 0$, то $y := x^4$ 4. $y := x - 2$ 5. Вивести y 6. Кінець
--	--

При тестуванні алгоритмів з розвилкою необхідно підбирати такі початкові дані, щоб можна було перевірити всі гілки. У наведеному вище прикладі повинні бути принаймні три тестові набори.

Приклад 2. Дано натуральне число n . Якщо число непарне і його подвоєння не приведе до виходу за 32767 (двобайтове ціле число із знаком), подвоїти його, інакше — залишити без зміни.

Щоб задовольнити умові подвоєння, число n має бути непарним і менше 16384.

	1. Ввести число n 2. Якщо число n непарне і менше 16384, то $n := n * 2$ 3. Виведення n 4. Кінець
--	--

Розглянутий приклад ілюструє неповну розвилку. Також слід зазначити, тут логічний вираз, що є умовою, містить 2 операнди.

Цикли

Якщо які-небудь оператори необхідно виконати кілька разів, то їх не переписують кожного разу наново, а організовують цикл.

Приклад 1. Підрахувати кількість непарних цифр в записі натурального числа n .

Ідея рішення. Із заданого числа вибирати з молодшого розряду цифру за цифрою до тих пір, поки воно не вичерпається, тобто стане рівним нулю. Кожну непарну цифру враховувати.

1. Ввести число n 2. $K := 0$ {підготувати лічильник} 3. Якщо $n = 0$, переход к п. 7 4. Якщо $n \bmod 10 \bmod 2 = 1$, то $K := K + 1$ 5. $n := n \div 10$ 6. Перехід к п. 3 7. Вивести K 8. Кінець

Завдання вирішене двома способами. Зліва рішення оформлене з використанням циклу з передумовою, справа — з постумовою.

Приклад 2. Дана послідовність, загальний член якої визначається формулою

Обчислити при $n > 2$ суму тих її членів, які більше заданого числа ϵ .

При рішенні задачі знаходиться черговий член послідовно i , якщо він більше ϵ , додається до суми.

	1. Ввести ϵ 2. $S := 0$ 3. $A := 1/4$ 4. $n := 3$ 5. Порівняти A з ϵ . Якщо $A \geq \epsilon$, перейти доп. 10 6. $S := S + A$ 7. $A := (n-1)/(n*n)$ 8. $n := n + 1$ 9. Перехід доп. 5 10. Виведення S 11. Кінець
--	---

У розглянутих вище прикладах кількість повторень заздалегідь невідома. У першому воно залежить від кількості цифр в записі натурального числа, в другому — від числа ϵ .

У тих же випадках, коли кількість кроків відома з умови завдання, простіше і вигідніше використовувати *цикл з параметром*.

Приклад 3. Знайти добуток перших k натуральних чисел, кратних трьом.

При складанні алгоритму врахуємо, що перше натуральне число, кратне 3, є трійка, а всі подальші більші попереднього на 3.

	1. Введення k 2. $P := 1$ { тут накопичуємо добуток } 3. $T := 0$ { тут будуть числа, кратні 3 } 4. $I := 1$ 5. Якщо $I > k$, перейти до п. 10 6. $T := T + 3$ 7. $P := P * T$ 8. $I := I + 1$
--	--

	9. Перейти до п. 5
	10. Виведення Р
	11. Кінець

Інші приклади будуть записані вже на МПВР. У даній же публікації зроблена спроба продемонструвати, що вивчення програмування розумно починати власне з розробки алгоритмів, не акцентуючи спочатку уваги на записі алгоритму на тій або іншій мові програмування. В той же час автор, будучи прихильником структурного підходу до програмування, пропонує дотримуватися цього підходу і при програмуванні на рівні блок-схем.

Вкладені цикли

Можливі випадки, коли усередині тіла циклу необхідно повторювати деяку послідовність операторів, тобто організувати внутрішній цикл. Така структура отримала назву циклу в циклі або вкладених циклів. Глибина вкладення циклів (тобто кількість вкладених один в одного циклів) може бути різною.

При використанні такої структури для економії машинного часу необхідно виносити з внутрішнього циклу в зовнішній всі операторів, які не залежать від параметра внутрішнього циклу.

Приклад вкладених циклів «для»

Обчислити суму елементів заданої матриці $A(5,3)$.

