

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра информационных технологий автоматизированных систем

**С.С. Смородинский, Н.В. Батин**

## **ОПТИМИЗАЦИЯ РЕШЕНИЙ НА ОСНОВЕ КОМПЬЮТЕРНЫХ ИМИТАЦИОННЫХ МЕТОДОВ И МОДЕЛЕЙ**

Учебное пособие  
по курсу “Моделирование систем”  
для студентов специальности  
«Автоматизированные системы обработки информации»  
дневной и дистанционной форм обучения

**В 2-х частях**

Часть 1

Минск 2004

УДК 681.3 + 007 (075.8)

ББК 32.97 я 73

С 51

Рецензент:

заведующий кафедрой метрологии и стандартизации БГУИР,  
д-р техн. наук, проф. В.И. Кириллов

**Сморodinский С.С.**

С 51

Оптимизация решений на основе компьютерных имитационных методов и моделей: Учеб. пособие по курсу “Моделирование систем” для студ. спец. “Автоматизированные системы обработки информации” дневн. и дистанц. форм обуч. В 2 ч. Ч. 1/ С.С. Смородинский, Н.В. Батин. - Мн.: БГУИР, 2004.- 80 с.: ил.

ISBN 985-444-707-3 (ч.1)

В пособии приводится теоретический и практический материал, связанный с анализом и оптимизацией решений на основе имитационного моделирования. Рассматриваются алгоритмы имитации технических и экономических систем на основе метода Монте-Карло и их компьютерная реализация.

Пособие предназначено для студентов специальности “Автоматизированные системы обработки информации”, изучающих перспективные технологии поддержки принятия решений в задачах прогнозирования, планирования, проектирования и управления. Пособие рекомендуется использовать при изучении курсов, связанных с применением методов компьютерного моделирования и поддержки принятия решений, в курсовом и дипломном проектировании. Пособие представляет интерес для специалистов, практическая деятельность которых связана с решением задач компьютерного моделирования в различных сферах науки, техники, экономики.

**УДК 681.3 + 007 (075.8)**

**ББК 32.97 я 73**

**ISBN 985-444-707-3 (ч.1)**

**ISBN 985-444-726-X**

© С.С. Смородинский, Н.В. Батин, 2004

© БГУИР, 2004

## СОДЕРЖАНИЕ

### Введение

#### 1. МЕТОД МОНТЕ-КАРЛО: КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ И ПРИНЦИПЫ РЕАЛИЗАЦИИ

- 1.1. Назначение и принцип работы метода Монте-Карло
- 1.2. Случайные равномерно распределенные числа

#### 2. РЕШЕНИЕ ЗАДАЧ МОДЕЛИРОВАНИЯ НА ОСНОВЕ АЛГОРИТМОВ ИМИТАЦИИ СЛУЧАЙНЫХ СОБЫТИЙ

- 2.1. Имитация случайных событий
- 2.2. Примеры решения задач на основе алгоритмов имитации случайных событий

#### 3. РЕШЕНИЕ ЗАДАЧ МОДЕЛИРОВАНИЯ НА ОСНОВЕ АЛГОРИТМОВ ИМИТАЦИИ СЛУЧАЙНЫХ ВЕЛИЧИН

- 3.1. Имитация дискретных случайных величин
- 3.2. Имитация непрерывных случайных величин
- 3.3. Примеры решения задач на основе алгоритмов имитации случайных величин
- 3.4. Интервальные оценки в имитационных моделях
- 3.5. Средства имитационного моделирования в Excel

#### 4. АНАЛИЗ И ОПТИМИЗАЦИЯ РЕШЕНИЙ В ДЕТЕРМИНИРОВАННЫХ ЗАДАЧАХ НА ОСНОВЕ МЕТОДА МОНТЕ-КАРЛО

- 4.1. Применение метода Монте-Карло для решения детерминированных задач оптимизации
- 4.2. Примеры решения задач оптимизации на основе метода Монте-Карло

### Литература

Приложение 1. Случайные равномерно распределенные числа

Приложение 2. Работа с системой программирования Visual Basic for Applications

Приложение 3. Некоторые сведения о вероятностных характеристиках случайных величин

Приложение 4. Алгоритмы имитации непрерывных случайных величин

Приложение 5. Примеры задач, решаемых на основе метода Монте-Карло

“Имитационное моделирование является мощным инструментом исследования поведения реальных систем.”

Таха Х., 2001 [1]

“По данным Института прикладной математики им. М.В. Келдыша, экономический эффект от внедрения методов математического моделирования в 10-100 раз превышает соответствующие затраты.”

Харин Ю.С. и др., 1997 [2]

## ВВЕДЕНИЕ

В учебном пособии рассматриваются методы анализа и принятия решений с использованием перспективных компьютерных технологий имитационного моделирования. В первой части пособия рассматривается метод Монте-Карло, который образует алгоритмическую основу для имитации случайных явлений в имитационном моделировании и который представляет собой универсальный компьютерный метод решения детерминированных и вероятностных задач (имитационные программы представлены на языке VBA). Во второй части рассматриваются особенности моделирования и оптимизации решений в среде новой перспективной системы GPSS World (имитационные программы представлены на языке GPSS World).

Моделирование представляет собой универсальный метод научного исследования, причем сущность метода состоит в построении модели и исследовании свойств моделируемой системы. Если реализуется математическое моделирование, то модель может быть либо аналитической, либо имитационной, либо комбинированной. В конечном счете это компьютерная программа. Таким образом, речь идет о компьютерном моделировании исследуемой системы в заданном классе вероятных условий. При этом “имитационное моделирование представляет собой достаточно гибкий инструмент исследования, который можно эффективно использовать при анализе сложных систем” [1, с.704].

Имитационное моделирование предусматривает имитацию поведения исследуемой системы в заданном классе вероятных условий (Computer Simulation или Simulation). Имитационное моделирование применяется обычно в тех случаях, когда проблема не поддается аналитическому решению, а непосредственные эксперименты с системой либо невозможны, либо нецелесообразны. Имитационная модель системы создается с использованием специальных средств программной поддержки. Экспериментируя с имитационной компьютерной программой, исследователь изучает свойства, закономерности и особенности познаваемого объекта, проверяет на модели эффективность предлагаемых решений, разрабатывает решения по оптимизации и рационализации исследуемой системы [1-4, 9].

К перспективным системам имитационного моделирования предъявляются следующие требования: 1) активное диалоговое общение пользователя с моделью; 2) простой язык описания модели, доступный для пользователя-непрограммиста; 3) возможность оперативной модификации модели и настройки на задачу пользователя; 4) возможность использования интерактивного редактора, позволяющего модифицировать синтаксис и семантику исходного текста; 5) возможность контроля процесса имитации в интерактивном режиме; 6) возможность диалоговой работы с архивом, содержащим типовые модели, результаты моделирования и др.; 7) организация базы знаний для поддержки имитационного моделирования; 8) использование программных средств искусственного интеллекта для организации вывода заключений; 9) использование программных средств, обеспечивающих режим реального времени; 10) отражение динамики имитируемого процесса и результатов имитации в удобной для восприятия форме [9-14].

Структура перспективной системы имитационного моделирования должна иметь развитую программную среду для поддержки процессов *подготовки, трансляции* и *выполнения* имитационных программ, причем процесс подготовки включает описание задачи пользователя, компоновку моделирующего алгоритма и формирование исходного текста имитационной программы. В перспективных системах имитационного моделирования реализуется идея «конструирования» имитационных программ из набора готовых блоков, причем трудоемкость имитационного моделирования уменьшается за счет повышения уровня языка программирования, автоматизации и интеллектуализации типовых модельных процедур. В процессе эволюции систем имитационного моделирования постепенно снижаются требования к уровню квалификации пользователя в части программирования (системы имитационного моделирования GPSS World, Arena 4, Simplex 3 и др.).

В первой части учебного пособия рассматриваются особенности компьютерной реализации метода Монте-Карло (раздел 1), алгоритмические основы метода и технология решения вероятностных и детерминированных задач с использованием имитационных компьютерных программ (разделы 2-4). Материал пособия образует основу для освоения перспективных специализированных программных средств поддержки имитационного моделирования, которые будут рассмотрены во второй части пособия. Рекомендуемая литература включает источники по математическим и алгоритмическим основам имитационного моделирования [1-12, 15], перспективным компьютерным системам имитационного моделирования [9-11, 13, 14], методам анализа и оптимизации решений на основе аналитических и имитационных моделей [1, 7-9, 15, 16].

# 1. МЕТОД МОНТЕ-КАРЛО: КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ И ПРИНЦИПЫ РЕАЛИЗАЦИИ

## 1.1. Назначение и принцип работы метода Монте-Карло

Метод Монте-Карло представляет собой универсальный метод, применяемый для решения задач самых различных классов: вероятностных и детерминированных, задач с дискретными и непрерывными величинами, задач моделирования и оптимизации и т.д.

Под *вероятностными* задачами будем понимать любые задачи, связанные с анализом случайных явлений (случайных событий, величин, процессов). *Детерминированные* задачи – это задачи, в постановке которых нет никаких случайных факторов.

Под задачами *с дискретными величинами* будем понимать задачи, в которых все анализируемые величины могут принимать значения только из некоторого конечного множества допустимых значений. Задачи *с непрерывными величинами* – это задачи, в которых анализируемые величины могут принимать любые значения из некоторого диапазона (этот диапазон может быть как ограниченным, так и неограниченным).

Задачи *моделирования* – это задачи, связанные с имитацией некоторого объекта (например, сложной технической системы) или явления (например, процесса развития отрасли) с целью определения их характеристик. Задачи *оптимизации* – это задачи, в которых требуется выбрать лучший вариант решения из нескольких возможных.

Большинство практических задач не могут быть однозначно отнесены к одному из приведенных выше классов, а содержат элементы, характерные для нескольких из этих классов.

В данном пособии в основном рассматривается применение метода Монте-Карло для решения задач моделирования сложных систем (производственных, технических, экономических и т.д.). Как правило, функционирование таких систем связано с разнообразными случайными факторами.

Решение задач моделирования на основе метода Монте-Карло основано на *многократной* имитации исследуемого явления, т.е. получении большого набора конкретных реализаций этого явления. Полученный набор реализаций затем обрабатывается с целью вычисления характеристик исследуемого явления, представляющих интерес в решаемой задаче.

Конкретный алгоритм решения задачи с помощью метода Монте-Карло полностью зависит от постановки задачи.

Во многих случаях задачи, решаемые методом Монте-Карло, в принципе могут быть решены другими (аналитическими) методами, т.е. путем расчетов по определенным формулам и алгоритмам. Однако применение аналитических методов, как правило, возможно только для задач небольшой размерности (с небольшим количеством исследуемых факторов). Для реальных задач применение аналитических методов нередко оказывается

невозможным или нецелесообразным из-за большого объема сложных вычислений. Поэтому во многих случаях применение метода Монте-Карло оказывается единственно возможным способом решения задачи.

## 1.2. Случайные равномерно распределенные числа

Метод Монте-Карло основан на применении *случайных равномерно распределенных чисел* (СРРЧ). Расчет (розыгрыш) СРРЧ выполняется по специальным алгоритмам, позволяющим получать бесконечную последовательность таких чисел. Эти алгоритмы разработаны таким образом, что СРРЧ *всегда* принимают значения из диапазона *от нуля до единицы*. При этом СРРЧ обладают следующими свойствами:

- равномерность: СРРЧ могут принимать значения из любой части диапазона  $(0;1)$  с одинаковой частотой;
- независимость: любое из значений СРРЧ не зависит от предыдущих СРРЧ;
- случайность: в последовательности СРРЧ нет какой-либо закономерности.

Эти свойства СРРЧ позволяют рассматривать их как случайные величины, распределенные по равномерному закону в диапазоне от нуля до единицы.

Рассмотрим один из алгоритмов получения СРРЧ – **алгоритм Лемера**. Для этого алгоритма должны быть заданы три параметра (некоторые числа), обозначаемые как  $R_0$ ,  $a$ ,  $m$ . Значения этих параметров подбираются в процессе отладки алгоритма таким образом, чтобы получаемые СРРЧ соответствовали свойствам равномерности, случайности и независимости. Розыгрыш СРРЧ на основе алгоритма Лемера реализуется в следующем порядке.

1. Вычисляется произведение  $aR_{n-1}$ , где  $n$  – номер разыгрываемого СРРЧ.
2. Вычисляется величина  $R_{n-1}$  как остаток от деления числа, полученного на шаге 1, на параметр  $m$ :

$$R_n = \text{mod}(aR_{n-1}/m), \quad (1.1)$$

где  $\text{mod}$  – остаток от деления.

3. Вычисляется СРРЧ:

$$R = \frac{R_n}{m}. \quad (1.2)$$

Приведем пример розыгрыша трех СРРЧ на основе алгоритма Лемера при  $a=3$ ,  $m=7$ ,  $R_0=2$ .

*Розыгрыш 1.* Вычисляется произведение  $aR_{n-1}=aR_0=3 \cdot 2=6$ . Вычисляется величина  $R_n$ :  $R_n=R_1=\text{mod}(6/7)=6$ . Вычисляется СРРЧ:  $R=6/7=0,8571$ .

*Розыгрыш 2.* Вычисляется произведение  $aR_{n-1}=aR_1=3 \cdot 6=18$  (здесь  $R_{n-1}=R_1=6$  было вычислено в предыдущем розыгрыше). Вычисляется величина  $R_n$ :  $R_n=R_2=\text{mod}(18/7)=4$ . Вычисляется СРРЧ:  $R=4/7=0,5714$ .

*Розыгрыш 3.* Вычисляется произведение  $aR_{n-1}=aR_2=3\cdot 4=12$ . Вычисляется величина  $R_n$ :  $R_n=R_3=\text{mod}(12/7)=5$ . Вычисляется СРРЧ:  $R=5/7=0,7143$ .

Для проверки и отладки алгоритма выполняется розыгрыш большого количества СРРЧ и производится их проверка на равномерность, независимость и случайность на основе методов математической статистики. Например, для проверки соответствия СРРЧ равномерному закону распределения может применяться критерий “хи-квадрат”, рассматриваемый в п.3.2.4.

Имеются и другие алгоритмы розыгрыша СРРЧ. Одна из возможных реализаций СРРЧ приведена в прил.1.

Алгоритмы для получения СРРЧ реализованы практически во всех языках программирования и во многих прикладных программах обработки данных. Например, в языке Visual Basic для получения СРРЧ используется функция RND, в языке Delphi – функция RANDOM, в табличном процессоре EXCEL – функция СЛЧИС.

СРРЧ применяются для многократной имитации случайных явлений, характерных для объекта моделирования. Примеры имитации случайных явлений на основе СРРЧ будут рассмотрены ниже.

## **2. РЕШЕНИЕ ЗАДАЧ МОДЕЛИРОВАНИЯ НА ОСНОВЕ АЛГОРИТМОВ ИМИТАЦИИ СЛУЧАЙНЫХ СОБЫТИЙ**

### **2.1. Имитация случайных событий**

Под случайным событием будем понимать любое событие, которое может произойти с некоторой вероятностью  $P$  (или не произойти с вероятностью  $1-P$ ).

**Имитация случайного события** на основе метода Монте-Карло выполняется следующим образом. Разыгрывается СРРЧ  $R$ . Если  $0 \leq R \leq P$  (где  $P$  - вероятность имитируемого события), то смоделировано появление данного события; если  $R \geq P$ , то смоделировано отсутствие события.

Обоснование этого алгоритма следующее. Методами теории вероятностей можно доказать, что вероятность попадания СРРЧ в любой интервал, имеющий длину  $P$  (где  $0 \leq P \leq 1$ ), равна самой величине  $P$ . Таким образом, вероятность попадания СРРЧ  $R$  в интервал  $(0; P)$  равна  $P$ . Другими словами, попадания СРРЧ  $R$  в диапазон  $(0; P)$  происходят с той же частотой, что и имитируемое событие, вероятность которого также равна  $P$ .

Например, вероятность попадания СРРЧ в интервал  $(0; 0,4)$  равна  $0,4$ . Это означает, что из 100 разыгранных СРРЧ примерно 40 принимают значения из интервала от 0 до  $0,4$ . Поэтому при попадании СРРЧ  $R$  в интервал  $(0; 0,4)$  можно считать, что смоделировано событие, вероятность которого равна  $0,4$ .

Примечание. Как отмечено выше, вероятность попадания СРРЧ в *любой* интервал, имеющий длину  $P$  (где  $0 \leq P \leq 1$ ), равна  $P$ . Например, вероятность попадания СРРЧ в любой интервал, имеющий длину  $0,4$  (например, в интервал

от 0,3 до 0,7 или от 0,6 до 1), также равна 0,4. Поэтому для моделирования события, вероятность которого равна  $P$ , может использоваться не только интервал  $(0; P)$ , но и любой другой интервал, имеющий длину  $P$  (из диапазона от 0 до 1).

**Пример 2.1.** Предприятие выпускает некоторые изделия. На основе наблюдений известно, что в 8% изделий имеется брак. Выпуск годного изделия дает предприятию прибыль в размере 7 ден.ед., выпуск бракованного изделия – убыток в размере 5 ден.ед. Требуется разработать алгоритм имитации выпуска изделий на основе метода Монте-Карло и реализовать его в виде программы. Определить среднюю прибыль предприятия от выпуска одного изделия.

