Державний вищий навчальний заклад
«Прикарпатський національний університет імені Василя Стефаника»
Кафедра комп’ютерних наук та інформаційних систем

Матеріали для самостійного вивчення дисципліни «Прикладне програмування» для студентів спеціальності 126 «Інформаційні системи та технології»

Івано-Франківськ – 2020

Матеріали для самостійного вивчення дисципліни «Прикладне програмування» для студентів спеціальності 126 «Інформаційні системи та технології» – 2020. – 16 с.

Розробник:
Ізмайлов Артем Вікторович, магістр, асистент кафедри комп'ютерних наук та інформаційних систем.

ВСТУП

Мова С++ чинить значний вплив на сучасні засоби програмування. Синтаксис та стандарти мови стали базою для розробки нових мов. Завдяки своїй універсальності її часто використовують для опису алгоритмів та технологій програмування.

Дисципліна «Прикладне програмування» посідає чільне місце серед інших базових дисциплін. Гармонійне поєднання математичного і прикладного аспектів робить її однаково привабливою як для теоретиків, так і для практиків.

Дана дисципліна формує важливі навики практичної та наукової діяльності бакалавра напрямів підготовки «Інформаційні системи та технології». При вивчення цієї навчальної дисципліни використовуються поняття і методи теорії програмування, мов програмування, математичних дисциплін, теорії алгоритмів тощо.

Мета викладання навчальної дисципліни полягає в:

- оволодінні студентами фундаментальними теоретичними поняттями теорії програмування;
- формуванні практичних навиків реалізації найбільш поширених алгоритмів та методів програмування;
- формуванні практичних навичок застосування теорії програмування для розв'язування прикладних задач природничих, економічних, соціальних та інших наук;
- встановленні предметних зв'язків даної навчальної дисципліни з іншими;
- отриманні студентами теоретичної підготовки і практичних навиків для успішного засвоєння фундаментальних і спеціальних дисциплін навчального плану, а також для можливості вивчення спеціальної літератури.

Завдання викладання навчальної дисципліни: навчити студентів прикладному застосуванню мови С++: практичному використанню структур даних у програмах, введенню-виведенню у консоль, файловому введенню-виведенню. Забезпечити вироблення практичних навичок роботи з вказівниками, посиланнями та операторами динамічного розподілу пам'яті. Забезпечити засвоєння операцій перевантаження функцій, конструкторів та конструкторів копіювання. Студенти повинні набути практичних навичок реалізації файлового обміну та організації представлення даних усередині програми.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- етапи обробки программ на ПЕОМ: редагування, трансляція, компонування;
- основні оператори мови;
- скалярні типи даних;
- структуровані типи даних: масиви, рядки, структури, файли, списки;
- модульний принцип разробки програм;
- подання основних структур даних мовою програмування С++;
- способи застосування динамічної пам'яті для обробки значних масивів даних;
- структуру та основні принципи роботи із бінарними файлами;

вміти:

- розробляти алгоритми методом покрокового уточнення,
- працювати із скалярними даними,
- складати програми обробки масивів,
- використовувати структуровані типи даних,
- реалізовувати багатомодульні програми,
- відлагоджувати програми в середовищі MS Visual Studio,
- організовувати дані всередині програми на основі масивів, списків та бінарних дерев,
- застосовувати динамічну пам'ять для обробки та зберігання даних,
- застосовувати бінарні файли для переносу та подання інформації.

ЗМІСТ ЛЕКЦІЙНОГО КУРСУ

Тема 1. Повторення основ C++.

1) Оператори C++.
2) Робота з масивами.
3) Функції користувача.

Тема 2. Рекурсія.

1) Задачі, які вирішуються за допомогою рекурсивних алгоритмів.
2) Властивості рекурсивних функцій.
3) Задання рекурсивних процедур і функцій у C++.

Тема 3. Динамічна пам’ять.