Матриця А		S:= 0; пцдляі от 1 до 5 пцдляj от 1 до 3 S:=S+A[i,j] кц кц
-----------	--	---

Приклад вкладених циклів «поки»

Обчислити добуток тих елементів заданої матриці $A(10,10)$, які розташовані на перетині парних рядків і парних стовпців.

	i:=2; P:=1 пцпокаі <= 10 j:=2 пцпокаj <= 10 P:=P*A[i,j] j:=j+2 кц i:=i+2 кц
--	--

Програмний спосіб запису алгоритмів

При записі алгоритму в словесній формі, у вигляді блок-схеми або на псевдокоді допускається певне свавілля при зображенні команд. Разом з тим такий запис точний настільки, що дозволяє людині зрозуміти суть справи і виконати алгоритм.

Проте на практиці як виконавці алгоритмів використовуються спеціальні автомати — комп'ютери. Тому алгоритм, призначений для виконання на комп'ютері, має бути записаний на зрозумілій йому мові. І тут на перший план висувається необхідність точного запису команд, що не залишає місця для довільного тлумачення їх виконавцем.

Отже, мова для запису алгоритмів має бути формалізований. Таку мову прийнято називати мовою програмування, а запис алгоритму на цій мові — програмою для комп'ютера.

Рівень мови програмування

В даний час в світі існує декілька сотень реально використовуваних мов програмування. Для кожного є своя сфера застосування.

Будь-який алгоритм, як ми знаємо, є послідовність розпоряджень, виконавши які можна за кінцеве число кроків перейти від початкових даних до результату. Залежно від ступеня деталізації розпоряджень зазвичай визначається рівень мови програмування — чим менше деталізація, тим вище рівень мови.

По цьому критерію можна виділити наступні рівні мов програмування:

- машинні;
- машинно-орієнтовані (асемблери) ;
- машинно-незалежні (мови високого рівня).

Машинні мови і машинно-орієнтовані мови — це мови низького рівня, що вимагають вказівки дрібних деталей процесу обробки даних. Мови ж високого рівня імітують природні мови, використовуючи деякі слова розмовної мови і загальноприйняті математичні символи. Ці мови зручніші для людини.

Мови високого рівня діляться на:

•процедурні (алгоритмічні) (Basic, Pascal, C і ін.), які призначені для однозначного опису алгоритмів; для вирішення завдання процедурні мови вимагають в тій або іншій формі явно записати процедуру її рішення;

•логічні (Prolog, Lisp і ін.), які орієнтовані не на розробку алгоритму рішення задачі, а на систематичний і формалізований опис завдання з тим, щоб рішення виходило з складеного опису;

•об'єктно-орієнтовані (Object Pascal, C++, Java і ін.), в основі яких лежить поняття об'єкту, що поєднує в собі дані і дії над ними. Програма на об'єктно-орієнтованій мові, вирішуючи деяку задачу, по суті описує частину світу, що відноситься до цього завдання. Опис дійсності у формі системи взаємодіючих об'єктів природніший, ніж у формі взаємодіючих процедур.

Достоїнства і недоліки машинних мов

Кожен комп'ютер має свою машинну мову, тобто свою сукупність машинних команд, яка відрізняється кількістю адрес в команді, призначенням інформації, що задається в адресах, набором операцій, які може виконати машина і ін.

При програмуванні на машинній мові програміст може тримати під своїм контролем кожен елемент пам'яті, використовувати всі можливості наявних машинних операцій.

Але процес написання програми на машинній мові дуже трудомісткий і утомливий. Програма виходить громіздкою, важкооглядовою, її важко відлагоджувати, змінювати і розвивати.

Тому у разі, коли потрібно мати ефективну програму, яка в максимальному ступені враховує специфіку конкретного комп'ютера, замість машинних мов використовують близькі до них машино-орієнтовані мови (асемблери).

Мова асемблера

Мова асемблера — це машино-залежна мова низького рівня, в якому короткі мнемонічні імена відповідають окремим машинним командам. Використовується для представлення в легкій для читання формі програм, записаних в машинному коді.

Мова асемблера дозволяє програмістові користуватися текстовими мнемонічними (тобто що легко запам'ятовуються людиною) кодами, на свій розсуд привласнювати символічні імена регістрам комп'ютера і пам'яті, а також задавати зручні для себе способи адресації. Крім того, він дозволяє використовувати різні системи числення (наприклад, десяткову або шістнадцатерічну) для уявлення числових констант, використовувати в програмі коментарі і ін.