В данном примере брак в изделии - случайное событие, вероятность которого равна 0,08.

Алгоритм имитации выпуска изделий может быть следующим.

Разыгрывается СРРЧ  $R$ . Если  $R < 0,08$ , то смоделирован выпуск бракованного изделия; прибыль предприятия уменьшается на 5 ден.ед. Если  $R \geq 0,08$ , то смоделирован выпуск годного изделия; прибыль предприятия увеличивается на 7 ден.ед. Эти действия выполняются многократно, например, 100 000 раз (выполняется 100 000 *испытаний*).

Таким образом, каждое испытание в этом алгоритме представляет собой имитацию выпуска одного изделия.

В табл.2.1 приведены пять испытаний алгоритма. Случайные числа (СРРЧ  $R$ ) взяты из прил.1.

Здесь, например, в первом испытании разыграно СРРЧ  $R=0,0795$ . Так как  $R < 0,08$ , смоделирован выпуск бракованного изделия, т.е. убыток в размере 5 ден.ед. Во втором испытании разыграно СРРЧ  $R=0,3780$ . Так как  $R \geq 0,08$ , смоделирован выпуск годного изделия, т.е. прибыль в размере 7 ден.ед.

Таблица 2.1

Номер испытания	$R$	Изделие годное/бракованное	Прибыль (убыток)
1	0,0795	Бракованное	-5
2	0,3780	Годное	7
3	0,0593	Бракованное	-5
4	0,7602	Годное	7
5	0,2847	Годное	7

Приведем программу на языке Visual Basic for Applications (VBA), реализующую алгоритм имитации выпуска изделий (основные правила работы с системой программирования VBA приведены в прил.2).

```
sub primer21()
' Имитация выпуска изделий
n=100000 : p=0.02 : prib=7 : ub=5
for i=1 to n
```

```

r=rnd
if r<p then
sumprib=sumprib-ub
else
sumprib=sumprib+prib
end if
next i
sredprib=sumprib/n
msgbox("Средняя прибыль от выпуска одного изделий "+str(round(sredprib,2)))
end sub

```

Результат выполнения этой программы должен быть следующим: средняя прибыль от выпуска одного изделия - примерно 6,76 ден.ед.

**Примечание.** Алгоритм имитации выпуска изделий можно было реализовать и по-другому: если  $R < 0,92$ , то смоделирован выпуск *годного* изделия, если  $R \geq 0,92$  - выпуск бракованного изделия.

Рассмотрим имитацию **полной группы событий**. Полная группа событий  $A_1, A_2, \dots, A_n$  - это события, которые являются несовместными (т.е. не могут произойти никакие два события вместе), но одно из них происходит обязательно. Сумма вероятностей таких событий равна единице:  $P_1 + P_2 + \dots + P_n = 1$ .

Полная группа событий имитируется следующим образом. Диапазон  $(0; 1)$  разбивается на интервалы  $(0; P_1)$ ,  $[P_1; P_1 + P_2)$ ,  $[P_1 + P_2; P_1 + P_2 + P_3)$ , ...,  $[1 - P_n; 1)$ . Разыгрывается СРРЧ  $R$ . Определяется интервал (с некоторым номером  $j$ ), в который попадает значение  $R$ , и фиксируется  $j$ -е событие ( $A_j$ ).

**Пример 2.2.** Предприятие по ремонту бытовой электроники выполняет ремонт радиоприемников, телевизоров, магнитофонов и телефонов. Известно, что примерно 20% заказов, поступающих на предприятие, составляют заказы на ремонт радиоприемников, 40% - заказы на ремонт телевизоров, 15% - заказы на ремонт магнитофонов, 25% - заказы на ремонт телефонов (будем считать, что по каждому заказу требуется ремонт только одного изделия). Прибыль предприятия от ремонта одного радиоприемника составляет 35 ден.ед, телевизора – 60 ден.ед., магнитофона – 40 ден.ед., телефона – 25 ден.ед.

Требуется разработать алгоритм имитации работы ремонтного предприятия на основе метода Монте-Карло и реализовать его в виде программы.

Определить среднюю прибыль предприятия от выполнения одного заказа.

В данном примере виды заказов (заказы на ремонт радиоприемников, телевизоров, магнитофонов и телефонов) составляют полную группу событий. Алгоритм имитации работы предприятия может быть следующим. Разыгрывается СРРЧ  $R$ . Если  $R < 0,2$ , то смоделировано поступление заказа на ремонт радиоприемника. Если  $0,2 \leq R < 0,6$ , то смоделирован заказ на ремонт телевизора, если  $0,6 \leq R < 0,75$  – заказ на ремонт магнитофона, если  $0,75 \leq R < 1$  – заказ на ремонт телефона. Прибыль предприятия увеличивается на соответствующую величину (35, 60, 40 или 25 ден.ед.). Эти действия повторяются многократно, например, 100 000 раз.

В табл.2.2 приведены пять испытаний алгоритма.

Таблица 2.2

Номер испытани я	$R$	Заказ	Прибы ль
1	0,07 95	Радиоприемни к	35
2	0,37 80	Телевизор	60
3	0,05 93	Радиоприемни к	35
4	0,76 02	Телефон	25
5	0,28 47	Телевизор	60

Приведем программу на языке VBA, реализующую алгоритм имитации работы ремонтного предприятия.

```

sub primer22()
' Имитация работы ремонтного предприятия
n=100000
for i=1 to n
r=rnd
if r<0.2 then sumprib=sumprib+35
if (r>=0.2)and(r<0.6) then sumprib=sumprib+60
if (r>=0.6)and(r<0.75) then sumprib=sumprib+40
if r>=0.75 then sumprib=sumprib+25
next i
sredprib=sumprib/n
msgbox("Средняя прибыль от ремонта одного изделия "+str(round(sredprib,2)))
end sub

```

Результат имитации должен быть следующим: средняя прибыль от ремонта одного изделия - примерно 43,28 ден.ед.

**Примечание.** Алгоритм имитации работы предприятия можно было реализовать и по-другому. Например, можно разбить диапазон (0; 1) на интервалы (0; 0,15), [0,15; 0,4), [0,4; 0,8), [0,8;1). В этом случае, если  $R < 0,15$ , то следует считать, что смоделирован заказ на ремонт магнитофона; если  $0,15 \leq R < 0,4$ , то смоделирован заказ на ремонт телефона, если  $0,4 \leq R < 0,8$  – заказ на ремонт телевизора, если  $0,8 \leq R < 1$  – заказ на ремонт радиоприемника.

## 2.2. Примеры решения задач на основе

### алгоритмов имитации случайных событий

**Пример 2.3.** При выпуске некоторых деталей требуется последовательная обработка на двух станках: сначала – на токарном, затем – на шлифовальном. Материал, из которого изготавливается деталь, стоит 20 ден.ед. Затраты, связанные с обработкой детали на токарном станке, составляют 5 ден.ед., на шлифовальном – 4 ден.ед.

При обработке на токарном станке в 8% случаев толщина детали оказывается меньше заданной, в 12% случаев – больше заданной, в остальных 80% случаев – нормальной. Детали, толщина которых оказывается меньше заданной, бракуются и не поступают на дальнейшую обработку (т.е. на шлифовальный станок). Детали, толщина которых больше заданной, исправляются. Затраты на исправление одной детали – 3 ден.ед.

При обработке на шлифовальном станке возможны дефекты двух видов: некачественная обработка верхней или нижней поверхности. Некачественная обработка верхней поверхности допускается в 3% случаев, нижней – в 6% случаев. Дефект может быть допущен как на одной из поверхностей, так и на обеих. Если имеется дефект только одной поверхности, то он устраняется. Затраты на устранение любого дефекта составляют 2 ден.ед. Если допущены дефекты обеих поверхностей, то они не устраняются, а деталь бракуется.

Готовые детали продаются по 35 ден.ед.

Требуется разработать алгоритм имитации производства деталей на основе метода Монте-Карло и реализовать его в виде программы. Определить: а) вероятность выпуска годной детали; б) среднюю прибыль предприятия от выпуска одной детали; в) определить, является ли выгодной замена токарного станка на новый, если при такой замене качество деталей повысится (толщина детали будет меньше заданной в 2% случаев, больше заданной – в 3% случаев), но затраты на обработку детали на токарном станке будут составлять 6 ден.ед.

Разработаем алгоритм имитации выпуска деталей (для случая, когда используется старый токарный станок). При разработке алгоритма необходимо учитывать, что после обработки на токарном станке деталь может оказаться нормальной (в 80% случаев), иметь толщину меньше заданной (в 8% случаев) или больше заданной (в 12% случаев). Очевидно, что одна и та же деталь не может иметь толщину и больше, и меньше заданной. После обработки на шлифовальном станке деталь может не иметь дефектов, может иметь дефект одной из поверхностей или дефект обеих поверхностей. Годными являются все детали, при выпуске которых не было допущено никаких дефектов, или эти дефекты были устранены.

Приведем алгоритм имитации производства деталей.

1. Имитируется обработка на токарном станке. Для этого разыгрывается СРРЧ  $R_1$ . Если  $R_1 < 0,08$ , то смоделирован брак (толщина детали оказалась меньше заданной) и выполняется переход к шагу 3. Если  $0,08 \leq R_1 < 0,2$ , то деталь требует исправления (толщина детали оказалась больше заданной). Если  $R_1 \geq 0,2$ , то на токарном станке выпущена деталь нормальной толщины.

2. Имитируется обработка на шлифовальном станке. Для этого разыгрываются два СРРЧ:  $R_{21}$  и  $R_{22}$ . Если  $R_{21} < 0,03$ , то допущен дефект верхней поверхности. Если  $R_{22} < 0,06$ , то допущен дефект нижней поверхности. Таким образом, если  $R_{21} < 0,03$  и  $R_{22} < 0,06$ , то деталь бракуется, так как имеются два дефекта. Если  $R_{21} < 0,03$  или  $R_{22} < 0,06$  (но не то и другое вместе), то имеется только один дефект; он устраняется, и выпускается *годная* деталь. Если  $R_{21} \geq 0,03$  и  $R_{22} \geq 0,06$ , то деталь не имеет дефектов, т.е. является годной.

Шаги 1-2 повторяются многократно, например, 100 000 раз. По окончании испытаний определяется вероятность выпуска годной детали как отношение количества годных деталей к общему количеству деталей (100 000).

Определяется также средняя прибыль от выпуска одной детали как отношение суммарной прибыли к общему количеству деталей.

Рассмотрим два испытания алгоритма.

*Первое испытание.* Пусть  $R_1=0,0795$ . Так как  $R_1 < 0,08$ , на токарном станке допущен брак: толщина детали оказалась меньше заданной. Деталь не поступает на дальнейшую обработку. Убыток составит  $20+5=25$  ден.ед., где 20 – стоимость материала, 5 – затраты на обработку на токарном станке.

*Второе испытание.* Пусть  $R_1=0,3780$ . Так как  $R_1 \geq 0,2$ , на токарном станке выпущена деталь нормальной толщины. Она поступает на обработку на шлифовальный станок. Разыгрываются СРРЧ  $R_{21}$  и  $R_{22}$ . Пусть  $R_{21}=0,0593$ ,  $R_{22}=0,7602$ . Так как  $R_{21} \geq 0,03$  и  $R_{22} \geq 0,06$ , на шлифовальном станке не допущено дефектов. Таким образом, выпущена годная деталь. Прибыль составит  $35-20-5-4=6$  ден.ед.

Приведем программу на языке VBA, реализующую алгоритм имитации выпуска деталей.

```
Sub primer23()  
' Имитация выпуска деталей, обрабатываемых на двух станках  
n = 100000  
pb11 = 0.08 : pb12 = 0.12  
pb21 = 0.03 : pb22 = 0.06  
material = 20 : obr1 = 5 : obr2 = 4  
zu1 = 3 : zu2 = 2 : cena = 35  
    For i = 1 To n  
        zatr = zatr + material + obr1  
        r1 = Rnd  
        If r1 < pb11 Then GoTo kon  
        If (r1 > pb11) And (r1 < pb11 + pb12) Then zatr = zatr + zu1  
        zatr = zatr + obr2  
        r21 = Rnd : r22 = Rnd  
        If (r21 < pb21) And (r22 < pb22) Then GoTo kon  
        If (r21 < pb21) Xor (r22 < pb22) Then zatr = zatr + zu2  
        prib = prib + cena  
        godn = godn + 1  
    kon: Next i  
    pgodn = godn / n  
    prib = prib - zatr  
    sredprib=prib/n  
    MsgBox ("Вероятность выпуска годной детали: " +str(round(pgodn,4)))  
    MsgBox ("Средняя прибыль от выпуска одной детали: "+str(round(sredprib,2)))  
End Sub
```

Результаты имитации будут примерно следующими: вероятность выпуска годной детали – примерно 0,9183 (т.е. процент годных деталей – 91,83%), средняя прибыль от выпуска одной деталей - примерно 2,94 ден.ед.

Чтобы определить, выгодна ли замена токарного станка на новый, достаточно заменить в программе для примера 2.3 следующие переменные:  $pb11=0,02$ ,  $pb12=0,03$ ,  $obr1=6$ . Результаты выполнения программы будут примерно следующими: вероятность выпуска годной детали – примерно

0,9779 (т.е. процент годных деталей – 97,79%), средняя прибыль от выпуска одной детали - около 4,05 ден.ед. Таким образом, замена станка является выгодной, так как она приводит к увеличению прибыли.

**Пример 2.4.** Предприятие выпускает электроприборы, состоящие из трех блоков (А,В,С). Прибор продолжает работать, пока исправен блок А и хотя бы один из блоков В или С. Из опыта эксплуатации приборов известно, что вероятность безотказной работы каждого из *блоков* в течение гарантийного срока составляет 90%.

Затраты предприятия на выпуск одного прибора следующие: стоимость каждого блока – 10 ден.ед., прочие расходы – 20 ден.ед. Приборы продаются по цене 65 ден.ед. В случае отказа прибора до окончания гарантийного срока предприятие бесплатно выполняет его ремонт. Затраты предприятия, связанные с гарантийным ремонтом отказавшего прибора, могут быть различными в зависимости от сложности ремонта, расходов на его доставку и т.д. Известно, что примерно в 60% случаев затраты на гарантийный ремонт составляют 25 ден.ед., в 30% случаев – 40 ден.ед., в 10% случаев – 50 ден.ед.

Требуется найти: а) вероятность отказа прибора до окончания гарантийного срока; б) среднюю прибыль предприятия от выпуска одного прибора; в) определить, выгодно ли для предприятия использовать в конструкции прибора новый, более надежный блок А, стоимость которого составляет 12 ден.ед., а вероятность безотказной работы в течение гарантийного срока – 98% (при этом повышение цены на приборы не планируется).

Для решения этой задачи разработаем имитационную модель на основе метода Монте-Карло.

Приведем алгоритм имитации выпуска и продажи приборов при использовании имеющейся конструкции прибора (т.е. без замены блока А на новый).

1. Имитируется работа прибора в течение гарантийного срока. Для этого разыгрываются три СРРЧ:  $R_A$ ,  $R_B$ ,  $R_C$ . Если выполняется условие  $R_A < 0,1$  (отказал блок А) или условия  $R_B < 0,1$  и  $R_C < 0,1$  (отказали блоки В и С), то смоделирован отказ прибора; в этом случае выполняется следующий шаг. В противном случае выполняется переход к шагу 3.

2. Имитируется гарантийный ремонт прибора. Для этого разыгрывается СРРЧ  $R_{rem}$ . Если  $R_{rem} < 0,6$ , то затраты на ремонт составляют 12 ден.ед., если  $0,6 \leq R_{rem} < 0,9$ , то затраты составляют 20 ден.ед, если  $R_{rem} \geq 0,9$ , то 35 ден.ед.

3. Вычисляется прибыль (или убыток) от выпуска прибора с учетом затрат на его изготовление, цены, а также затрат на гарантийный ремонт (если он потребовался).

Шаги 1-3 повторяются многократно, например, 100 000 раз. По окончании испытаний определяется вероятность отказа прибора до окончания гарантийного срока как отношение количества случаев отказов к общему количеству выпущенных приборов (100 000). Определяется также средняя прибыль от выпуска одного прибора как отношение суммарной прибыли к общему количеству приборов.

Рассмотрим два испытания алгоритма.

*Первое испытание.* Пусть  $R_A=0,0795$ ,  $R_B=0,3780$ ,  $R_C=0,0593$ . Так как  $R_A < 0,1$ ,  $R_B \geq 0,1$ ,  $R_C < 0,1$ , смоделирован отказ блока А (а также блока С). Таким образом, прибор отказал до окончания гарантийного срока. Разыгрываются затраты на его ремонт. Пусть  $R_{rem}=0,7602$ . Так как  $0,6 \leq R_{rem} < 0,9$ , затраты составляют 40 ден.ед. Таким образом, предприятие понесло *убыток* в размере  $65-30-20-40=-25$  ден.ед. (здесь 65 – цена прибора, 30 – стоимость блоков А,В,С, 20 – прочие расходы на выпуск прибора, 40 – затраты на гарантийный ремонт).