1) Статична і динамічна пам’ять, стек. Особливості функціонування.
2) Робота із динамічною пам’яттю у C++.
3) Динамічні змінні та масиви у C++.

Тема 4. Лінійні списки.

1) Принципи роботи.
2) Ефективність застосування лінійних списків у порівнянні зі статичними та динамічними масивами.
3) Робота із лінійними списками у C++.

Тема 5. Файли з двійковими даними.

1) Відмінності між текстовими та двійковими даними.
2) Особливості роботи із файлами, які містять двійкові дані.
3) Робота із бінарними файлами у C++.

Тема 6. Бінарні дерева.

1) Принципи роботи.
2) Ефективність застосування бінарних дерев у порівнянні з масивами та лінійними списками.
3) Робота із бінарними деревами у C++.
4) Збалансовані дерева (B-tree).

ЗМІСТ ТЕМ ЛАБОРАТОРНИХ ЗАНЯТЬ КУРСУ

Лабораторне заняття 1
Рекурсивні функції Ч.1

***Мета роботи*:** Навчитись розв'язувати прикладні задачі із застосуванням рекурсивних процедур.

Хід роботи

1. Написати програму для обчислення перших N≤100 членів послідовності Фібоначчі. Число N вводиться з клавіатури. Передбачити повідомлення користувача про уведення ним некоректного числа. Програму реалізувати за допомогою рекурсивної функції на основі безпосереднього визначення послідовності Фібоначчі (спосіб а) у теоретичних відомостях).
2. Написати програму для обчислення перших N≤100 членів послідовності Фібоначчі. Число N вводиться з клавіатури. Передбачити повідомлення користувача про уведення ним некоректного числа. Програму реалізувати за допомогою рекурсивної функції на основі удосконаленої форми обчислення послідовності Фібоначчі (спосіб б) у теоретичних відомостях).
3. Задано лабіринт квадратної форми розміром 6 на 6. Задано точку своїми координатами – початкове місцезнаходження людини. На карті лабіринту одиницею визначена стіна лабіринту, нулем – прохід, цифрою 9 – вихід. Рухатись можна вперед, назад, вправо та вліво. За скільки кроків людина вийде з лабіринту, якщо вихід один?

   Карту (структуру) лабіринту завантажувати з файлу map.txt. У структурі лабіринту передбачити наявність зовнішніх стін, які відображаються у файлі. Обробку масиву-карти НЕ реалізовувати глобальним масивом, а передачею посилання. Початкові координати людини отримувати з клавіатури. Передбачити можливість повідомлення користувача, про уведені ним некоректні початкові координати. Задачу розв'язати за допомогою рекурсивних функцій.
4. Зробити висновки стосовно особливостей написання програм із застосуванням рекурсивних процедур та факторів, які необхідно враховувати при їх застосуванні.

Контрольні запитання

1. Що таке рекурсія?
2. Які Ви знаєте види рекурсії?
3. Що таке стоп-умова?

4. Чим можна замінити рекурсію?

Лабораторне заняття 2
Рекурсивні функції Ч.2

***Мета роботи*:** Навчитись розв'язувати прикладні задачі із застосуванням рекурсивних процедур.

Хід роботи

1. Обчислити значення числа π, використовуючи формулу

$$\frac{\pi^4}{90} = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \frac{1}{5^4} + \frac{1}{6^4} + \frac{1}{7^4} + \dots,$$

   беручи 1000, 2000 членів ряду. Ряд обчислювати за допомогою використання рекурсії.

2. Обчислити ланцюговий дріб:

$$R = 2 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{1}{23 + \cfrac{1}{34 + \cfrac{1}{47 + \dots}}}}},$$

   де $a_n = (n+2)^2 - 2$,

   взявши 100 – 500 членів. ($\approx 2.1414185\ldots$).