Програми, написані на мові асемблера, вимагають значно меншого об'єму пам'яті і часу виконання. Знання програмістом мови асемблера і машинного коду дає йому розуміння архітектури машини. Не дивлячись на те, що більшість фахівців в області програмного забезпечення розробляють програми на мовах високого рівня, таких, як Object Pascal або C, найбільш могутнє і ефективне програмне забезпечення повністю або частково написане на мові асемблера.

Мови високого рівня були розроблені для того, щоб звільнити програміста від обліку технічних особливостей конкретних комп'ютерів, їх архітектури. В протилежність цьому, мова асемблера розроблена з метою врахувати конкретну специфіку процесора. Отже, для того, щоб написати програму на мові асемблера для конкретного комп'ютера, важливо знати його архітектуру.

Як приклад приведемо програму на мові асемблера для IBM PC. Програма обчислює значення $a = b + 3$ для цілих a, b і z :

.MODEL SMALL			Директива .MODEL задає механізм розподілу		
.DATA			пам'яті під дані і команди.		
b	DW	5	Директива .DATA визначає початок ділянки		
c	DW	3	програми з даними.		
a	DW	?	Директиви DW задають типи змінних і їх значення.		
.CODE			Директива .CODE визначає початок ділянки		
begin	MOV	AX,@DATA	програми з командами.		
	MOV	DS,AX	Команди MOV Ax,@data і MOV Ds,ax		
	MOV	AX,B	записують адресу сегменту даних в регістр		
	ADD	AX,C	DS (Data Segment).		
	MOV	A,AX	Для обчислення а використовуються команди MOV		
	MOV	AH,4CH	AX, B, ADD AX, C і MOV A, AX.		
	INT	21H	У директиві END задана мітка першої виконуваної		
END	begin		програми програми begin.		

Переклад програми з мови асемблера на машинну мову здійснюється спеціальною програмою, яка називається асемблером і є, по суті, простим транслятором.

Переваги алгоритмічних мов перед машинними

Основні переваги такі:

- алфавіт алгоритмічної мови значно ширший за алфавіт машинної мови, що істотно підвищує наочність тексту програми;
- набір операцій, допустимих для використання, не залежить від набору машинних операцій, а вибирається з міркувань зручності формулювання алгоритмів вирішення завдань певного класу;
- формат пропозицій достатньо гнучкий і зручний для використання, що дозволяє за допомогою однієї пропозиції задати достатньо змістовний етап обробки даних;
- необхідні операції задаються за допомогою загальноприйнятих математичних позначень;

- даним в алгоритмічних мовах привласнюються індивідуальні імена, вибрані програмістом;

- у мові може бути передбачений значно ширший набір типів даних в порівнянні з набором машинних типів даних.

Таким чином, алгоритмічні мови значною мірою є машинно-незалежними. Вони полегшують роботу програміста і підвищують надійність створюваних програм.

Компоненти алгоритмічної мови

Алгоритмічну мову (як і будь-яку іншу мову) утворюють три його складові: алфавіт, синтаксис і семантика.

Алфавіт — це фіксований для даної мови набір основних символів, тобто "букв алфавіту", з яких повинен складатися будь-який текст на цій мові, — ніякі інші символи в тексті не допускаються.

Синтаксис — це правила побудови фраз, що дозволяють визначити, правильно або неправильно написана та або інша фраза. Точніше кажучи, синтаксисом мови є набір правил, що встановлюють, які комбінації символів є осмисленими пропозиціями на цій мові.

Семантика визначає смислове значення пропозицій мови. Будучи системою правил тлумачення окремих мовних конструкцій, семантика встановлює, які послідовності дій описуються тими або іншими фразами мови і, зрештою, який алгоритм визначений даним текстом на алгоритмічній мові.

Поняття використовувані алгоритмічними мовами

Кожне поняття алгоритмічної мови має на увазі деяку синтаксичну одиницю (конструкцію) і визначувані нею властивості програмних об'єктів або процесу обробки даних.

Поняття мови визначається у взаємодії синтаксичних і семантичних правил. Синтаксичні правила показують, як утворюється дане поняття з інших понять і букв алфавіту, а семантичні правила визначають властивості даного поняття

Основними поняттями в алгоритмічних мовах зазвичай є наступні.