*Второе испытание.* Пусть  $R_A=0,7602$ ,  $R_B=0,2847$ ,  $R_C=0,6133$ . Так как  $R_A \geq 0,1$ ,  $R_B \geq 0,1$ ,  $R_C \geq 0,1$ , смоделирована нормальная работа прибора в течение всего гарантийного срока (ни один из блоков не отказал). Предприятие получает прибыль в размере  $65-30-20=15$  ден.ед.

Приведем программу на языке VBA, реализующую алгоритм имитации выпуска и гарантийного ремонта приборов.

```
Sub primer24()  
'Имитация выпуска приборов с учетом гарантийного ремонта  
n = 100000  
sta = 10: stb = 10: stc = 10: proch = 20: cena = 65  
pa = 0.1: pb = 0.1: pc = 0.1  
prem1 = 0.6: prem2 = 0.9  
zrem1 = 25: zrem2 = 40: zrem3 = 50  
For i = 1 To n
```

***sumprib = sumprib - sta - stb - stc - proch + cena***

```
ra = Rnd: rb = Rnd: rc = Rnd  
If (ra < pa) Or ((rb < pb) And (rc < pc)) Then  
otkaz = otkaz + 1  
rrem = Rnd  
If rrem < prem1 Then zrem = zrem1  
If (rrem >= prem1) And (rrem < prem2) Then zrem = zrem2  
If rrem >= prem2 Then zrem = zrem3  
sumprib = sumprib - zrem  
End If  
Next i  
potkaz = otkaz / n  
sredprib=sumprib/n  
MsgBox ("Вероятность отказа до окончания гарантийного срока: " + Str(Round(potkaz,4)))  
MsgBox ("Средняя прибыль от выпуска одного прибора: " + Str(Round(sredprib,2)))  
End Sub
```

Результаты имитации следующие: вероятность отказа прибора до окончания гарантийного срока – примерно 0,1095 (т.е. около 10,95%), средняя прибыль от выпуска одного прибора – около 11,5 ден.ед.

Чтобы оценить целесообразность использования более надежного (и более дорогого) блока А, достаточно выполнить имитацию снова, изменив в программе следующие переменные:  $sta=12$ ,  $pa=0,02$ . Результаты имитации следующие: вероятность отказа прибора до окончания гарантийного срока –

примерно 0,0298, средняя прибыль от выпуска одного прибора – около 12,04 ден.ед. Таким образом, использование нового блока А позволяет повысить прибыль за счет снижения затрат на гарантийные ремонты. Кроме того, резкое снижение количество отказов в течение гарантийного срока благоприятно скажется на репутации предприятия. Поэтому использование нового блока следует признать выгодным.

**Пример 2.5.** Предприятие выпускает датчики четырех типов (А,В,С,Д) для автоматизированных систем управления технологическими процессами. Из опыта работы предприятия известно, что примерно 30% всех заказов составляют заказы на датчики типа А, 20% - В, 15% - С, 35% - Д. Все датчики могут выпускаться в обычном исполнении (для работы в обычных условиях) или в специальном исполнении (для работы при высокой влажности, во взрывоопасной среде или при высокой температуре).

Известно, что примерно в 40% всех заказов требуется датчик для работы при высокой влажности, в 15% заказов - для работы во взрывоопасной среде, в 25% заказов – для работы при высокой температуре. При этом к одному датчику может предъявляться несколько дополнительных требований (например, может быть заказан датчик для работы при высокой влажности и температуре).

Затраты предприятия на выпуск одного датчика в обычном исполнении следующие: датчик типа А - 25 ден.ед., В – 15 ден.ед., С – 35 ден.ед., Д – 30 ден.ед. Дополнительные затраты предприятия при выпуске датчика для работы при высокой влажности составляют 8 ден.ед., во взрывоопасной среде – 12 ден.ед., при высокой температуре – 10 ден.ед. (эти затраты не зависят от типа датчика).

Датчики, выпущенные в обычном исполнении, продаются по следующим ценам: А – 45 ден.ед., В – 35 ден.ед., С – 60 ден.ед., Д – 50 ден.ед. За каждое дополнительное требование цена датчика повышается на 20% от исходной цены.

Например, если будет заказан датчик типа А для работы при высокой влажности и температуре, то затраты на его выпуск составят  $25+8+10=43$  ден.ед. Он будет продан по цене  $45+0,4\cdot 45=63$  ден.ед. (т.е. по цене, повышенной на 40%, так как при выпуске датчика выполнены два дополнительных требования). Прибыль от выпуска такого датчика составит  $63-43=20$  ден.ед.

Требуется разработать алгоритм имитации выпуска датчиков на основе метода Монте-Карло и реализовать его в виде программы. Определить: а) среднюю прибыль предприятия от выпуска одного датчика; б) долю датчиков специального исполнения (т.е. хотя бы с одним дополнительным требованием) в общем объеме заказов.

Приведем алгоритм имитации выпуска датчиков.

1. Имитируется тип датчика. Для этого разыгрывается СРРЧ  $R$ . Если  $R < 0,3$ , то заказан датчик типа А, если  $0,3 \leq R < 0,5$  – датчик типа В, если  $0,5 \leq R < 0,65$  – датчик типа С, если  $R \geq 0,65$  – датчик типа Д.

2. Имитируется исполнение датчика (обычное или специальное). Для этого разыгрываются три СРРЧ:  $R_1, R_2, R_3$ . Если  $R_1 < 0,4$ , то заказан датчик для работы при высокой влажности. Если  $R_2 < 0,15$ , то заказан датчик для работы во взрывоопасной среде. Если  $R_3 < 0,25$ , то заказан датчик для работы при высокой температуре. Если  $R_1 \geq 0,4, R_2 \geq 0,15$  и  $R_3 \geq 0,25$  (т.е. выполняются все три условия), то заказан датчик в обычном исполнении.

Шаги 1-2 повторяются 100 000 раз. Определяется средняя прибыль предприятия от выпуска одного датчика: отношение суммарной прибыли от выпуска датчиков к общему количеству выпущенных датчиков (100 000). Определяется также доля датчиков специального исполнения как отношение количества случаев, когда к датчику предъявлялось хотя бы одно дополнительное требование, к общему количеству выпущенных датчиков.

Рассмотрим два испытания алгоритма.

*Первое испытание.* Пусть  $R=0,0795$ . Так как  $R < 0,3$ , заказан датчик типа А. Пусть  $R_1=0,3780, R_2=0,0593, R_3=0,7602$ . Так как  $R_1 < 0,4, R_2 < 0,15, R_3 \geq 0,25$ , заказан датчик для работы при высокой влажности и во взрывоопасной среде. Затраты на его выпуск составят  $25+8+12=45$  ден.ед. Он будет продан по цене  $45+0,4 \cdot 45=63$  ден.ед. (т.е. по цене, повышенной на 40%, так как при выпуске датчика выполнены два дополнительных требования). Прибыль от выпуска датчика составит  $63-45=18$  ден.ед.

*Второе испытание.* Пусть  $R=0,2847$ . Так как  $R < 0,3$ , заказан датчик типа А. Пусть  $R_1=0,8197, R_2=0,6133, R_3=0,5766$ . Так как  $R_1 \geq 0,4, R_2 \geq 0,15, R_3 \geq 0,35$ , заказан датчик в обычном исполнении (т.е. к нему не предъявлено каких-либо дополнительных требований). Затраты на его выпуск составят 25 ден.ед., а цена – 45 ден.ед. Прибыль составит  $45-25=20$  ден.ед.

Приведем программу на языке VBA, реализующую алгоритм имитации выпуска датчиков.

```
sub primer25()
' Имитация выпуска датчиков обычного и специального исполнения
pa = 0.3: pb = 0.2: pc = 0.15: pd = 0.35
pvl = 0.4: pvz = 0.15: pt = 0.25
za = 25 : zb = 15: zc = 35: zd = 30
zvl = 8: zvz = 12: zt = 10
ca = 45: cb = 35: cc = 60: cd = 50
n = 100000
For i = 1 To n
r = Rnd
If r < pa Then sumzatr = sumzatr + za: cena = ca
If (r > pa) And (r < pa + pb) Then sumzatr = sumzatr + zb: cena = cb
If (r > pa + pb) And (r < pa + pb + pc) Then sumzatr = sumzatr + zc: cena = cc
If (r > pa + pb + pc) And (r < pa + pb + pc + pd) Then sumzatr = sumzatr + zd: cena = cd
sumvir = sumvir + cena
r1 = Rnd: r2 = Rnd: r3 = Rnd
If r1 < pvl Then sumzatr = sumzatr + zvl: sumvir = sumvir + 0.2 * cena
If r2 < pvz Then sumzatr = sumzatr + zvz: sumvir = sumvir + 0.2 * cena
If r3 < pt Then sumzatr = sumzatr + zt: sumvir = sumvir + 0.2 * cena
If (r1 < pvl) Or (r2 < pvz) Or (r3 < pt) Then spec = spec + 1

```

```

Next i
sumprib = sumvir – sumzatr
sredprib=sumprib/n
procspec = (spec / n) * 100
MsgBox ("Средняя прибыль от выпуска одного датчика: "+str(round(sredprib,2)))
MsgBox ("Процент спец.исполнения: "+str(round(procspec,2)))
End Sub

```

Результаты выполнения этой программы должны быть примерно следующими: средняя прибыль от выпуска одного датчика – 20,77 ден.ед., доля датчиков специального исполнения – 61,46%.

### 3. РЕШЕНИЕ ЗАДАЧ МОДЕЛИРОВАНИЯ НА ОСНОВЕ АЛГОРИТМОВ ИМИТАЦИИ СЛУЧАЙНЫХ ВЕЛИЧИН

#### 3.1. Имитация дискретных случайных величин

Под случайной величиной будем понимать величину, точное значение которой не может быть точно известно заранее.

**Дискретная случайная величина**  $X$  - это величина, которая может принимать некоторые фиксированные значения  $x_1, x_2, \dots, x_n$  с вероятностями  $P_1, P_2, \dots, P_n$ , где  $P_1 + P_2 + \dots + P_n = 1$ . Количество возможных значений дискретной случайной величины (число  $n$ ) может быть как конечным, так и бесконечным.

Если количество возможных значений дискретной случайной величины невелико, то ее имитация выполняется так же, как имитация полной группы событий (см. подраздел 2.1). Так, в примере 2.2 прибыль от ремонта одного изделия представляет собой дискретную случайную величину, которая может принимать значения 35 (с вероятностью 0,2), 60 (с вероятностью 0,4), 40 (с вероятностью 0,15) или 25 (с вероятностью 0,25).

Для имитации дискретных случайных величин, имеющих большое (или бесконечное) количество возможных значений, требуется разрабатывать специальные алгоритмы. Такие алгоритмы разрабатываются на основе методов теории вероятностей. Ниже приводятся алгоритмы имитации некоторых дискретных случайных величин, часто применяемых в практических задачах.

Имитация дискретной равномерной случайной величины

Дискретная равномерная случайная величина – это случайная величина, принимающая любые целочисленные значения на отрезке  $[A; B]$  (где  $A$  и  $B$  – целые числа) с равной вероятностью.

**Примечание.** Необходимо различать *дискретную равномерную случайную величину*, принимающую *только целочисленные значения* в некотором заданном диапазоне, и *непрерывную равномерную случайную величину*, принимающую *любые значения* в заданном диапазоне. Как правило, если говорится о “равномерной случайной величине”, то имеется в виду *непрерывная величина*.

Для имитации дискретной равномерной случайной величины применяется следующая формула:

$$X = [A + (B - A + 1) \cdot R], \quad (3.1)$$

где  $R$  – СРРЧ;

[ ] – выделение целой части числа.

**Пример 3.1.** Предприятие выпускает книжные полки по заказам. Для крупных заказов предусмотрена скидка: за изготовление одной полки предприятие берет с заказчика 25 ден.ед., если заказ составляет от одной до десяти полок, и 20 ден.ед. – если заказ превышает 10 полок. Затраты предприятия на выпуск одной полки составляют 12 ден.ед. Из опыта работы предприятия известно, что заказы обычно составляют от одной до 40 полок (заказы на изготовление свыше 40 полок крайне редки). Предлагается отменить скидку для крупных заказов, т.е. брать с заказчика 25 ден.ед. за каждую полку, независимо от размера заказа. Однако в этом случае можно ожидать, что количество крупных заказов уменьшится, и заказы будут составлять, как правило, от одной до 20 полок. Требуется определить, выгодна ли для предприятия отмена скидки.

Для решения этой задачи разработаем имитационную модель на основе метода Монте-Карло. Используя такую модель, определим среднюю прибыль предприятия от выполнения одного заказа.

Приведем алгоритм имитации работы предприятия при использовании скидки.

1. Разыгрывается размер заказа  $X$  (количество заказанных полок) по формуле (3.1) при  $A=1$ ,  $B=40$ .

2. Определяется прибыль предприятия от выполнения заказа. Если заказано не более десяти полок ( $X \leq 10$ ), то прибыль предприятия от выпуска одной полки будет составлять  $25-12=13$  ден.ед., а прибыль от выполнения заказа в целом –  $13X$ . Если заказано более десяти полок ( $X > 10$ ), то прибыль предприятия от выпуска одной полки будет составлять  $20-12=8$  ден.ед., а прибыль от выполнения заказа в целом –  $8X$ .

Шаги 1-2 повторяются многократно, например, 100 000 раз. Вычисляется средняя прибыль предприятия от выполнения одного заказа (как отношение суммарной прибыли к количеству заказов).

Рассмотрим два испытания алгоритма.

*Первое испытание.* Пусть при использовании формулы (3.1) разыграно СРРЧ  $R=0,0795$ . Тогда  $X=[1+(40-1+1) \cdot 0,0795]=4$ , т.е. смоделирован заказ на выпуск четырех полок. В этом случае предприятие берет с заказчика 25 ден.ед. за каждую полку. Затраты предприятия на изготовление каждой полки – 12 ден.ед. Таким образом, прибыль составит  $(25-12) \cdot 4=52$  ден.ед.

*Второе испытание.* Пусть разыграно СРРЧ  $R=0,378$ . Тогда  $X=[1+(40-1+1) \cdot 0,378]=16$ , т.е. смоделирован заказ на выпуск 16 полок. В этом случае предприятие берет с заказчика 20 ден.ед. за каждую полку. Прибыль составит  $(20-12) \cdot 16 = 128$  ден.ед.

Приведем программу на языке VBA, реализующую алгоритм имитации работы предприятия.

```
Sub primer31_1()
```

```
'Имитация работы с использованием скидок
```

```
n = 100000
```

```

min_zakaz = 1: max_zakaz = 40 : zatr = 12: plata1 = 25: plata2 = 20
For i = 1 To n
r = Rnd
x = Int(min_zakaz + (max_zakaz + 1 - min_zakaz) * Rnd)
If x <= 10 Then prib = (plata1 - zatr) * x Else prib = (plata2 - zatr) * x
sumprib = sumprib + prib
Next i
sredprib=sumprib/n
MsgBox ("Средняя прибыль от выполнения заказа: " + Str(Round(sredprib,2)))
End Sub

```

По результатам имитации средняя прибыль от выполнения одного заказа составит около 170,5 ден.ед.

Выполним имитацию работы предприятия без использования скидки. Для имитации размера заказа будет использоваться формула (3.1) при  $A=1$ ,  $B=20$ . Прибыль предприятия от выпуска одной полки будет всегда составлять  $25-12=13$  ден.ед., а прибыль от выполнения заказа -  $13X$ , где  $X$  – размер заказа. Программа для имитации работы предприятия без использования скидки имеет следующий вид.

```

Sub primer31_2()
'Имитация работы без скидок
n = 100000
min_zakaz = 1: max_zakaz = 20 : zatr = 12: plata = 25
For i = 1 To n
r = Rnd
x = Int(min_zakaz + (max_zakaz + 1 - min_zakaz) * Rnd)
prib = (plata - zatr) * x
sumprib = sumprib + prib
Next i
sredprib=sumprib/n
MsgBox ("Средняя прибыль от выполнения заказа: " + Str(Round(sredprib,2)))
End Sub

```

По результатам имитации, прибыль от выполнения одного заказа составит в этом случае около 136,55 ден.ед.

Таким образом, отмену скидки следует признать невыгодой, так как она приводит к снижению прибыли из-за сокращения размеров заказов.

#### **Имитация пуассоновской случайной величины**

Пуассоновская случайная величина – это величина, принимающая значения  $m=0, 1, 2, \dots$  с вероятностями, определяемыми по следующей формуле:

$$P_m = \frac{\lambda^m}{m!} e^{-\lambda}, \quad (3.2)$$

где  $\lambda$  – параметр распределения (т.е. некоторое число).

Таким образом, пуассоновская случайная величина может принимать любые неотрицательные целочисленные значения (от нуля до бесконечности).

Например, пуассоновскому закону распределения соответствует количество событий в пуассоновском потоке [15, 16], происходящих за некоторый интервал времени  $t$ . В этом случае  $\lambda=\Lambda t$ , где  $\Lambda$  – интенсивность потока событий (т.е. среднее количество событий, происходящих в единицу времени).

Для имитации пуассоновской случайной величины применяется следующий алгоритм.