3. Обчислити значення числа π, використовуючи формулу

$$\frac{\pi}{4} = \left(1 - \frac{1}{3\times 2^3} + \frac{1}{5\times 2^5} - \frac{1}{7\times 2^7} + \frac{1}{9\times 2^9} - \dots\right) + \left(1 - \frac{1}{3\times 3^3} + \frac{1}{5\times 3^5} - \frac{1}{7\times 3^7} + \frac{1}{9\times 3^9} - \dots\right) =$$

$$= \left(1 + \frac{1}{3}\right) - \frac{1}{3}\times\left(\frac{1}{2^3} + \frac{1}{3^3}\right) + \frac{1}{5}\times\left(\frac{1}{2^5} + \frac{1}{3^5}\right) - \frac{1}{7}\times\left(\frac{1}{2^7} + \frac{1}{3^7}\right) + \frac{1}{9}\times\left(\frac{1}{2^9} + \frac{1}{3^9}\right) - \dots$$

   беручи по 100 – 500 значень в кожній сумі. Ряд обчислювати за допомогою використання рекурсії.

4. Зробити висновки стосовно особливостей написання програм із застосуванням рекурсивних процедур та факторів, які необхідно враховувати при їх застосуванні.

Контрольні запитання

1. Що таке рекурсія?
2. Які Ви знаєте види рекурсії?
3. Що таке стоп-умова?
4. Чим можна замінити рекурсію?

Лабораторне заняття 3
Динамічна пам’ять Ч.1

***Мета роботи*:** Навчитись розв’язувати прикладні задачі із застосуванням одновимірних динамічних масивів.

Хід роботи

1. Для заданого цілого числа n створити динамічний масив розміру n, заповнити його числами 1, 2, 3, 4, 5, 6, ... , 5, 4, 3, 2, 1. Вивести масив на екран.
2. Для введеного з клавіатури цілого числа n створити масив із n дійсних чисел від 0.0000 до 10.0000 та впорядкувати його за зростанням елементів і вивести на екран. Масив заповнювати псевдовипадковими числами у заданих межах або використати періодичні функції з параметрами, наприклад, $c \cdot \sin(ax + b) + d$.
3. Знайти мільйонне просте число. Для цього використати динамічний масив, в який записувати послідовні прості числа. При перевірці подільності для ефективності перевіряти подільність на всі **прості** числа, що не перевищують кореня тестованого числа, і виконувати **break** одразу після появи ознаки того, що число складене. Масив має вигляд $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, …, $p_{999999} = 15485863$.

   Зауваження: умову $p \le \sqrt{x}$ найшвидше перевіряє конструкція p*p<=x, а не p<=sqrt(x) чи p<=pow(x, 0.5).
4. Створити три динамічні масиви X, Y, Z – кожен із 1000 символів;
   створити звичайний масив K із 32 символів.
   Закодувати рядок X методом здійснення побайтового XOR із ключовим словом K. Результат занести в масив Y.
   Масиви X та K зчитати з файлів text.txt та key.txt, відповідно.
   Приклад: для тексту "Information" та ключа "Key":

I	n	f	o	r	m	a	t	i	o	n	\0	.	.	.	

XOR

K	e	y	K	e	y	K	e	y	K	e					

=

?	?	?	?	?	?	?	?	?	?	?	\0				

   Тобто **Y[i] = X[i] ^ K[i%keylen],** де keylen – довжина ключа.
   Отриманий шифр роздрукувати (можливі спецсимволи). При роздрукуванні (тільки при роздрукуванні, але **не в масиві** Y) всі символи, ASCII-коди яких менші за 32, замінити крапками.
   Розкодувати масив Y в масив Z (застосувати повторно операцію XOR-ення для масиву Y за тим самим ключем).
   Якщо все зроблено правильно, то текст Z повинен співпадати з текстом X.

5. Зробити висновки стосовно особливостей написання програм із застосуванням рекурсивних процедур та факторів, які необхідно враховувати при їх застосуванні.