1. **Імена (ідентифікатори)** — уживаються для позначення об'єктів програми (змінних, масивів, функцій і ін.).

2. **Операції.** Типи операцій:

- арифметичні операції $+$, $-$, $*$, $/$ і ін. ;
- логічні операції \wedge , або \vee , не \neg ;
- операції відношення $<$, $>$, \leq , \geq , $=$, \neq ;
- операція зчеплення (інакше, "приєднання", "конкатенації") символічних значень один з одним з утворенням одного довгого рядка; зображається знаком "+".

3. **Дані** — величини, що обробляються програмою. Є три основні види даних: константи, змінні і масиви.

- **Константи** — це дані, які зафіксовані в тексті програми і не змінюються в процесі її виконання.

Приклади констант:

- числові 7.5 , 12 ;
- логічні так (істина), немає (брехня);
- символічні (містять рівно один символ) "A" , "+" ;
- літерні (містять довільну кількість символів) "a0", "Мир", "" (порожній рядок).

- **Змінні** позначаються іменами і можуть змінювати свої значення в ході виконання програми. Змінні бувають цілі, речові, логічні, символічні і літерні.

- **Масиви** — послідовності однотипних елементів, число яких фіксоване і яким привласнено одне ім'я. Положення елементу в масиві однозначно визначається його індексами (одним, у разі одновимірного масиву, або декількома, якщо масив багатовимірний). Іноді масиви називають таблицями.

4. **Вирази** — призначаються для виконання необхідних обчислень, складаються з констант, змінних, показників функцій (наприклад, $\exp(x)$), об'єднаних знаками операцій.

Вирази записуються у вигляді лінійних послідовностей символів (без підрядкових і нарядкових символів, "багатоповерхових" дробів і так далі), що дозволяє вводити їх в комп'ютер, послідовно натискаючи на відповідні клавіші клавіатури.

Розрізняють вирази арифметичні, логічні і рядкові.

- Арифметичні вирази служать для визначення одного числового значення. Наприклад, $(1 + \sin(x)) / 2$. Значення цього виразу при $x=0$ дорівнює 0.5, а при $x=\pi/2$ — одиниці.

- Логічні вирази описують деякі умови, які можуть задовольнятися або не задовольнятися. Таким чином, логічний вираз може набувати тільки два значення — "істину" або "брехню" (та чи ні). Розглянемо як приклад логічний вираз $x*x + y*y < r*r$, що визначає приналежність точки з координатами (x, y) внутрішньої області круга радіусом r з центром на початку координат. При $x=1, y=1, r=2$ значення цього виразу — "істина", а при $x=2, y=2, r=1$ — "брехня".

- Рядкові (літерні) вирази, значеннями яких є тексти. У рядкові вирази можуть входити літерні і рядкові константи, літерні і рядкові змінні, літерні функції, розділені знаками операції зчеплення. Наприклад, $A + U$ означає приєднання рядка B до кінця рядка A . Якщо $A = \text{"кущ"}$, а $B = \text{"зелений"}$, то значення виразу $A + U$ є "кущ зелений".

5. **Оператори** (команди). Оператор — це найбільш крупне і змістовне поняття мови: кожен оператор є закінченою фразою мови і визначає деякий цілком закінчений етап обробки даних. До складу операторів входять:

- ключові слова;
- дані;
- вирази і так далі

Оператори підрозділяються на виконуваних і невиконуваних. Невиконувані оператори призначені для опису даних і структури програми, а виконувані — для виконання різних дій (наприклад, оператор привласнення, оператори введення і виводу, умовний оператор, оператори циклу, оператор процедури і ін.).

Стандартні функції

При вирішенні різних завдань за допомогою комп'ютера буває необхідно обчислити логарифм або модуль числа, синус кута і так далі

Обчислення функцій, що часто вживаються, здійснюються за допомогою підпрограм, званих стандартними функціями, які заздалегідь запрограмовані і вбудовані в транслятор мови.