1. Значение моделируемой величины принимается равным нулю:  $m=0$ .
2. Вычисляется значение вспомогательной переменной  $P_0$ :  $P_0 = e^{-\lambda}$ .
3. Вспомогательная переменная  $Q$  принимается равной единице:  $Q=1$ .
4. Разыгрывается СРРЧ  $R$ .
5. Вычисляется новое значение вспомогательной переменной  $Q$ :  $Q=Q \cdot R$ .
6. Если  $Q \geq P_0$ , то значение моделируемой величины увеличивается на единицу ( $m=m+1$ ), и выполняется возврат к шагу 4. Если  $Q < P_0$ , то алгоритм завершается. Смоделированное значение пуассоновской случайной величины равно  $m$ .

Приведем программную реализацию рассмотренного алгоритма в виде подпрограммы на языке VBA.

```
Sub puasson(l, m)
p0=exp(-l)
q=1 : m=0
povtor:
r=Rnd
q=q*r
If q>=p0 Then
m=m+1
Goto povtor
End If
End Sub
```

Здесь переменная  $l$  – параметр пуассоновского распределения. Эта величина должна задаваться при вызове подпрограммы.

Применение рассмотренного алгоритма имитации пуассоновской случайной величины показано в примере 3.9.

### 3.2. Имитация непрерывных случайных величин

Под **непрерывной случайной величиной** будем понимать случайную величину, которая может принимать любые значения в некотором диапазоне  $(a; b)$ . Этот диапазон может быть как ограниченным, так и неограниченным.

Для описания непрерывных случайных величин обычно используются две функции: функция распределения  $F(x)$  и плотность распределения (или плотность вероятности)  $f(x)$ . Смысл этих функций показан в прил. 3.

Принцип работы всех методов имитации непрерывных случайных величин следующий. Над СРРЧ  $R$  (т.е. случайными величинами, распределенными по равномерному закону в диапазоне от 0 до 1) выполняются определенные математические преобразования, позволяющие получить другие случайные величины с заданным законом распределения, т.е. с заданной функцией распределения  $F(x)$  или плотностью распределения  $f(x)$ .

### 3.2.1. Имитация непрерывных случайных величин на основе метода обратных функций

Метод обратных функций применяется, если известна функция распределения имитируемой непрерывной случайной величины  $F(x)$ , и можно найти обратную к ней функцию  $F^{-1}(R)$  (смысл аргумента  $R$  будет показан в описании метода).

Метод обратных функций основывается на следующем утверждении, доказываемом методами теории вероятностей. Пусть  $F(x)$  - функция распределения некоторой непрерывной случайной величины, а  $R$  - СРРЧ. Тогда, если записать уравнение  $F(X)=R$  и выразить из него  $X$ , то  $X$  будет представлять собой случайную величину, имеющую функцию распределения  $F(x)$ .

Построение алгоритма (т.е. вывод формул) для имитации непрерывных случайных величин методом обратных функций выполняется в следующем порядке.

1. Находится функция распределения случайной величины  $F(x)$ .
2. Записывается уравнение  $F(X)=R$ .
3. Из уравнения  $F(X)=R$  находится выражение для  $X$ :  $X=F^{-1}(R)$ . Оно представляет собой алгоритм для имитации непрерывной случайной величины  $X$ .

Примечание. В некоторых случаях функция  $F^{-1}(R)$  оказывается неоднозначной (например, если в выражении  $F(X)$  имеется квадрат). В таких случаях требуется проверить все возможные варианты  $F^{-1}(R)$  и выбрать тот из них, при котором имитируемая случайная величина  $X$  принимает значения в заданном диапазоне  $(a; b)$ .

Примечание. Применение метода обратных функций в некоторых случаях оказывается сложным, так как не для всех непрерывных случайных величин можно аналитически (т.е. в виде формулы) записать функцию распределения  $F(x)$ ; в других случаях, даже если функция  $F(x)$  известна, оказывается невозможным записать обратную к ней функцию. Пример такого случая рассматривается в п.3.2.3. В таких случаях применяются другие методы имитации непрерывных случайных величин, или для реализации метода обратных функций используются численные методы.

**Пример 3.2.** При изготовлении пластмассовых плит выполняются две операции: внесение в пластмассу упрочняющих добавок и охлаждение плиты в форме. Из наблюдений за производственным процессом известно, что время внесения добавок – случайная величина, плотность распределения которой может быть задана графиком, приведенным на рис.3.1. Время охлаждения плиты в форме – случайная величина, которую можно считать распределенной по экспоненциальному закону; в среднем охлаждение плиты занимает 30 минут.

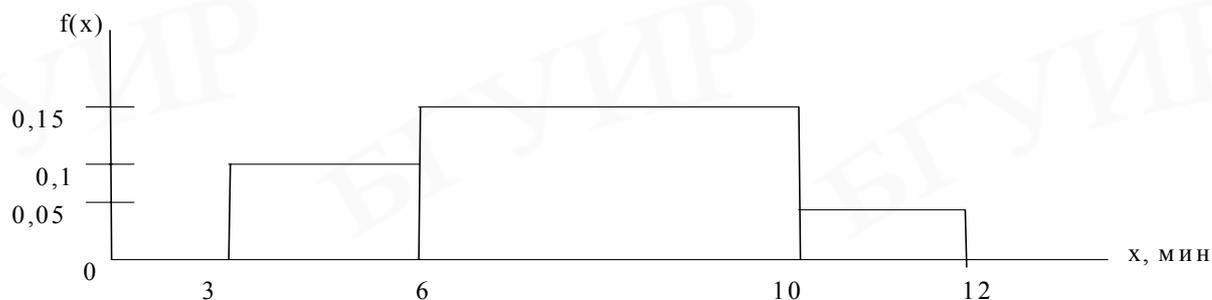


Рис.3.1. Плотность распределения времени внесения добавок (для примера 3.2)

Требуется разработать алгоритмы имитации случайных величин, используемых в задаче, и программную реализацию этих алгоритмов.

Примечание. Применение разработанных алгоритмов имитации будет показано в примере 3.6.

В данной задаче используются две случайные величины: время внесения добавок и время охлаждения плиты.

Рассмотрим разработку *алгоритма имитации времени внесения добавок* на основе метода обратных функций.

По графику, приведенному на рис.3.1, запишем уравнение плотности распределения времени внесения добавок:

$$f(x) = \begin{cases} 0, & x < 3; \\ 0,1, & 3 \leq x < 6; \\ 0,15, & 6 \leq x < 10; \\ 0,05, & 10 \leq x < 12; \\ 0, & x \geq 12. \end{cases} \quad (3.3)$$

Используя формулу (П3.3), находим выражение для функции распределения:

- на интервале  $x < 3$ :  $\int_{-\infty}^x 0 dx = 0$ ;
- на интервале  $3 \leq x < 6$ :  $\int_{-\infty}^3 0 dx + \int_3^x 0,1 dx = 0,1x - 0,3$ ;
- на интервале  $6 \leq x < 10$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^x 0,15 dx = 0,15x - 0,6$ ;
- на интервале  $10 \leq x < 12$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^{10} 0,15 dx + \int_{10}^x 0,05 dx = 0,05x + 0,4$ ;
- на интервале  $x \geq 12$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^{10} 0,15 dx + \int_{10}^{12} 0,05 dx + \int_{12}^x 0 dx = 1$ .

Таким образом, функция распределения времени внесения добавок имеет следующий вид:

$$F(x) = \begin{cases} 0, & x < 3; \\ 0,1x - 0,3, & 3 \leq x < 6; \\ 0,15x - 0,6, & 6 \leq x < 10; \\ 0,05x + 0,4, & 10 \leq x < 12; \\ 1, & x \geq 12. \end{cases} \quad (3.4)$$

Записываем уравнения  $F(X)=R$ :

$$\begin{aligned} 0,1X - 0,3 &= R, & 3 \leq X < 6; \\ 0,15X - 0,6 &= R, & 6 \leq X < 10; \\ 0,05X + 0,4 &= R, & 10 \leq X < 12. \end{aligned}$$

Из уравнений  $F(X)=R$  находим выражение  $X = F^{-1}(R)$ , т.е. формулу (алгоритм) для имитации случайной величины  $X$ :

$$X = \begin{cases} (R + 0,3) / 0,1, & 0 \leq R < 0,3; \\ (R + 0,6) / 0,15, & 0,3 \leq R < 0,9; \\ (R - 0,4) / 0,05, & 0,9 \leq R < 1. \end{cases} \quad (3.5)$$

Границы диапазонов значений  $R$  получены подстановкой соответствующих значений  $X$  в уравнение  $F(X)=R$ . Например, при  $3 \leq X < 6$  уравнение  $F(X)=R$  имеет следующий вид:  $0,1X - 0,3 = R$ . Подставляя в это уравнение значения границ диапазона  $X$  (т.е. 3 и 6), получим границы диапазона  $R$ : 0 и 0,3.

Выполним одно испытание полученного алгоритма. Пусть разыграно значение СРРЧ  $R=0,8197$ . Тогда  $X=(0,8197+0,6)/0,15=9,465$ . Таким образом, в данном испытании смоделировано значение времени внесения добавок, равное 9,465 мин.

Приведем программную реализацию разработанного алгоритма в виде подпрограммы на языке VBA.

```
sub vnesdob(x)
r=rnd
if r<0.3 then x=(r+0.3)/0.1
if (r>=0.3)and(r<0.9) then x=(r+0.6)/0.15
if r>=0.9 then x=(r-0.4)/0.05
end sub
```

Рассмотрим получение алгоритма имитации времени охлаждения плит на основе метода обратных функций. Согласно постановке задачи, можно считать, что эта величина распределена по экспоненциальному закону с математическим ожиданием (средним значением) 30 мин. Формула плотности распределения для случайной величины, распределенной по экспоненциальному закону, известна из теории вероятностей:

$$f(x) = \begin{cases} 0, & x < 0; \\ \lambda e^{-\lambda x}, & x \geq 0, \end{cases} \quad (3.6)$$

где  $\lambda$  – параметр распределения;  $\lambda=1/\bar{X}$ , где  $\bar{X}$  - математическое ожидание (среднее значение) экспоненциальной случайной величины.

Таким образом, в данном примере  $\lambda=1/30$ .

Случайная величина, распределенная по экспоненциальному закону, может принимать (теоретически) любые неотрицательные значения.

По формуле (ПЗ.1) найдем выражение для функции распределения экспоненциальной случайной величины:

- на интервале  $x < 0$ :  $\int_{-\infty}^x 0 dx = 0$ ;
- на интервале  $x \geq 0$ :  $\int_{-\infty}^0 0 dx + \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}$ .

Таким образом, функция распределения экспоненциальной случайной величины имеет следующий вид:

$$F(x) = \begin{cases} 0, & x < 0; \\ 1 - e^{-\lambda x}, & x \geq 0. \end{cases} \quad (3.7)$$

Записываем уравнение  $F(X)=R$ :

$$1 - e^{-\lambda X} = R.$$

Из уравнения  $F(X)=R$  выражаем  $X$ , т.е. находим формулу (алгоритм) для имитации экспоненциальной случайной величины  $X$ :

$$X = -\frac{1}{\lambda} \ln(1 - R). \quad (3.8)$$

Так как величина  $1-R$ , как и  $R$ , представляет собой СРРЧ, для имитации экспоненциальной случайной величины можно использовать следующую формулу:

$$X = -\frac{1}{\lambda} \ln R. \quad (3.9)$$

Так как  $\lambda=1/\bar{X}$ , эту формулу можно записать по-другому:

$$X = -\bar{X} \ln R. \quad (3.10)$$

Для данного примера формула для имитации времени охлаждения плиты будет иметь следующий вид:  $X=-30 \cdot \ln R$ .

Выполним одно испытание полученного алгоритма. Пусть разыграно значение СРРЧ  $R=0,0795$ . Тогда  $X=-30 \cdot \ln 0,0795 = 75,96$ . Таким образом, в данном испытании смоделировано значение времени охлаждения плиты, равное 75,96 мин.

Приведем программную реализацию алгоритма имитации экспоненциальной случайной величины в виде подпрограммы на языке VBA.

```
sub expon(xsred,x)
r=rnd
x=-xsred*log(r)
end sub
```

Здесь  $x_{sred}$  – среднее значение имитируемой случайной величины. Эта величина должна задаваться при вызове подпрограммы. Для данного примера эту подпрограмму следует вызывать из основной программы следующим оператором:

```
call expon(30,x)
```

Приведенную подпрограмму можно использовать для имитации любой случайной величины, распределенной по экспоненциальному закону (а не только времени охлаждения плиты в рассматриваемом примере). Для этого достаточно задать значение переменной  $x_{sred}$ , равное математическому ожиданию имитируемой величины.

Таким образом, алгоритмы для имитации случайных величин, используемых в примере 3.2, получены. Их применение будет показано в примере 3.6.

Приведем (без вывода) алгоритмы имитации некоторых непрерывных случайных величин, часто применяемых в задачах моделирования. Эти алгоритмы также могут быть получены на основе метода обратных функций.

#### **Равномерное распределение:**

$$X = a + (b - a)R, \quad (3.11)$$

где  $a$  и  $b$  – границы диапазона возможных значений равномерной случайной величины.

#### **Распределение Эрланга $k$ -го порядка:**

$$X = -\frac{1}{k\lambda} \sum_{i=1}^k R_i. \quad (3.12)$$

где  $\lambda$  – параметр распределения.

Следует обратить внимание, что для имитации каждого значения случайной величины, распределенной по закону Эрланга  $k$ -го порядка, требуется использовать не одно, а  $k$  СРРЧ.

#### **Треугольное распределение:**

$$X = \begin{cases} a + \sqrt{R(b-a)(c-a)}, & 0 \leq R < (c-a)/(b-a); \\ b - \sqrt{(1-R)(b-a)(b-c)}, & (c-a)/(b-a) \leq R < 1, \end{cases} \quad (3.13)$$

где  $a$  и  $b$  – границы диапазона возможных значений треугольной случайной величины;

$c$  – мода треугольной случайной величины (т.е. точка максимума плотности распределения этой величины).

### **3.2.2. Метод исключения**

Этот метод применяется для имитации непрерывных случайных величин, принимающих значения в некотором ограниченном диапазоне  $(a; b)$ . Для применения метода должна быть известна плотность распределения

непрерывной случайной величины  $f(x)$ , а также максимальное значение плотности распределения  $f_{\max}$ .

Метод исключения основывается на следующем утверждении (теорема Неймана): если взять два СРРЧ  $R_1$  и  $R_2$  и с их помощью получить случайные числа  $R' = a + (b - a) \cdot R_1$  и  $R'' = f_{\max} \cdot R_2$ , то, если выполняется условие  $R'' \leq f(R')$ , случайное число  $X = R'$  имеет заданную плотность распределения  $f(x)$ .

Алгоритм имитации непрерывной случайной величины с помощью метода исключения реализуется в следующем порядке.

1. Разыгрываются два СРРЧ:  $R_1$  и  $R_2$ .
2. Вычисляются два случайных числа:  $R' = a + (b - a) \cdot R_1$  и  $R'' = f_{\max} \cdot R_2$ .
3. Проверяется условие:  $R'' \leq f(R')$ . Если это условие выполняется, то определяется значение имитируемой случайной величины:  $X = R'$ . Если условие  $R'' \leq f(R')$  не выполняется, то происходит возврат к шагу 1.

**Пример 3.3.** Используя метод исключения, для примера 3.2 разработать алгоритм имитации времени внесения добавок.

Время внесения добавок может составлять от 3 до 12 минут. Максимальное значение плотности распределения известно; оно равно 0,15. Таким образом, для имитации времени внесения добавок можно использовать алгоритм, приведенный выше, при  $a=3$ ,  $b=12$ ,  $f_{\max}=0,15$ .

Приведем два испытания алгоритма.

*Первое испытание.* Разыгрываются два СРРЧ. Пусть разыграны  $R_1=0,0795$ ,  $R_2=0,3780$ . Тогда  $R'=3+(12-3) \cdot 0,0795=3,7155$ ,  $R''=0,15 \cdot 0,3780 = 0,0567$ . Вычислим  $f(R')$ :  $f(3,7155)=0,1$ . Условие  $R'' \leq f(R')$  выполняется. Значит, определяется значение имитируемой случайной величины:  $X = R' = 3,7155$ . Таким образом, в данном испытании смоделировано значение времени внесения добавок, равное 3,7155 мин.

*Второе испытание.* Разыгрываются два СРРЧ. Пусть разыграны  $R_1=0,0593$ ,  $R_2=0,7602$ . Тогда  $R'=3+(12-3) \cdot 0,0593=3,5337$ ,  $R''=0,15 \cdot 0,7602 = 0,11403$ . Вычислим  $f(R')$ :  $f(3,5337)=0,1$ . Условие  $R'' \leq f(R')$  не выполняется. Таким образом, в данном испытании значение имитируемой случайной величины не получено.

Приведем программную реализацию алгоритма имитации времени внесения добавок в виде подпрограммы на языке VBA.

```
Sub iskl()  
nach: r1 = Rnd: r2 = Rnd  
r1s = 3 + 9 * r1 : r2s = 0.15*r2  
If (r1s >= 3) And (r1s <= 6) Then fr = 0.1  
If (r1s > 6) And (r1s <= 10) Then fr = 0.15  
If (r1s > 10) And (r1s <= 12) Then fr = 0.05  
If r2s <= fr Then x = r1s Else GoTo nach
```

Следует обратить внимание, что применение метода исключения возможно не всегда. Например, этим методом нельзя воспользоваться для имитации случайной величины, распределенной по экспоненциальному закону, так как такая величина может принимать любые неотрицательные значения, а

метод исключения может применяться только для имитации величин, принимающих значения в некотором *ограниченном* диапазоне.