Контрольні запитання

1. Що таке динамічна пам’ять і як вона працює?
2. Як оголосити динамічну змінну?
3. Як оголосити динамічний масив?
4. Як і для чого необхідно очищати динамічну пам’ять після використання?

Лабораторне заняття 4
Динамічна пам’ять Ч.2

***Мета роботи*:** Навчитись розв’язувати прикладні задачі із застосуванням двовимірних динамічних масивів.

Хід роботи

1. Ввести з клавіатури числа I та J. Створити динамічний масив розміру I×J цілих чисел, який заповнити такими значеннями: 1) рамка – числами 1; 2) друга рамка – числами 2; 3) все, що залишилося – числами 5. Масив роздрукувати у вигляді таблиці.
   Напр., для I=5, J=6 потрібно створити таблицю:
   111111
   122221
   125521
   122221
   111111
2. Ввести з клавіатури числа I та J. Створити динамічний масив розміру I×J цілих чисел, який заповнити псевдовипадковими числами з діапазону 0…999. Створити масив цілих чисел розміру I, в який занести максимальні значення таблиці по рядках. Створити масив цілих чисел розміру J, в який занести максимальні значення таблиці по стовпцях. Знайти також загальний максимум. Роздрукувати отримані результати.
   Приклад виконання для I = 3, J = 5.

```
34   54  12   3  45  |   54
212 234 533 332   2  |  533
243  33  57 546 765  |  765
--------------------+------
243 234 533 546 765  |  765
```

3. Зробити висновки стосовно особливостей написання програм із застосуванням рекурсивних процедур та факторів, які необхідно враховувати при їх застосуванні.

Контрольні запитання

1. Що таке динамічна пам’ять і як вона працює?
2. Як оголосити динамічну змінну?
3. Як оголосити динамічний масив?
4. Як і для чого необхідно очищати динамічну пам’ять після використання?

Лабораторне заняття 5
Лінійні списки

***Мета роботи*:** Навчитись розв’язувати прикладні задачі із застосуванням двовимірних лінійних списків.

Хід роботи

1. Написати програму, де використовуються списки студентів:
   1.1 Структура Student містить такі поля: ім'я, прізвище, 3 оцінки (цілі числа) та рейтинг (дійсне число).
   1.2 Написати програму, яка містить усі необхідні операції для роботи зі списком, не змінює порядку елементів у процесі додавання, забезпечує знаходження довжини списку.
   1.3 Організувати впорядкований за прізвищем список студентів.
   1.4 Продемонструвати роботу всіх функцій роботи зі списком.

Контрольні запитання

1 Яка структура вузлів лінійного списку?
2 Які основні властивості лінійного списку?
3 Опишіть кроки алгоритму видалення довільного елементу лінійного списку.
4 У яких випадках доцільно використовувати лінійні списки замість масивів та динамічних масивів?