Таблиця стандартних функцій шкільної алгоритмічної мови

Назва і математичне позначення функції		Покажи к функції
Абсолютна величина (модуль)	$ x $	<code>abs(x)</code>
Корінь квадратний		<code>sqrt(x)</code>
Натуральний логарифм	$\ln x$	<code>ln(x)</code>
Десятковий логарифм	$\lg x$	<code>lg(x)</code>
Експонента (ступінь числа $e \sim 2.72$)	e^x	<code>exp(x)</code>
Знак числа x (- 1, якщо $x < 0$; 0, якщо $x = 0$; 1, якщо $x > 0$)	$\text{sign } x$	<code>sign(x)</code>
Ціла частина числа x (тобто максимальне ціле число, не більше x)		<code>int(x)</code>
Мінімум з чисел x та y		<code>min(x,y)</code>
Максимум з чисел x та y		<code>max(x,y)</code>
Частное відділення цілох на цілеу		<code>div(x,y)</code>
Залишок від ділення цілох на цілеу		<code>mod(x,y)</code>
Випадкове число в діапазоні від 0 до $x - 1$		<code>rnd(x)</code>
Синус (кут в радіанах)	$\sin x$	<code>sin(x)</code>
Косинус (кут в радіанах)	$\cos x$	<code>cos(x)</code>
Тангенс (кут в радіанах)	$\text{tg } x$	<code>tg(x)</code>
Котангенс (кут в радіанах)	$\text{ctg } x$	<code>ctg(x)</code>
Арксинус (головне значення в радіанах)	$\arcsin x$	<code>arcsin(x)</code>
Арккосинус (головне значення в радіанах)	$\arccos x$	<code>arccos(x)</code>
Арктангенс (головне значення в радіанах)	$\text{arctg } x$	<code>arctg(x)</code>
Арккотангенс (головне значення в радіанах)	$\text{arcctg } x$	<code>arcctg(x)</code>

Як аргументи функцій можна використовувати константи, змінні і вирази. Наприклад:

<code>sin (3.05)</code>	<code>sin (x)</code>	<code>sin (2 * y + t / 2)</code>	<code>sin((exp(x) + 1) ** 2)</code>
<code>min (a, 5)</code>	<code>min (a, b)</code>	<code>min (a + b , a * b)</code>	<code>min(min(a, b), min(c, d))</code>

Кожна мова програмування має свій набір стандартних функцій.

Арифметичні вирази

Арифметичні вирази записуються по наступних правилах:

- Не можна опускати знак множення між співмножниками і ставити поряд два знаки операцій.
- Індеси елементів масивів записуються в квадратних (шкільний АЯ, Pascal, 3) або круглих (Basic) дужках.
- Для позначення змінних використовуються букви латинського алфавіту.

- Операції виконуються в порядку старшинства: спочатку обчислення функцій, потім піднесення до ступеня, потім множення і ділення і в останню чергу — складання і віднімання.

- Операції одного старшинства виконуються зліва направо. Проте, в шкільній АМ є одне виключення з цього правила: операції піднесення до ступеня виконуються зправа наліво. Так, вираз $2^{(3^2)}$ в шкільній АМ обчислюється як $2^{(3^2)} = 512$. У мові Qbasic аналогічний вираз 2^3^2 обчислюється як $(2^3)^2 = 64$. А в мові Pascal взагалі не передбачена операція піднесення до ступеня, в Pascal x^y записується як $\exp(y \cdot \ln(x))$, а x^y^z як $\exp(\exp(z \cdot \ln(y)) \cdot \ln(x))$.

Приклади запису арифметичних виразів

Математичний запис	Запис на шкільній алгоритмічній мові
	$x * y / z$
	$x / (y * z)$ або $x / y / z$
	$(a^{**3} + b^{**3}) / (b * c)$
	$(a[i+1] + b[i-1]) / (2 * x * y)$
	$(-b + \text{sqrt}(b * b - 4 * a * c)) / (2 * a)$
$(x < 0)$	$\text{sign}(x) * \text{abs}(x) ** (1/5)$
	$0.49 * \exp(a * a - b * b) + \ln(\cos(a * a)) ** 3$
	$x / (1 + x * x / (3 + (2 * x)^{**3}))$

Типові помилки в записі виразів:

$5x + 1$ $a + \sin x$ $((a + b)/c)^{**3}$	Пропущений знак множення між 5 і x Аргумент x функції sin x не поміщений в дужки Не вистачає закриваючої дужки
---	--

Логічні вирази

У записі логічних виразів окрім арифметичних операцій складання, віднімання, множення, ділення і піднесення до ступеня використовуються операції відношення $<$ (менше), $<=$ (менше або рівно), $>$ (більше), $>=$ (більше або рівно), $=$ (рівно), $<>$ (не рівно), а також логічні операції і, або, ні.