### 3.2.3. Имитация гауссовского распределения

Случайная величина имеет гауссовское (нормальное) распределение, если ее плотность распределения задается следующим выражением:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \quad (3.14)$$

где  $m$  - математическое ожидание гауссовской случайной величины, а  $\sigma$  - ее среднее квадратическое отклонение (стандартное отклонение).

Гауссовская случайная величина теоретически может принимать любые значения (от  $-\infty$  до  $+\infty$ ). Однако с вероятностью 0,997 такая величина принимает значения в диапазоне  $(m-3\sigma; m+3\sigma)$ ; это свойство получило название "правила трех сигм". Поэтому гауссовская случайная величина может использоваться для имитации любой величины, о которой известно, что она обычно принимает значения, близкие к некоторому среднему значению.

Использование метода обратных функций для имитации гауссовской случайной величины затруднено, так как для такой величины невозможно записать формулу функции распределения: интеграл от плотности распределения (3.14) - неберущийся.

Алгоритм имитации гауссовской случайной величины выведен на основе следующего утверждения, доказываемого методами теории вероятностей: сумма  $k$  СРРЧ  $R$  (при достаточно большом  $k$ ) имеет распределение, близкое к гауссовскому, с математическим ожиданием  $k/2$  и стандартным отклонением  $\sqrt{k/12}$ . Из этого утверждения может быть выведена следующая формула для имитации гауссовской случайной величины с математическим ожиданием  $m$  и стандартным отклонением  $\sigma$ :

$$X = m + \sigma \sqrt{\frac{12}{k}} \left( \sum_{i=1}^k R_i - \frac{k}{2} \right), \quad (3.15)$$

где  $k$  - количество используемых при имитации СРРЧ.

Чем больше СРРЧ  $R$  суммируется (чем больше  $k$ ), тем выше точность имитации, т.е. тем ближе распределение величины  $X$  к гауссовскому распределению. Для практических задач достаточная точность достигается при  $k=6$ . В этом случае формулу (3.15) можно записать в следующем виде:

$$X = m + \sigma \sqrt{2} \left( \sum_{i=1}^6 R_i - 3 \right). \quad (3.16)$$

Как видно, при использовании формулы (3.16) для имитации *каждого* значения гауссовской случайной величины требуется разыгрывать *шесть* СРРЧ  $R$ .

Приведем программную реализацию алгоритма имитации гауссовской случайной величины по формуле (3.16).

```

Sub gauss(m, sigma, x)
k = 6 : rsum=0
For i = 1 To k
r = Rnd
rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub

```

Здесь переменная  $m$  – математическое ожидание (среднее значение) имитируемой случайной величины,  $\sigma$  – ее стандартное отклонение. Примеры использования этой подпрограммы приведены в подразд.3.3.

Для имитации гауссовской случайной величины может также применяться метод исключения. В качестве диапазона значений случайной величины в этом случае используется диапазон  $(m-3\sigma; m+3\sigma)$ , так как, по правилу “трех сигм”, почти все значения гауссовской случайной величины попадают в этот диапазон. Максимальное значение плотности распределения (3.14) следующее:  $f_{\max} = 1/(\sigma\sqrt{2\pi})$ . Таким образом, для имитации гауссовской случайной величины может использоваться алгоритм, приведенный в п.3.2.2, при  $a=m-3\sigma, b=m+3\sigma, f_{\max} = 1/(\sigma\sqrt{2\pi})$ .

### 3.2.4. Проверка алгоритмов имитации случайных величин

Проверка алгоритма имитации случайной величины состоит в проверке соответствия величин, получаемых при использовании этого алгоритма, заданному закону распределения моделируемой величины. Если распределение величин, получаемых при использовании алгоритма имитации, близко к заданному закону распределения, то можно считать, что алгоритм имитации правильный.

Рассмотрим два способа проверки алгоритмов имитации случайных величин: с помощью гистограммы и на основе критерия “хи-квадрат”.

#### *Проверка алгоритмов имитации на основе гистограммы*

Гистограмма распределения представляет собой график (в виде столбиков), отражающий частоту попаданий случайной величины в различные интервалы значений. Гистограмма строится в следующем порядке.

1. Находятся границы диапазона возможных значений случайной величины:  $X_{\min}, X_{\max}$ . Если эти границы точно не известны, то они задаются приближенно и могут быть скорректированы по результатам моделирования.

2. Диапазон значений моделируемой величины разбивается на  $k$  интервалов. Количество интервалов  $k$  обычно составляет от 5 до 25. Оно, как правило, подбирается таким образом, чтобы длина интервала  $d=(X_{\max}-X_{\min})/k$ , а также границы интервалов представляли собой “удобные” числа (например, целые).

3. Выполняется большое количество ( $n$ ) испытаний алгоритма имитации. При этом подсчитываются частоты попаданий имитируемой случайной величины в каждый из интервалов:  $m_i, i=1, \dots, k, m_1+m_2+\dots+m_k=n$ .

4. Вычисляются относительные частоты попаданий имитируемой случайной величины в каждый из интервалов:  $f_i = m_i / (n \cdot d)$ ,  $i=1, \dots, k$ .

Сама гистограмма строится следующим образом: по оси абсцисс откладываются интервалы значений случайной величины (заданные на шаге 2), и на каждом интервале строится столбик (прямоугольник) высотой  $f_i$ .

Если алгоритм имитации случайной величины разработан правильно, то гистограмма должна быть близка к *графику плотности распределения* имитируемой случайной величины.

**Пример 3.4.** Требуется проверить алгоритмы имитации случайных величин, разработанные в примере 3.2, используя гистограммы.

Выполним **проверку алгоритма имитации времени внесения добавок**. Эта случайная величина задана плотностью распределения, график которой приведен на рис.3.1. Время внесения добавок может принимать значения от 3 до 12 минут; таким образом,  $X_{\min}=3$ ,  $X_{\max}=12$ . Разобьем этот диапазон на  $k=9$  интервалов: (3; 4), (4; 5), ..., (11; 12). Длина интервала  $d=1$ .

Приведем программу на языке VBA для получения данных, необходимых для построения гистограммы. В этой программе выполняется большое количество (1000) испытаний алгоритма имитации времени внесения добавок, полученного в п.3.2.1 (подпрограмма vnesdob). Подсчитываются частоты попаданий имитируемой величины в каждый из интервалов и вычисляются относительные частоты. Границы интервалов заданы в массиве gran. Частоты попаданий в интервалы подсчитываются в массиве m. Относительные частоты вычисляются в массиве f.

```
Sub primer33_1()
Dim gran(1 To 10), f(1 To 10), m(1 To 10)
gran(1) = 3: gran(2) = 4: gran(3) = 5: gran(4) = 6: gran(5) = 7: gran(6) = 8
gran(7) = 9: gran(8) = 10: gran(9) = 11: gran(10) = 12
k = 9: n = 1000: d = 1
For i = 1 To n
call vnesdob(x)
For j = 1 To k
If (x > gran(j)) And (x <= gran(j + 1)) Then m(j) = m(j) + 1
Next j
Next i
MsgBox ("Частоты")
For i = 1 To k
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(m(i)))
Next i
MsgBox ("Относительные частоты")
For i = 1 To k
f(i) = m(i) / (n * d)
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(Round(f(i),3)))
Next i
End Sub

Sub vnesdob(x)
r=Rnd
if r<0.3 then x=(r+0.3)/0.1
if (r>=0.3)and(r<0.9) then x=(r+0.6)/0.15
if r>=0.9 then x=(r-0.4)/0.05
End Sub
```

Таблица 3.1

Интервал	Частота, $m_i$	Относительная частота, $f_i$	<b>Интервал</b>	Частота, $m_i$	Относительная частота, $f_i$
3 – 4	93	0,093	8 – 9	137	0,137
4 – 5	90	0,09	9 – 10	140	0,140
5 – 6	113	0,113	10 – 11	38	0,038
6 – 7	175	0,175	11 – 12	53	0,053
7 - 8	161	0,161			

На рис.3.2,а приведена гистограмма, построенная по значениям относительных частот. Видно, что гистограмма близка к графику плотности распределения имитируемой величины, приведенному на рис.3.1. Таким образом, можно предполагать, что алгоритм имитации правильный. Более строгая проверка алгоритма (с использованием критерия “хи-квадрат”) будет рассмотрена ниже.

Аналогично выполним **проверку алгоритма имитации времени охлаждения плиты**. Эта случайная величина распределена по экспоненциальному закону с математическим ожиданием 30 мин. Экспоненциальная случайная величина может принимать (теоретически) любые неотрицательные значения. Поэтому точно задать для нее верхнюю границу ( $X_{max}$ ) невозможно. Попробуем использовать в качестве верхней границы значение 120. Таким образом,  $X_{min}=0$ ,  $X_{max}=120$ . Разобьем этот диапазон на  $k=12$  интервалов: (0; 10), (10; 20),..., (110; 120). Длина интервала  $d=10$ .

Внесем необходимые изменения в программу для построения гистограммы. Имитация времени охлаждения плиты будет выполняться с помощью подпрограммы expon, приведенной в п.3.2.1. Границы интервалов задаются в массиве gran. Требуется также изменить переменные  $k$  (количество интервалов) и  $d$  (длина интервала). Кроме того, так как верхняя граница (120) задана приближенно, следует предусмотреть подсчет значений, превышающих эту границу; для этого используется переменная zaverhgran.

```

Sub primer33_2()
Dim gran(1 To 13), f(1 To 13), m(1 To 13)
gran(1) = 0: gran(2) = 10: gran(3) = 20: gran(4) = 30: gran(5) = 40: gran(6) = 50 : gran(7) = 60
gran(8) = 70: gran(9) = 80: gran(10) = 90: gran(11) = 100: gran(12) = 110 : gran(13) = 120
k = 12: n = 1000: d = 10
For i = 1 To n
Call expon(30, x)

```

**For j = 1 To k**

```

If (x > gran(j)) And (x <= gran(j + 1)) Then m(j) = m(j) + 1
Next j
If (x > gran(k+1)) Then zaverhgran= zaverhgran + 1
Next i
MsgBox (“Частоты”)
For i = 1 To k

```

```

MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(m(i)))
Next i
MsgBox ("Относительные частоты")
For i = 1 To k
f(i) = m(i) / (n * d)
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(Round(f(i),4)))
Next I
MsgBox ("Превышают верхнюю границу: " + Str(zaverhgran))
End Sub

Sub expon(xsred, x)
r = Rnd
x = -xsred * Log(r)
End Sub

```

Результаты выполнения программы приведены в табл.3.2.

Таблица 3.2

Интервал	Частота, $m_i$	Относительная частота, $f_i$	<b>Интервал</b>	Частота, $m_i$	Относительная частота, $f_i$
0 – 10	285	0,0285	70 - 80	30	0,0030
10 – 20	192	0,0192	80 – 90	27	0,0027
20 – 30	149	0,0149	90 – 100	13	0,0013
30 – 40	108	0,0108	100 – 110	9	0,0009
40 – 50	88	0,0088	110 - 120	2	0,0002
50 – 60	55	0,0055	Более 120	13	-
60 – 70	29	0,0029			

Как видно, почти все значения, полученные при моделировании, попадают в диапазон (0; 120); за пределами этого диапазона оказалось только 13 значений. Поэтому можно использовать полученные результаты для построения гистограммы (рис.3.2,б).

Примечание. Если бы количество значений, выходящих за указанный диапазон ( $X_{\min}$ ;  $X_{\max}$ ), оказалось большим, то потребовалось бы расширить этот диапазон (в данном случае – увеличить верхнюю границу).

Гистограмма приведена на рис.3.2,б. Видно, что она близка к графику плотности распределения экспоненциальной случайной величины. Таким образом, можно предполагать, что алгоритм имитации времени охлаждения плиты также правильный.

Примечание. Основное назначение гистограммы распределения состоит не в проверке правильности имитации случайной величины, а в наглядном представлении значений случайной величины и частот их появления.

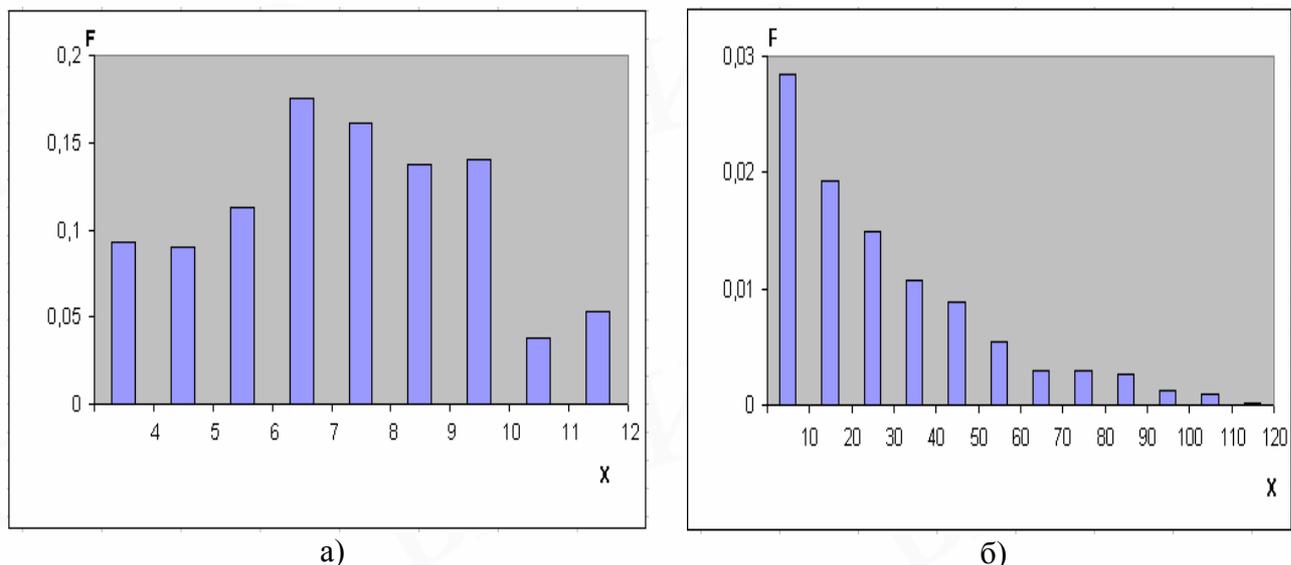


Рис.3.2. Гистограммы для проверки алгоритмов имитации, полученных в примере 3.2: а) для времени внесения добавок; б) для времени охлаждения плиты.

### **Проверка алгоритмов имитации на основе критерия “хи-квадрат”**

Критерий “хи-квадрат” ( $\chi^2$ ) представляет собой один из методов математической статистики. Одна из возможностей его применения состоит в следующем. Пусть имеется набор значений некоторой случайной величины. Критерий “хи-квадрат” позволяет проверить гипотезу о том, что случайная величина, для которой получен этот набор, распределена по некоторому конкретному закону. Таким образом, критерий “хи-квадрат” позволяет проверить, соответствуют ли значения случайной величины, полученные в ходе имитации, тому закону распределения, который требовалось имитировать.

Примечание. “Хи-квадрат” - это название одного из распределений непрерывных случайных величин, используемого при проверке по данному критерию.

Проверка соответствия случайной величины заданному закону распределения на основе критерия “хи-квадрат” выполняется в следующем порядке.

1. Определяется диапазон значений случайной величины ( $X_{\min}; X_{\max}$ ). Этот диапазон разбивается на  $k$  интервалов и подсчитывается количество попаданий имитируемой случайной величины в каждый из этих интервалов (частоты  $m_i, i=1, \dots, k$ ). См. алгоритм построения гистограммы (п.3.2.4), шаги 1-3.

2. По заданному закону распределения имитируемой случайной величины вычисляются *теоретические* вероятности ее попадания в каждый из интервалов:  $P_i, i=1, \dots, k$ . Для этого может использоваться, например, формула (П3.4) или (П3.5).

3. Вычисляется значение критерия “хи-квадрат” по формуле

$$\chi^2 = \sum_{i=1}^k \frac{m_i^2}{n \cdot P_i} - n, \quad (3.17)$$

где  $n$  - количество значений случайной величины (количество испытаний).

4. По таблицам распределения “хи-квадрат” определяется квантиль этого распределения  $\chi_{\alpha, s}^2$ . Здесь  $\alpha$  - уровень значимости. Величина  $\alpha$ , как правило,

задается из диапазона от 0,01 до 0,1. Чем меньшей задается величина  $\alpha$ , тем более строгой является проверка. Обычно используется значение  $\alpha=0,05$ . Величина  $s$  – число степеней свободы распределения “хи-квадрат”. Эта величина определяется следующим образом:  $s=k-1$ .

Квантиль  $\chi_{\alpha,s}^2$  можно также найти с помощью табличного процессора Excel, используя функцию ХИ2ОБР с аргументами Вероятность –  $\alpha$ , Степени свободы –  $s$ .

5. Значение критерия “хи-квадрат” сравнивается с квантилем  $\chi_{\alpha,s}^2$ . Если  $\chi^2 < \chi_{\alpha,s}^2$ , то можно утверждать, что значения случайной величины, полученные при моделировании, соответствуют заданному закону распределения. В этом случае можно считать, что алгоритм имитации правильный. Если условие  $\chi^2 < \chi_{\alpha,s}^2$  не выполняется, это означает, что значения, полученные при моделировании, существенно отклоняются от заданного закона распределения. Это может указывать на ошибку в алгоритме имитации.