Лабораторне заняття 6
Робота з файлами. Файли з двійковими даними

***Мета роботи*:** Навчитись розв’язувати прикладні задачі із застосуванням файлів, які містять двійкові дані.

Хід роботи

1. Маючи файл "1.bmp", написати програму, яка генерує такі файли:
   "2.bmp" – копія вихідного файлу, але зі зменшеним вдвічі червоним та синім кольором;
   "3.bmp" – копія вихідного файлу з додаванням "шуму" – до кольору точки пододавати випадкові числа;
   "4.bmp" – на свій смак відредагований вихідний файл (зміна кольорів, градієнтне додавання кольорів, ...)
2. Зчитати файл "1.txt" в динамічний масив символів X. **Файл читати як двійковий.**
   Створити ще один динамічний масив Y такого ж розміру, що й X.
   Зчитати з клавіатури літеру в літерну змінну (нехай в char c).
   Зашифрувати рядок X методом здійснення побітового XOR із ключем c. Результат занести в масив Y.
   Тобто:
   ```
   Y[i] = X[i] ^ c.
   ```
   Отриманий шифр занести у файл "2.txt".
3. Написати програму підбору літери c для розшифрування файла "2.txt", отриманого від попередньої програми. Результати підбору, для яких усі символи дешифрованого «тексту» є літерами або цифрами (їх ASCII-коди входять у допустимі для читабельних символів межі), виводити на екран.
4. Маючи файл "1.bmp", на його подобу (використовуючи шапку та габарити) створити файл "2.bmp", для кожної точки якого придумати залежність кольору від координат h та w.
   Напр.,
   Градієнт по червоному кольору (w = 0: p.r = 0; w = Width-1: p.r = 255):
   ```
   p.r = 256*w/Width;
   ```
   або періодичні повторюваності по вертикалі:
   ```
   p.g = 200*sin(h/100);
   ```
   Можна малювати круги, напр.,
   ```
   radius2 = (w-wo)*(w-wo)+(h-ho)*(h-h0) ;
   if (radius2 < 100 {p.r = radius; p.g = 255 – radius; p.b = radius+100;}
   ```
   І т. д.

Контрольні запитання

1. Яка відмінність текстового файлу від бінарного?
2. Які переваги зберігання інформації у бінарному файлі?
3. Які функції використовуються для роботи з бінарними файлами?
4. Які особливості роботи із .bmp файлами, як двійковими?

Лабораторне заняття 7
Бінарні дерева

***Мета роботи*:** Навчитись розв’язувати прикладні задачі із застосуванням бінарних дерев.

Хід роботи

1. Написати програму, в якій дані про студентів записуються в дерева. Вузол дерева містить таку корисну інформацію: прізвище (string), рік народження, оцінка:
   1.1. Зчитати з текстового файлу 10 "студентів" у двійкове дерево, яке організоване за порядком прізвищ.
   1.2. Роздрукувати отримане дерево.
   1.3. "Пересипати" дані з першого дерева у друге дерево того ж типу, тільки організованого за роком народження; роздрукувати його.
   1.4. "Пересипати" дані з другого дерева у третє дерево того ж типу, тільки організованого за оцінками; роздрукувати його.
   1.5. Дописати функцію видалення з пам’яті всього дерева (рекурсивна функція, яка: 1) видаляє ліве піддерево, а ліву гілку занулює; 2) видаляє праве піддерево, а праву гілку занулює; видаляє сам вузол, потім зануливши вказівник на нього).
   1.6. Видалити з пам’яті всі три дерева та продемонструвати їх відсутність у пам’яті.

Контрольні запитання

1. Яка структура вузла бінарного дерева?
2. Які переваги зберігання інформації у вигляді бінарного дерева?
3. За яким принципом будуються упорядковані бінарні дерева?
4. Що таке B-tree?

Рекомендована література

Базова література

1. Г. Шилдт. Полный справочник по C++. 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2010. – 800 с.
2. Лаптев В. В., Морозов А. В., Бокова А. В. C++. Обьектно-ориентированное программирование. Задачи и упражнения. – СПб.: Питер, 2007. – 288 с.: ил.
3. Эккель Б. Философия C++. Введение в стандартный C++. 2-е изд. – СПб.: Питер, 2004. – 572 с.: ил.

Допоміжна література

1. Ivor Horton’s Beginning Visual C++® 2010. Wiley Publishing, Inc. – 1276
2. Мозговой М. В. C++ Мастер-класс. 85 нетривиальных проектов, решений и задач. – СПб.: Наука и Техника, 2007. – 272 с.
3. Якушев Д. М. «Философия» программирования на языке C++. / Д. М. Якушев. — 2_е изд. — М.: Бук_пресс, 2006. — 320 с.

Інформаційні ресурси

www.scientific-library.net – Електронна бібліотека науково-технічної літератури

www.elibrary.ru – Наукова електронна бібліотека науково-технічної літератури