Приклади запису логічних виразів, істинних при виконанні вказаних умов.

Умова	Запис на шкільній алгоритмічній мові
Дробна частина дійсного числа a дорівнює нулю	$\text{int}(a) = 0$
Ціле число a — парне	$\text{mod}(a, 2) = 0$
Ціле число a — непарне	$\text{mod}(a, 2) = 1$
Ціле число k кратне семи	$\text{mod}(a, 7) = 0$
Кожне з чисел a, b додатне	$(a > 0) \text{ і } (b > 0)$
Тільки одне з чисел a, b додатне	$((a > 0) \text{ і } (b <= 0)) \text{ або } ((a <= 0) \text{ і } (b > 0))$
Хоча б одне з чисел a, b, c є від'ємним	$(a < 0) \text{ або } (b < 0) \text{ або } (c < 0)$

Число x задовольняє умові $a < x < b$	$(x > a) \text{ і } (x < b)$
Число x має значення в межах $[1, 3]$	$(x \geq 1) \text{ і } (x \leq 3)$
Цілі числа a і b мають однакову парність	$((\text{mod}(a, 2) = 0) \text{ і } (\text{mod}(b, 2) = 0))$ або $((\text{mod}(a, 2) = 1) \text{ і } (\text{mod}(b, 2) = 1))$
Точка c з координатами (x, y) лежить в кругу радіусу r з центром в точці (a, b)	$(x-a)^2 + (y-b)^2 < r^2$
Рівняння $ax^2 + bx + c = 0$ не має дійсних коренів	$b^2 - 4ac < 0$
Точка (x, y) належить першій або третій чверті	$((x > 0) \text{ і } (y > 0))$ або $((x < 0) \text{ і } (y > 0))$
Точка (x, y) лежить поза одиничним кругом з центром в початку координат або його другій чверті	$(x^2 + y^2 > 1)$ або $((x^2 + y^2 \leq 1) \text{ і } (x < 0) \text{ і } (y > 0))$
Цілі числа a і b є взаємно протилежними	$a = -b$
Цілі числа a і b є взаємно зворотними	$a * b = 1$
Число a більше середнього арифметичного чисел b, c, d	$a > (b+c+d) / 3$
Число a не менше середнього геометричного чисел b, c, d	$a \geq (b+c+d) ** (1/3)$
Хоча б одна з логічних змінних $F1$ і $F2$ має значення так	$F1$ або $F2$
Обидві логічні змінні $F1$ і $F2$ мають значення так	$F1$ і $F2$
Обидві логічні змінні $F1$ і $F2$ мають значення ні	не $F1$ і не $F2$
Логічна змінна $F1$ має значення так , а логічна змінна $F2$ має значення ні	$F1$ і не $F2$
Тільки одна з логічних змінних $F1$ і $F2$ має значення так	$(F1 \text{ і не } F2) \text{ або } (F2 \text{ і не } F1)$

Питання:

1. Що таке алгоритм?
2. Що таке "Виконавець алгоритму"?
3. Якими властивостями володіють алгоритми?
4. У якій формі записуються алгоритми?
5. Що таке словесний спосіб запису алгоритмів?
6. Що таке графічний спосіб запису алгоритмів?
7. Що таке псевдокод?
8. Як записуються алгоритми на шкільній алгоритмічній мові?
9. Що таке базові алгоритмічні структури?
10. Які цикли називають ітераційними?
11. Що таке вкладені цикли?
12. Чим відрізняється програмний спосіб запису алгоритмів від інших?

13. Що таке рівень мови програмування?
14. Які у машинних мов гідності і недоліки?
15. Що таке мова асемблера?
16. У чому переваги алгоритмічних мов перед машинними?
17. Які компоненти утворюють алгоритмічну мову?
18. Які поняття використовують алгоритмічні мови?
19. Що таке стандартна функція?
20. Як записуються арифметичні вирази?
21. Як записуються логічні вирази?