**Пример 3.5.** Требуется проверить алгоритмы имитации случайных величин, разработанные в примере 3.2, используя критерий “хи-квадрат”.

**Выполним** проверку алгоритма имитации времени внесения добавок.

**Частоты  $m_i$  получены ранее (см. табл.3.1). Используя плотность распределения (3.3) или функцию распределения (3.4), указанные в п.3.2.1, найдем теоретические вероятности попадания времени внесения добавок в интервалы (3; 4), (4; 5),..., (11; 12). Для этого воспользуемся формулой (П3.4) или (П3.5). Например, используя формулу (П3.5), вероятность попадания в интервал (3; 4) можно найти так:**

$$P_1 = \int_3^4 0,1 dx = 0,1.$$

Аналогично вычисляются остальные вероятности  $P_2, \dots, P_9$ . Они приведены в табл. 3.3.

Таблица 3.3

Интервал	3 – 4	4 – 5	5 – 6	6 – 7	7 – 8	8 – 9	9 – 10	10 – 11	11 – 12
Вероятность, $P_i$	0,1	0,1	0,1	0,15	0,15	0,15	0,15	0,05	0,05

Найдем значение критерия “хи-квадрат” по формуле (3.17):

$$\chi^2 = \frac{93^2}{1000 \cdot 0,1} + \frac{90^2}{1000 \cdot 0,1} + \dots + \frac{53^2}{1000 \cdot 0,05} - 1000 = 13,01.$$

Найдем квантиль распределения “хи-квадрат”  $\chi_{\alpha,s}^2$ . Зададим уровень значимости  $\alpha=0,05$ . Число степеней свободы определяется следующим образом:  $s=k-1=9-1=8$ . Квантиль  $\chi_{0,05;8}^2$  можно найти из таблиц распределения “хи-квадрат” или с помощью табличного процессора Excel, используя функцию ХИ2ОБР с аргументами Вероятность – 0,05, Степени свободы – 8. В данном

примере  $\chi^2_{\alpha,s} = \chi^2_{0,05;8} = 15,507$ . Так как  $\chi^2 < \chi^2_{\alpha,s}$ , можно утверждать, что случайные величины, получаемые с помощью алгоритма имитации времени внесения добавок (и вычисляемые с помощью подпрограммы vnesdob), соответствуют плотности распределения (3.3). Таким образом, алгоритм имитации времени внесения добавок можно признать правильным.

**Выполним** проверку алгоритма имитации времени охлаждения плиты.

**Частоты  $m_i$  приведены в табл.3.2. Используя плотность распределения (3.6) или функцию распределения (3.7), найдем теоретические вероятности попадания времени охлаждения плиты в интервалы (0; 10), (10; 20),..., (110; 120). Например, вероятность попадания в интервал (0; 10) можно найти по формуле (П3.5) следующим образом:**

$$P_1 = \int_0^{10} \frac{1}{30} e^{-\frac{1}{30}x} dx = 0,283.$$

Аналогично вычисляются остальные вероятности:  $P_2, \dots, P_{12}$ . Они приведены в табл. 3.4.

Таблица 3.4

Интервал	0 - 10	10 - 20	20 - 30	30 - 40	40 - 50	50 - 60	60 - 70	70- 80	80- 90	90- 100	100- 110	110- 120
Вероятно сть, $P_i$	0,283	0,203	0,146	0,104	0,075	0,054	0,038	0,027	0,020	0,014	0,010	0,007

Найдем значение критерия “хи-квадрат” по формуле (3.17):

$$\chi^2 = \frac{285^2}{1000 \cdot 0,283} + \frac{192^2}{1000 \cdot 0,203} + \dots + \frac{2^2}{1000 \cdot 0,0007} - 1000 = 4,778.$$

Найдем квантиль распределения “хи-квадрат”:  $\chi^2_{\alpha,s} = \chi^2_{0,05;11} = 19,675$ . Так как  $\chi^2 < \chi^2_{\alpha,s}$ , можно утверждать, что случайные величины, получаемые с помощью алгоритма имитации времени охлаждения плит (и вычисляемые с помощью подпрограммы exron), соответствуют плотности распределения (3.6). Таким образом, алгоритм имитации времени охлаждения плит также можно признать правильным.

После проверки алгоритмов имитации они могут применяться для решения практических задач анализа и оптимизации объектов моделирования.

### 3.3. Примеры решения задач на основе алгоритмов имитации случайных величин

**Пример 3.6.** По данным примера 3.2 требуется разработать алгоритм и программу для имитации процесса производства пластмассовых плит. Определить среднее время изготовления одной плиты.

При разработке модели будем считать, что время изготовления плиты складывается из двух случайных величин: времени внесения добавок и времени охлаждения плиты в форме.

Приведем алгоритм имитации производства плит.

3. Выполняется имитация времени внесения добавок ( $t_{\text{внес}}$ ). Для этого используется формула (3.5).

4. Выполняется имитация времени охлаждения плиты в форме ( $t_{\text{охлажд}}$ ): экспоненциальная случайная величина со средним значением 30 мин. Для этого используется формула (3.10) при  $\bar{X}=30$ .

5. Определяется время изготовления плиты:  $t = t_{\text{внес}} + t_{\text{охлажд}}$ .

Шаги 1-3 повторяются многократно (например, 100 000 раз). Определяется среднее время выпуска *одной* плиты: отношение *суммы времени выпуска всех плит* к количеству этих плит (т.е. к количеству испытаний).

Рассмотрим одно испытание алгоритма имитации.

Выполняется имитация времени внесения добавок по формуле (3.5). Пусть разыграно  $R=0,0795$ . Тогда  $t_{\text{внес}}=(0,0795+0,3)/0,1=3,795$  мин.

Выполняется имитация времени охлаждения плиты по формуле (3.10). Пусть разыграно  $R=0,3780$ . Тогда  $t_{\text{охлажд}}=-30 \cdot \ln(0,3780)=29,186$  мин.

Таким образом, в данном испытании смоделировано время изготовления плиты  $t=3,795+29,186=32,981$  мин.

Приведем программу на языке VBA, реализующую алгоритм имитации производства плит.

```
Sub primer36()  
'Имитация выпуска пластмассовых плит  
n = 100000  
tzs = 30  
For i = 1 To n  
t = 0  
Call vnesdob(tvnes)      'Имитация внесения добавок  
t = t + tvnes  
Call expon(tzs, tohlazd) 'Имитация охлаждения  
t = t + tohlazd  
tsum = tsum + t  
Next i  
tsred = tsum / n  
MsgBox ("Среднее время изготовления плиты " + Str(Round(tsred,2)))  
End Sub  
  
Sub vnesdob(x)  
r = Rnd  
If r < 0.3 Then x = (r + 0.3) / 0.1  
If (r >= 0.3) And (r < 0.9) Then x = (r + 0.6) / 0.15  
If r >= 0.9 Then x = (r - 0.4) / 0.05  
End Sub  
  
Sub expon(xsred, x)  
r = Rnd  
x = -xsred * Log(r)  
End Sub
```

Среднее время изготовления плиты, полученное по результатам моделирования, составляет около 37 минут.

**Пример 3.7.** В ремонтной службе предприятия выполняется наладка и ремонт некоторых приборов. Каждый прибор состоит из пяти схем. Каждая из

схем может оказаться неисправной с вероятностью 0,1 (т.е. в 10% случаев).  
Наладка и ремонт прибора включает следующие операции:

- осмотр: от 2 до 5 мин на каждую схему;
- замена неисправных схем: время замены одной схемы - гауссовская случайная величина со средним значением 5 мин и стандартным отклонением 1 мин;
- наладка. Если в приборе не потребовалась замена схем, то выполняется мелкая наладка; если была заменена хотя бы одна схема, то необходима полная наладка. Время наладки можно считать случайной величиной, распределенной по экспоненциальному закону. Мелкая наладка занимает в среднем 10 мин, полная – 15 мин.

Требуется разработать алгоритм и программу имитации наладки и ремонта приборов на основе метода Монте-Карло. Определить: а) среднее время наладки и ремонта одного прибора; б) процент случаев, когда требуется полная наладка; в) среднее количество схем, заменяемых в одном приборе.

При разработке модели будем считать, что время наладки и ремонта прибора складывается из нескольких случайных величин: времени осмотра всех схем, времени замены неисправных схем (если такие схемы есть), времени наладки прибора.

Приведем алгоритм имитации процесса наладки и ремонта приборов.

1. Выполняется имитация осмотра пяти схем прибора. Время осмотра каждой схемы - случайная величина, распределенная по равномерному закону в интервале (2; 5). Для имитации такой величины используется формула (3.11), при  $a=2$ ,  $b=5$ . Кроме того, имитируется результат осмотра (исправное или неисправное состояние схемы). Для этого разыгрывается СРРЧ  $R$ ; если  $R < 0,1$ , то схема неисправна, в противном случае – исправна. Действия этого шага повторяются пять раз (для каждой схемы).

2. Если на шаге 1 была смоделирована неисправность хотя бы одной схемы, то выполняется шаг 3, в противном случае – шаг 4.

3. Выполняется имитация замены неисправных схем. Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=5$ ,  $\sigma=1$ . Действия этого шага повторяются столько раз, сколько неисправных схем было смоделировано на шаге 1.

4. Выполняется имитация наладки прибора. Для этого используется формула (3.10) при  $\bar{X}=10$ , если на шаге 1 не было смоделировано ни одной неисправной схемы, или  $\bar{X}=15$ , если на шаге 1 была смоделирована хотя бы одна неисправная схема.

5. Вычисляется время наладки и ремонта прибора: сумма значений времени, смоделированных на шагах 1,3,4.

Шаги 1-5 повторяются многократно (например, 100 000 раз). Вычисляются величины, которые требуется определить по результатам моделирования. *Среднее время наладки и ремонта одного прибора* определяется как отношение суммы времени наладки и ремонта всех приборов к количеству этих приборов (т.е. к количеству испытаний). *Процент случаев полной наладки* вычисляется как отношение количества случаев полной наладки к количеству испытаний (умноженное на 100). *Среднее количество*

схем, заменяемых в одном приборе, определяется как отношение общего количества неисправных схем к количеству испытаний.

Приведем программу на языке VBA, реализующую алгоритм имитации наладки и ремонта приборов.

```
Sub primer37()  
    ' Имитация наладки и ремонта приборов  
    n = 100000  
    tpoln = 15: tmelk = 10  
    For i = 1 To n  
        kolneisp1 = 0: t = 0  
        For j = 1 To 5  
            Call ravn(2, 5, tosm)          'Имитация осмотра всех схем прибора  
            t = t + tosm  
            r = Rnd  
            If r < 0.1 Then kolneisp1 = kolneisp1 + 1 'Подсчет количества неисправных схем в приборе  
        Next j  
        kolneisp = kolneisp + kolneisp1 'Подсчет количества неисправных схем во всех приборах  
        If kolneisp1 > 0 Then 'Проверка: есть ли в приборе неисправные схемы  
            For j = 1 To kolneisp1 'Имитация замены неисправных схем  
                Call gauss(5, 1, tzam)  
                t = t + tzam  
            Next j  
            Call expon(tpoln, tnal)      'Имитация полной наладки  
            t = t + tnal  
            kolpn = kolpn + 1  
        Else  
            Call expon(tmelk, tnal)     'Имитация мелкой наладки  
            t = t + tnal  
        End If  
        tsum = tsum + t  
    Next i  
    tsred = tsum / n  
    ppn = (kolpn / n) * 100  
    kolzamsred = kolneisp / n  
    MsgBox ("Среднее время наладки и ремонта прибора: " + Str(Round(tsred,2)))  
    MsgBox ("Процент случаев полной наладки: " + Str(Round(ppn,2)))  
    MsgBox ("Среднее количество заменяемых схем на один прибор: " + Str(Round(kolzamsred,2)))  
End Sub  
  
Sub ravn(a, b, x)  
    r = Rnd  
    x = a + (b - a) * r  
End Sub  
  
Sub expon(xsred, x)  
    r = Rnd  
    x = -xsred * Log(r)  
End Sub  
  
Sub gauss(m, sigma, x)  
    k = 6  
    For i = 1 To k  
        r = Rnd: rsum = rsum + r  
    Next i  
    x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
```

End Sub

Здесь подпрограмма `gaup` реализует алгоритм имитации равномерной случайной величины (формула (3.11)).

Результаты моделирования следующие: среднее время ремонта – около 32,07 мин, процент случаев полной наладки – 40,79%, среднее количество заменяемых схем в одном приборе – 0,5.

**Пример 3.8.** Некоторые изделия изготавливаются путем наложения двух пластин. Номинальная толщина первой пластины – 6 мм, второй – 4 мм. Так как в производственном процессе неизбежны отклонения, фактически толщина первой пластины представляет собой гауссовскую случайную величину со средним значением 6 мм и стандартным отклонением 0,2 мм. Толщина второй пластины – также гауссовская случайная величина со средним значением 4 мм и стандартным отклонением 0,1 мм. Готовое изделие считается годным, если его толщина составляет от 9,7 до 10,3 мм.

Требуется разработать алгоритм и программу имитации выпуска изделий на основе метода Монте-Карло. Определить среднюю толщину готового изделия и вероятность выпуска годного изделия.

Приведем алгоритм имитации процесса изготовления изделий.

1. Выполняется имитация толщины первой пластины ( $X_1$ ). Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=6$ ,  $\sigma=0,2$ .

2. Выполняется имитация толщины второй пластины ( $X_2$ ). Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=4$ ,  $\sigma=0,1$ .

3. Определяется толщина изделия:  $X=X_1+X_2$ .

4. Если  $9,7 \leq X \leq 10,3$ , то изделие годное, в противном случае – бракованное.

Шаги 1-4 повторяются многократно (например, 100 000 раз). Вычисляются величины, которые требуется определить по результатам моделирования. *Средняя толщина готового изделия* определяется как отношение суммы значений толщины всех изделий к количеству этих изделий (т.е. к количеству испытаний). *Вероятность выпуска годного изделия* вычисляется как отношение количества годных изделий к количеству испытаний.

Приведем программу на языке VBA, реализующую алгоритм имитации изготовления изделий.

```
Sub primer38()
```

```
‘ Имитация изготовления изделий из двух пластин
```

```
n = 100000
```

```
For i = 1 To n
```

```
Call gauss(6, 0.2, x1) ‘Имитация толщины первой пластины
```

```
Call gauss(4, 0.1, x2) ‘Имитация толщины второй пластины
```

```
x=x1+x2 ‘Определение толщины изделия
```

```
If (x>=9.7) And (x<=10.3) Then kolgodn=kolgodn+1 ‘Имитация проверки изделия
```

```
xsum=xsum+x
```

```
Next i
```

```
xsred = xsum / n
```

```
pgodn = kolgodn / n
```

```
MsgBox ("Средняя толщина изделия: " + Str(Round(xsred,2)))
```

```
MsgBox ("Вероятность выпуска годного изделия: " + Str(Round(pgodn,4)))
```

```
End Sub
```

```

Sub gauss(m, sigma, x)
k = 6
For i = 1 To k
r = Rnd: rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub

```

Результаты моделирования следующие: средняя толщина изделия – примерно 10 мм, вероятность выпуска годного изделия – 0,82.

**Пример 3.9.** В автоматизированной системе управления технологическим процессом передаются сигналы от производственного оборудования (объекта управления) к управляющему компьютеру. Длительность передачи сигнала - случайная величина, распределенная по экспоненциальному закону со средним значением 5 мс. В канале связи возможны помехи. Интервалы между моментами помех - случайные величины, распределенные по экспоненциальному закону. Помехи возникают в среднем 20 раз в секунду. Если во время передачи сигнала возникает хотя бы одна помеха, то сигнал искажается.

Требуется разработать алгоритм и программу имитации передачи сигналов на основе метода Монте-Карло. Определить вероятность передачи сигнала без искажений.

Так как *интервалы времени* между помехами представляют собой экспоненциальные случайные величины, *количество* помех за некоторый интервал времени  $t$  представляет собой случайную величину, распределенную по пуассоновскому закону с параметром  $\lambda = \Lambda t$ , где  $\Lambda$  – интенсивность потока помех. В данном примере  $\Lambda = 20$  помех/с, или 0,02 помехи/мс.

Приведем алгоритм имитации процесса передачи сигналов.

1. Выполняется имитация времени передачи сигнала ( $t$ ). Для этого разыгрывается экспоненциальная случайная величина по формуле (3.10) при  $\bar{X} = 5$ .

2. Разыгрывается количество помех ( $m$ ) за время передачи сигнала  $t$ . Для этого используется алгоритм имитации пуассоновской случайной величины, приведенный в подразд. 3.1, при  $\lambda = 0,02t$ .

3. Если  $m = 0$  (за время передачи сигнала не было ни одной помехи), то сигнал не искажен.

Шаги 1-3 повторяются многократно (например, 100 000 раз). Вычисляется вероятность передачи сигнала без искажений как отношение количества сигналов, переданных без искажений, к общему количеству смоделированных сигналов (количеству испытаний).

Приведем программу на языке VBA, реализующую алгоритм имитации передачи сигналов.

```

Sub primer 39()
' Имитация передачи сигналов в АСУТП
n = 100000
tsred = 5 : intens = 0.02
For i = 1 To n
Call expon(tsred,t) 'Имитация времени передачи сигнала
l=intens*t 'Вычисление параметра пуассоновского распределения
Call puasson(l, m) 'Розыгрыш количества помех за время передачи сигнала
If m=0 then kolnorm=kolnorm+1 'Если помех не было, то сигнал передан без искажений

```

```

Next i
pnorm = kolnorm / n
MsgBox ("Вероятность нормальной передачи сигнала: " + Str(Round(pnorm,4)))
End Sub

Sub expon(xsred, x)
r = Rnd
x = -xsred * Log(r)
End Sub

Sub puasson(l, m) 'Подпрограмма имитации пуассоновской случайной величины
p0=exp(-l) 'на основе алгоритма, приведенного в подразд. 3.1
q=1 : m=0
povtor:
r=Rnd
q=q*r
If q>=p0 Then
m=m+1
Goto povtor
End If
End Sub

```

По результатам моделирования, вероятность передачи сигнала без искажений составляет примерно 0,909.

**Пример 3.10.** Предприятие выпускает электроприборы, состоящие из четырех блоков. Прибор продолжает работать, пока исправен хотя бы один блок. Время безотказной работы каждого блока - случайная величина, распределенная по экспоненциальному закону; среднее время безотказной работы блока – два года. Срок действия гарантии на прибор - один год.

Требуется разработать алгоритм и программу имитации работы прибора на основе метода Монте-Карло. Определить: а) среднее время безотказной работы прибора; б) вероятность безотказной работы прибора в течение гарантийного срока.

При разработке алгоритма решения этой задачи под временем безотказной работы прибора будем понимать время, в течение которого прибор продолжает работать (даже если некоторые из блоков, входящих в состав прибора, отказали).

Приведем алгоритм имитации работы прибора.

1. Разыгрываются значения времени безотказной работы каждого из блоков ( $t_{rb1}, t_{rb2}, t_{rb3}, t_{rb4}$ ) – четыре случайных величины, распределенные по экспоненциальному закону, со средним значением 2. Для этого используется формула (3.10) при  $\bar{X}=2$ .

2. Определяется время безотказной работы прибора – максимальная из величин, разыгранных на шаге 1 (так как прибор продолжает работать, пока исправен хотя бы один из блоков):  $t_{rp}=\max(t_{rb1}, t_{rb2}, t_{rb3}, t_{rb4})$ .

3. Если  $t_{rp}\geq 1$  (т.е. не менее одного года), то прибор отработал гарантийный срок без отказа.

Шаги 1-3 повторяются многократно (например, 100 000 раз). По окончании испытаний вычисляются величины, которые требуется определить по результатам моделирования. *Среднее время безотказной работы прибора* определяется как отношение суммы значений времени безотказной работы всех приборов (т.е. суммы величин  $t_{rp}$ ) к количеству этих приборов. *Вероятность безотказной работы в течение гарантийного срока* вычисляется как

отношение количества приборов, отработавших гарантийный срок без отказа, к общему количеству приборов.

Приведем программу на языке VBA, реализующую алгоритм имитации работы приборов.

```
Sub primer310()  
' Имитация работы приборов в течение гарантийного срока  
Dim trb(1 To 4)  
n = 100000  
tsred = 2  
For i = 1 To n  
For j = 1 To 4  
Call expon(tsred, trb(j))      'Имитация времени безотказной работы блока  
Next j  
trp = 0  
For j = 1 To 4                'Определение времени работы прибора -  
If trp < trb(j) Then trp = trb(j) 'максимального из значений времени работы блоков  
Next j  
tsum = tsum + trp  
If trp >= 1 Then bezotk = bezotk + 1 'Подсчет приборов, отработавших гарантийный срок  
Next i  
trpsred = tsum / n  
pbezotk = bezotk / n  
MsgBox ("Среднее время безотказной работы прибора: " + Str(Round(trpsred,2)))  
MsgBox ("Вероятность безотказной работы в течение гарантийного срока: " + Str(Round(pbezotk,4)))  
End Sub  
  
Sub expon(xsred, x)  
r = Rnd  
x = -xsred * Log(r)  
End Sub
```

По результатам моделирования среднее время безотказной работы прибора составляет примерно 4,17 года, а вероятность безотказной работы в течение гарантийного срока - 0,9754.

### 3.4. Интервальные оценки в имитационных моделях

Величины, определяемые по результатам имитационного моделирования, не могут быть абсолютно точными. Выполнив последовательность испытаний имитационной модели (например, 1000 или 100 000 испытаний) и получив значения исследуемых величин (например, среднюю прибыль от выпуска изделия  $\bar{X}$  и вероятность выпуска годного изделия  $P$ ), важно понимать, что по результатам *другой* последовательности испытаний были бы получены *другие* значения этих величин. Для получения *истинного* (абсолютно точного) значения потребовалось бы (теоретически) *бесконечно большое* количество испытаний. Поэтому представляют интерес оценки, полученные с заданной точностью, а также интервальные оценки. Конкретные значения исследуемых величин, полученные по результатам имитационного моделирования, представляют собой **точечные оценки**. Все результаты, полученные в рассмотренных выше примерах, являются точечными оценками.

Под *точностью* оценки будем понимать ее *отклонение от истинного значения* исследуемой величины (т.е. разность между полученной оценкой и истинным значением), а под *относительной точностью* оценки – то же *отклонение*, выраженное в *процентах* (или в долях) от *максимально возможного отклонения*. Будем обозначать точность как  $\varepsilon\beta$ , а относительную точность – как  $\varepsilon\beta_0$  (смысл обозначения  $\beta$  будет показан ниже). При этом важно понимать, что истинное значение исследуемой величины не может быть известно точно.

**Примечание.** Максимально возможное отклонение полученной точечной оценки от истинного значения исследуемой величины определяется по-разному в зависимости от того, какую величину требуется определить по результатам моделирования. Например,

можно доказать, что максимально возможное отклонение *точечной оценки среднего значения* исследуемой величины от *истинного значения среднего* можно считать равным  $3\sigma$ , где  $\sigma$  – стандартное отклонение данной величины. Максимально возможное отклонение *точечной оценки вероятности* некоторого события от *истинного значения* этой вероятности равно единице.

В практических задачах обычно требуется получение таких оценок, чтобы их *отклонение* от истинного (точно не известного) значения *не превышало некоторую предельную величину с заданной* (достаточно высокой) *вероятностью*. Эту вероятность будем обозначать как  $\beta$ .

Для решения такой задачи необходимо задать требуемую относительную точность оценки, т.е. величину  $\varepsilon_{\beta 0}$ , обычно выражаемую в процентах. Как правило, требуется значение  $\varepsilon_{\beta 0}$  на уровне 3-5%. Например,

если задана относительная точность  $\varepsilon_{\beta 0}=5\%$  и доверительная вероятность  $\beta=0,995$ , это значит, что

требуется получить оценку, отклонение которой от истинного значения исследуемой величины с вероятностью 0,995 будет составлять не более 5% от максимально возможного значения такого отклонения.

Указав заданную относительную точность  $\varepsilon_{\beta 0}$  и доверительную вероятность  $\beta$ , можно определить

*количество испытаний*, необходимое для обеспечения заданной относительной точности. Очевидно, что чем выше требуемая точность, тем больше необходимое количество испытаний.

Как видно из примеров, рассмотренных в предыдущих подразделах, в задачах моделирования обычно

требуется определять оценки *среднего значения* некоторой величины  $\bar{X}$  или *вероятности* некоторого события  $P$ . Можно доказать, что количество испытаний  $n$ , необходимое для получения этих оценок с заданной относительной точностью при доверительной вероятности  $\beta=0,995$ , определяется по следующим формулам:

- для среднего значения:

$$n = \frac{1}{\varepsilon_{\beta 0}^2}; \quad (3.18)$$

- для вероятности:

$$n = \frac{2}{\varepsilon_{\beta 0}^2}, \quad (3.19)$$

где  $\varepsilon_{\beta 0}$  - заданная относительная точность оценки.

Выполнив необходимое количество испытаний  $n$ , можно определить точность полученной оценки  $\varepsilon_{\beta}$  с заданной доверительной вероятностью  $\beta$ .

Для **среднего значения** точность оценки определяется по следующей формуле:

$$\varepsilon_{\beta} = 3\varepsilon_{\beta 0} \cdot \sigma. \quad (3.20)$$

Здесь  $\sigma$  – стандартное отклонение, вычисляемое по формуле

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2}, \quad (3.21)$$

где  $x_i, i=1, \dots, n$  – значения исследуемой величины, полученные в каждом из  $n$  испытаний в ходе имитационного моделирования;

$\bar{X}$  - среднее значение, вычисленное по результатам  $n$  испытаний.

Для программной реализации более удобна следующая формула для вычисления стандартного отклонения, которую можно вывести из формулы (3.21):

$$\sigma = \sqrt{\frac{n}{n-1} \left( \frac{\sum_{i=1}^n x_i^2}{n} - \bar{X}^2 \right)}. \quad (3.22)$$

Для **вероятности** точность оценки определяется по следующей формуле:

$$\varepsilon_\beta = 2,8 \sqrt{\frac{P(1-P)}{n}}, \quad (3.23)$$

где  $P$  – значение вероятности, вычисленное по результатам  $n$  испытаний.

Как показано в примерах выше, вероятность некоторого события  $P$  вычисляется по результатам имитационного моделирования как отношение количества случаев, когда было смоделировано наступление этого события, к общему количеству испытаний ( $n$ ).

Вычислив точность  $\varepsilon_\beta$ , можно получить интервальную оценку исследуемой величины. **Интервальная оценка** некоторой величины – это числовой интервал, содержащий *истинное* значение этой величины с *заданной* (достаточно высокой) *доверительной вероятностью*.

**Примечание.** Важно понимать, что построение интервальной оценки с доверительной вероятностью, равной единице, невозможно: для этого требуется бесконечное количество испытаний.

Интервальная оценка для **среднего значения** определяется следующим образом:

$$\bar{X} \pm \varepsilon_\beta, \quad (3.24)$$

где  $\varepsilon_\beta$  - точность оценки, найденная по формуле (3.20).

Интервальная оценка для **вероятности** определяется следующим образом:

$$P \pm \varepsilon_\beta, \quad (3.25)$$

где  $\varepsilon_\beta$  - точность оценки, найденная по формуле (3.23).

Таким образом, интервальные оценки могут быть получены с использованием имитационного моделирования в следующем порядке.

1. Задается требуемая относительная точность оценки  $\varepsilon\beta_0$ .
2. Определяется необходимое количество испытаний  $n$  по формуле (3.18) или (3.19).
3. Выполняется  $n$  испытаний имитационной модели.
4. По результатам имитационного моделирования вычисляются точечные оценки  $\bar{X}$  или  $P$ .
5. Определяется точность полученных оценок по формулам (3.20) или (3.23).
6. Определяются интервальные оценки по формулам (3.24) или (3.25).

**Пример 3.11.** В условиях примера 3.8 требуется определить среднюю толщину готового изделия и вероятность выпуска годного изделия с относительной точностью 5%.

Найдем необходимое количество испытаний. Количество испытаний для определения *средней* толщины готового изделия вычисляется по формуле (3.18):  $n=1/0,05^2=400$ . Количество испытаний для определения *вероятности* выпуска годного изделия вычисляется по формуле (3.19):  $n=2/0,05^2=800$ . Так как необходимо обеспечить заданную точность для обеих величин, потребуется выполнить 800 испытаний.

**Примечание.** Если бы в задаче требовалось определить с заданной относительной точностью только среднюю толщину, то было бы достаточно 400 испытаний.

В программе, реализующей имитационную модель, необходимо предусмотреть вычисление величин точности по формулам (3.20) и (3.23) и интервальных оценок по формулам (3.24) и (3.25). Чтобы вычислить точность оценки средней толщины изделия, потребуется также предусмотреть вычисление стандартного отклонения по формуле (3.22). Программа будет иметь следующий вид.

```

Sub primer311()
' Имитация изготовления изделий из двух пластин
n = 800 : epsb0=0.05
For i = 1 To n
Call gauss(6, 0.2, x1)
Call gauss(4, 0.1, x2)
x=x1+x2
If (x>=9.7) And (x<=10.3) Then kolgodn=kolgodn+1
xsum=xsum+x
xsum2=xsum2+x^2
Next i
xsred = xsum / n
pgodn = kolgodn / n
sigma=sqr((n/(n-1))*(xsum2/n-xsred^2))
epsb_sred=3*epsb0*sigma
xniz=xsred-epsb_sred : xverh=xsred+epsb_sred
epsb_ver=2.8*sqr(pgodn*(1-pgodn)/n)
pniz=pgodn-epsb_ver : pverh=pgodn+epsb_ver
MsgBox ("Средняя толщина изделия: " + Str(Round(xsred,2)))
MsgBox ("Интервальная оценка средней толщины: " + Str(Round(xniz,2))+” – “+ Str(Round(xverh,2)))
MsgBox ("Вероятность выпуска годного изделия: " + Str(Round(pgodn,4)))
MsgBox ("Интервальная оценка вероятности: " + Str(Round(pniz,4))+” – “+Str(Round(pverh,4)))
End Sub

Sub gauss(m, sigma, x)
k = 6
For i = 1 To k
r = Rnd: rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub

```

Здесь переменная  $xsum2$  используется для вычисления суммы квадратов значений толщины. Эта сумма требуется для вычисления стандартного отклонения по формуле (3.22). Смысл остальных переменных, добавленных в программу, следующий:  $epsb0=0,05$  – заданная относительная точность;  $\sigma$  (в основной программе) – стандартное отклонение, вычисляемое по формуле (3.22);  $epsb\_sred$ ,  $epsb\_ver$  – значения точности оценок средней толщины и вероятности, вычисляемые по формулам (3.20) и (3.23) соответственно;  $xniz$ ,  $xverh$  – границы доверительного интервала для средней толщины, вычисляемые по формуле (3.24);  $pniz$ ,  $pverh$  – границы доверительного интервала для вероятности, вычисляемые по формуле (3.25).

Результаты моделирования будут примерно следующими: средняя толщина изделия – 9,99 мм, интервальная оценка – (9,96; 10,02) мм; вероятность выпуска годного изделия – 0,8338, интервальная оценка – (0,7969; 0,8706). Это означает, что с вероятностью 0,995 (или 99,5%) истинная средняя толщина изделия будет составлять от 9,96 до 10,02 мм, а вероятность выпуска годного изделия – от 0,7969 до 0,8706. С вероятностью 0,995 разность между точечной оценкой средней толщины, полученной в результате моделирования (9,99 мм), и истинным (точно не известным) значением средней толщины составляет не более 5% от максимально возможного значения такой разности. Аналогично, с вероятностью 0,995 разность между точечной оценкой вероятности выпуска годного изделия (0,8338) и истинным (точно не известным) значением этой вероятности составляет не более 5% от максимально возможного значения такой разности.

### Определение точности оценок при заданном количестве испытаний.

Если задано количество испытаний имитационной модели ( $n$ ), то можно определить относительную точность оценок, полученных по результатам моделирования. Из формул (3.18) и (3.19) видно, что относительную точность оценок можно определить следующим образом:

- для среднего значения:

$$\varepsilon_{\beta 0} = \frac{1}{\sqrt{n}}; \quad (3.26)$$

- для вероятности:

$$\varepsilon_{\beta 0} = \sqrt{\frac{2}{n}}. \quad (3.27)$$

**Пример 3.12.** Найти относительную точность оценок среднего времени безотказной работы прибора и вероятности его безотказной работы в течение гарантийного срока, полученных в примере 3.10.

Оценки (среднее время безотказной работы прибора - 4,17 года, вероятность безотказной работы в течение гарантийного срока - 0,97) были получены по результатам 100 000 испытаний. Относительную точность оценки среднего времени безотказной работы найдем по формуле (3.26):  $\varepsilon_{\beta 0} = \sqrt{1/100000} = 0,0032$ , или 0,32%. Относительная точность оценки вероятности безотказной работы в течение гарантийного срока определяется по формуле (3.27):  $\varepsilon_{\beta 0} = \sqrt{2/100000} = 0,0045$ , или 0,45%. Это означает, что с вероятностью 0,995 разность между оценкой среднего времени безотказной работы, полученной по результатам моделирования (4,17 года), и истинным (точно не известным) значением среднего времени безотказной работы составляет не более 0,32% от максимально возможного значения такой разности. Разность между оценкой вероятности безотказной работы, полученной по результатам моделирования (0,9754), и истинным (точно не известным) значением вероятности безотказной работы с вероятностью 0,995 составляет не более 0,45% от максимально возможного значения такой разности.

### 3.5. Средства имитационного моделирования в Excel

Для получения СРРЧ в табличном процессоре Excel используется функция СЛЧИС (из категории Математические).

Кроме того, в Excel имеются стандартные средства для имитации некоторых случайных величин, часто применяемых в практических задачах моделирования. Имеется возможность розыгрыша чисел, распределенных по гауссовскому (нормальному), равномерному, пуассоновскому, биномиальному и некоторым другим законам. Для этого используется инструмент Генерация случайных чисел (меню Сервис – Анализ данных).