Література

1. В.Э. Фигурно. IBM PC для пользователя. Изд. 6, М., "Инфра-М", 1995.
2. К.С. Ахметов, А.Е. Борзенко. Современный персональный компьютер. М., "Компьютер-пресс", 1995.
3. А.Е. Борзенко, А.Г. Фёдоров. Мультимедиа для всех. М., "Компьютер-пресс", 1995.
4. В. Денисов. Windows 95 с самого начала. С-Пб, "Питер", 1996.
5. Н. Николь, Р. Альбрехт. Excel 5.0. Электронные таблицы. М., "Эком", 1994.
6. С.А. Каратыгин, А.Ф. Тихонов, В.Г. Долголаптев. Базы данных. т.1 и 2, М., "ABF", 1995.
7. Информатика. Учебное пособие и сборник задач с решениями. Под общей ред. проф. В.А. Каймина и Л.А. Муравья. М., "БРИДЖ", 1994.
8. А.В. Болдачев. Компьютер IBM PC для детей. М., "Аквариум", 1996.
9. В.Б. Комягин, А.О. Коцюбинский. Самоучитель работы на персональном компьютере. М., Изд. "Триумф", 1996.
10. Лэмонт Вуд. Web-графика. Справочник. С-Пб, Изд. "Питер", 1998.
11. Левин. Самоучитель работы на компьютере. 5-е изд., "Нолидж", М., 1998.
12. Информатика. Под ред. проф. Н.В. Макаровой. 2-е изд. "Финансы и статистика", М., 1998.
13. М.Е. Степанов. Сто графических задач по информатике. Программирование на языке Бейсик. М., РЦИСО, 1995.
14. Ю.А. Шафрин. Информационные технологии. М., Лаборатория Базовых Знаний, 1998.
15. Пэтти Винтер. Microsoft Word 97. Справочник. "Питер", С-Пб, 1998.
16. О.В. Ефимова, Ю.А. Шафрин. Практикум по компьютерной технологии. М., "ABF", 1997.
17. А.М. Кенин, Н.С. Печёнкина. IBM PC для пользователей или как научиться работать на компьютере. 4-е издание. Екатеринбург, Изд. "АРД ЛТД", 1997.
18. В.А. Гольденберг. Введение в программирование. Библиотека школьника. Учебное пособие. Минск, "Харвест", 1997.
19. Н.П. Петрова. Виртуальная реальность. Современная компьютерная графика и анимация. М., "Аквариум", 1997.
20. А.О. Коцюбинский, С.В. Грошев. Современный самоучитель работы в сети Интернет. М., "Триумф", 1997.
21. А.П. Частиков. История компьютера. М., "Информатика и образование", 1996.
22. К.С. Ахметов. Windows 98 для всех. М., "Компьютер-пресс", 1998.
23. С. Пономаренко. Adobe Photoshop 3.0 одним взглядом. "ВНУ-Санкт-Петербург", 1996.
24. Ю.М. Платонов, Ю.Г. Уткин. IBM PC в вопросах и ответах. "Радио и связь", М., 1998.
25. Дибкова Л.М. Информатика та комп'ютерна техніка: Посібник для студентів вищих навчальних закладів. - К.: Видавничий центр "Академія", 2002.
26. Информатика: Комп'ютерна техніка. Комп'ютерні технології. Посіб./За ред. О.І.Пушкаря - К.: Видавничий центр "Академія", 2001.
27. Попов В.Б. Основы комп'ютерних технологій. - М.: Фінанси та статистика, 2002. - 704с.
28. Зайден М. Word 2000 наглядно, понятно, быстро. - М. Лаборатория знаний, 1999. - 336с.
29. Вейскопф Дж. Excel 2000. Базовый курс (русифицированная версия). - СПб.: Корона принт, 2000. - 400с.
30. Васильев А., Андреев А. VBA в Office 2000: Учебный курс - СПб.: Питер, 2001

31. Основы современных компьютерных технологий: Учеб. пособие. Под ред.: А.Д.Хомоненко. - СПб.: ООО "Корона", 1998
32. Руденко В.Д., Макачук О.М., Патланжоглу М.О. Практичний курс інформатики. – К.: Фенікс, 1997. – 304 с.
33. Фигурнов В.Э. IBM PC для пользователя. – М.: ИНФРА-М, 1999. – 640 с.
34. Леонтьев В. Новейшая энциклопедия персонального компьютера. – М.: ОЛМА-ПРЕСС, 1999. – 640 с.
35. Стоцкий Ю. Самоучитель по Office 2000. - СПб: Питер, 1997. – 576 с.
36. С. Коупстейк. MS Office 97. – М.: Бином, 1997. – 224 с.
37. Основы современных компьютерных технологий под.ред. А.Д.Хомсненко. Санк Петербург "Коронп принт", 1998-446с.