**Пример 3.13.** Решить в Excel задачу из примера 3.8. Выполнить 100 испытаний.

Задача решается в следующем порядке.

1. Выбрать из меню Excel команду **Сервис – Анализ данных – Генерация случайных чисел**.

2. В поле **Число переменных** ввести 1; это означает, что создаваемые случайные числа должны выводиться в один столбец. В поле **Число случайных чисел** ввести 100; это означает, что требуется получить 100 случайных чисел. В поле **Распределение** выбрать значение **Нормальное**. В поле **Среднее** ввести значение 6, в поле **Стандартное отклонение** – значение 0,2. Поле **Случайное рассеивание** можно оставить пустым (его смысл показан ниже). Установить переключатель **Выходной интервал**, и в соответствующем поле ввести значение **A1**; это означает, что заданное количество чисел (100) будет выведено в столбец, начиная с ячейки A1. Нажать **ОК**. В ячейки A1:A100 выводится 100 случайных чисел, распределенных по гауссовскому закону с указанными параметрами. Эти числа имитируют толщину первой пластины.

3. Аналогичным образом получить в столбце B 100 случайных чисел, имитирующих толщину второй пластины. Для этого снова выбрать команду **Сервис – Анализ данных – Генерация случайных чисел**. Установить параметры: **Число переменных** – 1, **Число случайных чисел** – 100, **Распределение** – **Нормальное**, **Среднее** - 4, **Стандартное отклонение** – 0,1, **Выходной интервал** - B1. Нажать **ОК**.

4. В столбце 100 вычислить значения толщины готового изделия (суммы случайных чисел, полученных в столбцах A и B). Для этого в ячейке C1 ввести формулу:  $=A1+B1$ . Распространить ее на ячейки C2:C100.

5. В ячейке C101 вычислить среднюю толщину готовых изделий по результатам 100 испытаний. Для этого использовать функцию **СРЗНАЧ** с аргументом C1:C100.

6. В ячейке D1 ввести функцию **И** (из категории **Логические**). В поле **Логическое1** ввести условие  $C1 \geq 9,7$ , в поле **Логическое2** – условие  $C1 \leq 10,3$ . Нажать **ОК**. Распространить содержимое ячейки D1 на ячейки D2:D100. В результате в ячейках D1:D100 будут получены значения **ИСТИНА**, если выполняется условие  $9,7 \leq C1 \leq 10,3$  (т.е. изделие годное), или **ЛОЖЬ** – если это условие не выполняется (изделие бракованное).

7. В ячейке D101 подсчитать количество годных изделий. Для этого ввести функцию **СЧЕТЕСЛИ** (из категории **Статистические**) со следующими аргументами: **Диапазон** – D1:D100, **Условие** – **Истина**. Подсчитывается количество случаев, когда в ячейках D1:D100 имеется значение **ИСТИНА**, т.е. количество смоделированных годных изделий.

8. В ячейке D102 вычислить вероятность выпуска годного изделия. Для этого разделить количество годных изделий на количество испытаний:  $=D101/100$ .

Результаты, полученные в Excel, могут несколько отличаться от полученных в примере 3.8. Точность результатов, полученных в Excel, ниже, так как они получены по результатам меньшего количества испытаний. Однако моделирование в Excel можно считать приемлемым способом получения приближенных оценок для несложных задач.

Примечание. Поле **Случайное рассеивание** можно использовать для *неоднократного* получения *одного и того же* набора случайных чисел. Для этого в поле **Случайное рассеивание** необходимо указать некоторое произвольное число. Если затем снова выполнить генерацию случайных чисел, указав *то же* число в поле **Случайное рассеивание**, то будет получен такой же набор случайных чисел. Конечно, для этого все остальные параметры моделирования (**Число случайных чисел**, **Распределение** и т.д.) также должны быть прежними.

**Пример 3.14.** Решить в Excel задачу из примера 3.10. Выполнить 100 испытаний.

В рассматриваемом примере время безотказной работы блоков, входящих в состав прибора, представляет собой случайную величину, распределенную по экспоненциальному закону. В Excel нет стандартных средств для имитации таких величин. Поэтому для розыгрыша случайных чисел, имитирующих время работы блоков, потребуется воспользоваться формулой (3.10). Для розыгрыша СРРЧ будет использоваться функция СЛЧИС.

Задача решается в следующем порядке.

1. Ввести в ячейку A1 формулу  $=-2*LN(СЛЧИС())$ . Таким образом выполняется розыгрыш случайного числа, распределенного по экспоненциальному закону, со средним значением  $\bar{X}=2$ , т.е. имитация времени безотказной работы *одного блока*.

2. Распространить формулу, введенную в ячейку A1, на диапазон ячеек A1:D100. Таким образом имитируется время безотказной работы 400 блоков, входящих в 100 приборов. Четыре случайных числа, разыгрываемых в каждой строке, имитируют время безотказной работы четырех блоков, входящих в один прибор.

3. В столбце E найти значения времени безотказной работы каждого из приборов (максимальное из значений времени безотказной работы блоков, входящих в прибор). Для этого в ячейке E1 ввести формулу:  $=МАКС(A1:D1)$ . Распространить ее на ячейки E2:E100.

4. В ячейке E101 вычислить среднее время безотказной работы приборов по результатам 100 испытаний. Для этого использовать функцию СРЗНАЧ с аргументом E1:E100.

5. В столбце F получить значения ИСТИНА для случаев, когда смоделирована безотказная работа прибора в течение гарантийного срока, и значения ЛОЖЬ – для случаев, когда смоделирован отказ прибора. Для этого в ячейке F1 ввести функцию ЕСЛИ (из категории Логические) со следующими аргументами: Логическое выражение –  $E1 \geq 1$ , Значение если истина – ИСТИНА, Значение если ложь – ЛОЖЬ. Нажать ОК. Распространить содержимое ячейки F1 на ячейки F2:F100.

6. В ячейке F101 подсчитать количество приборов, отработавших гарантийный срок без отказа. Для этого ввести функцию СЧЕТЕСЛИ (из категории Статистические) со следующими аргументами: Диапазон – F1:F100, Условие – Истина. Подсчитывается количество случаев, когда в ячейках F1:F100 имеется значение ИСТИНА, т.е. количество случаев, когда смоделирована нормальная работа прибора в течение гарантийного срока.

7. В ячейке F102 вычислить вероятность безотказной работы прибора в течение гарантийного срока:  $=F101/100$ .

## **4. АНАЛИЗ И ОПТИМИЗАЦИЯ РЕШЕНИЙ В ДЕТЕРМИНИРОВАННЫХ ЗАДАЧАХ НА ОСНОВЕ МЕТОДА МОНТЕ-КАРЛО**

### **4.1. Применение метода Монте-Карло для решения**

#### **детерминированных задач оптимизации**

Хотя метод Монте-Карло основан на использовании случайных величин (СРРЧ), он может эффективно применяться для решения детерминированных задач, т.е. задач, в постановке которых нет никаких случайных факторов. Метод Монте-Карло может применяться для решения как вычислительных задач (решение уравнений и систем уравнений, вычисление интегралов), так и задач оптимизации, т.е. задач, в которых требуется выбор лучшего из нескольких возможных вариантов решения.

Решение задач оптимизации методом Монте-Карло в большинстве случаев основано на многократном случайном выборе вариантов решения и их сравнении. При достаточном количестве испытаний (т.е. вариантов решения, выбранных случайным образом) находится решение, близкое к оптимальному, а в некоторых случаях - оптимальное.

Для многократного случайного выбора вариантов решения задачи применяются СРРЧ. Из вариантов решений, выбранных случайным образом, отбирается лучший. Чтобы найти вариант решения, достаточно близкий к оптимальному, обычно требуется выбрать много вариантов решения (от нескольких десятков до тысяч). Поэтому применение метода Монте-Карло возможно только с использованием программных средств.

Во многих случаях задачи оптимизации, решаемые методом Монте-Карло, в принципе могут быть решены другими (аналитическими) методами, позволяющими найти точное оптимальное решение. Это могут быть методы линейного программирования, динамического программирования, поиска экстремумов функций многих переменных и т.д. Однако применение точных аналитических методов, как правило, возможно только для задач небольшой размерности (с небольшим количеством ограничений, переменных и т.д.). Для реальных задач применение аналитических методов нередко оказывается невозможным или нецелесообразным из-за большого объема сложных вычислений. Поэтому во многих случаях применение метода Монте-Карло может оказаться единственно возможным способом решения задачи.

Алгоритм поиска решения на основе метода Монте-Карло полностью зависит от конкретной задачи.

## 4.2. Примеры решения задач оптимизации

### на основе метода Монте-Карло

**Пример 4.1.** Предприятие имеет возможность заключить контракты на выполнение нескольких заказов. В табл.4.1 приведены характеристики всех контрактов, которые могут быть заключены предприятием: сроки, необходимые предприятию для выполнения каждого из контрактов, и ожидаемая прибыль.

Таблица 4.1

Номер контракта	1	2	3	4	5	6
Срок выполнения, дней	15	20	10	8	12	5
Прибыль, тыс.ден.ед.	25	32	20	14	23	7

Например, если предприятие заключит первый контракт, то для его выполнения потребуется 15 дней (следует обратить внимание, что 15 дней – это именно срок, *необходимый предприятию* для выполнения контракта, а не установленный заказчиком). Прибыль предприятия в результате выполнения контракта составит 25 тыс. ден.ед.

Одновременно предприятие может работать только над одним контрактом (параллельная работа над несколькими контрактами невозможна). Все контракты, которые будут выполняться предприятием, должны быть выполнены не более чем за 49 дней.

Требуется определить перечень контрактов, которые следует заключить предприятию, чтобы получить максимальную прибыль при своевременном выполнении всех контрактов.

Принцип работы предлагаемого алгоритма решения задачи следующий. Для каждого контракта случайным образом принимается решение о том, следует ли заключать его. Выбранный набор контрактов проверяется на допустимость: определяется, будут ли все заключенные контракты выполнены за 49 дней. Такой выбор контрактов выполняется многократно. Из всех допустимых наборов контрактов выбирается тот, который обеспечивает предприятию максимальную прибыль.

Предлагаемый алгоритм реализуется следующим образом.

1. Разыгрываются шесть СРРЧ (по числу контрактов):  $R_1, R_2, \dots, R_6$ .
2. Определяется набор контрактов, заключаемых предприятием. Для этого проверяются условия  $R_i < 0,5$ ,  $i=1, \dots, 6$ . Если  $R_i < 0,5$ , то следует заключить соответствующий контракт.
3. Для выбранных контрактов находится общий срок их выполнения ( $T$ ).
4. Если  $T \leq 49$ , то выбранный набор контрактов является допустимым (т.е. предприятие выполняет все заключенные контракты в срок). В этом случае для выбранного набора контрактов рассчитывается суммарная прибыль ( $P$ ).

Шаги 1-4 выполняются многократно. Из числа допустимых наборов контрактов выбирается набор, обеспечивающий предприятию максимальную прибыль.

В табл.4.2 приведен пример десяти испытаний данного алгоритма. Значения СРРЧ взяты из прил.1. Следует обратить внимание, что при программной реализации значения СРРЧ, а значит, и варианты решений могут быть другими.

Таблица 4.2

Номер испытания	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$	Набор контрактов	$T$	$P$
1	0,0795	0,3780	0,0593	0,7602	0,2847	0,8197	1,2,3,5	57	—
2	0,2410	0,5962	0,2978	0,6458	0,9762	0,0523	1,3,6	30	52
3	0,8754	0,5900	0,3421	0,7919	0,1204	0,7492	3,5	22	43
4	0,6058	0,5321	0,8064	0,0540	0,9046	0,5778	4	8	14
5	0,3716	0,3549	0,0112	0,8438	0,3233	0,5052	1,2,3,5	57	—
6	0,0644	0,7612	0,5707	0,2759	0,8396	0,9127	1,4	23	39
7	0,7195	0,1977	0,2245	0,3979	0,0652	0,0136	2,3,4,5,6	55	—
8	0,5994	0,8691	0,0564	0,9681	0,7945	0,1137	3,6	15	27
9	0,9561	0,2147	0,9299	0,1944	0,0147	0,1935	2,4,5,6	45	76
10	0,4403	0,1689	0,0453	0,1344	0,5162	0,0540	1,2,3,4,6	58	—

Здесь, например, в первом испытании  $R_1 < 0,5$ ,  $R_2 < 0,5$ ,  $R_3 < 0,5$ ,  $R_4 > 0,5$ ,  $R_5 < 0,5$ ,  $R_6 > 0,5$ . Таким образом, для заключения выбраны первый, второй, третий и пятый контракты. Однако это решение недопустимо, так как срок выполнения этих контрактов (57 дней) превышает допустимый (49 дней).

По результатам десяти испытаний наиболее эффективный вариант действий предприятия следующий: заключить второй, четвертый, пятый и шестой контракты. Эти контракты будут выполнены предприятием за 45 дней. Прибыль составит 76 тыс ден.ед.

Приведем программную реализацию предложенного алгоритма на языке VBA, выполняющую 1000 испытаний.

```

Sub primer41()
' Выбор набора контрактов
Dim t(1 to 6), p(1 to 6), vybor(1 to 6), opt_vybor(1 to 6)
n = 1000: kol = 6: tdop = 49
t(1) = 15: t(2) = 20: t(3) = 10: t(4) = 8: t(5) = 12: t(6) = 5
p(1) = 25: p(2) = 32: p(3) = 20: p(4) = 14: p(5) = 23: p(6) = 7
For i = 1 To n
tsum = 0: psum = 0
For j = 1 To kol
r = Rnd
If r < 0.5 Then
vybor(j) = 1 'Если j-й заказ выбран, то j-му элементу массива vybor присваивается 1
tsum = tsum + t(j): psum = psum + p(j) 'Подсчет затрат времени и прибыли
Else
vybor(j) = 0 'Если j-й заказ не выбран, то j-му элементу массива vybor присваивается 0
End If

```

```

Next j
If (tsum <= tdop) And (psum > pmax) Then
pmax = psum      'Запоминается лучший вариант из рассмотренных
For j = 1 To kol
opt_vybor(j) = vybor(j)
Next j
End If
Next i
MsgBox ("Выбраны контракты")
For j = 1 To kol
If opt_vybor(j) = 1 Then
MsgBox (j)
End If
Next j
MsgBox ("Прибыль: "+Str(pmax))
End Sub

```

По результатам выполнения программы получено следующее решение: предприятию следует заключить первый, третий, четвертый и пятый контракты. Они будут выполнены за  $15+10+8+12=45$  дней, т.е. своевременно. Прибыль предприятия составит  $25+20+14+23=82$  тыс.ден.ед.

Примечание. Это решение совпадает с точным оптимальным решением, найденным методом линейного целочисленного программирования. Следует обратить внимание, что предложенный алгоритм решения задачи методом Монте-Карло, а также его программная реализация значительно проще, чем алгоритм и программная реализация методов линейного целочисленного программирования (симплекс-метод, метод ветвей и границ). Следует также обратить внимание, что решение, найденное по результатам всего десяти испытаний (табл.4.2), оказалось достаточно близким к оптимальному.

**Пример 4.2** (задача о назначениях). Требуется разместить на четырех предприятиях (П1, П2, П3, П4) заказы, связанные с выполнением четырех работ. Каждое предприятие может выполнить любой из заказов, но только один. Затраты (в денежных единицах), связанные с выполнением заказов на каждом из предприятий, приведены в табл.4.3.

Таблица 4.3

Предприятия	Заказы			
	1	2	3	4
П1	5	8	12	7
П2	8	9	7	14
П3	8	12	10	13
П4	15	12	9	10

Требуется разместить заказы таким образом, чтобы общая стоимость их выполнения была минимальной.

Алгоритм решения такой задачи на основе метода Монте-Карло реализуется в следующем порядке.

1. Разыгрывается (т.е. случайным образом выбирается) заказ для предприятия П1. Для этого диапазон (0;1) разбивается на четыре интервала: (0; 1/4), (1/4; 1/2), (1/2; 3/4), (3/4; 1) (число интервалов равно числу заказов, из которых выбирается заказ для первого предприятия). Разыгрывается СРРЧ  $R_1$ . Номер заказа, выделяемого предприятию П1, принимается равным

номеру интервала, в который попадает число  $R_1$ . Например, если  $1/4 < R_1 < 1/2$ , то предприятию П1 выделяется второй заказ.

2. Разыгрывается заказ для предприятия П2. Для этого диапазон  $(0; 1)$  разбивается на три интервала:  $(0; 1/3)$ ,  $(1/3; 2/3)$ ,  $(2/3; 1)$ . Количество интервалов равно трем, так как для второго предприятия выбирается один заказ из трех оставшихся. Разыгрывается СРРЧ  $R_2$ . По номеру интервала выбирается заказ для предприятия П2 (из *трех* оставшихся заказов).

3. Разыгрывается заказ для предприятия П3. Для этого диапазон  $(0; 1)$  разбивается на два интервала:  $(0; 1/2)$ ,  $(1/2; 1)$ . Разыгрывается СРРЧ  $R_3$ . По номеру интервала выбирается заказ для предприятия П3 (из *двух* оставшихся заказов).

4. Оставшийся (единственный) заказ выделяется предприятию П4.

5. По результатам шагов 1-4 и по табл.4.3 определяются суммарные затраты на выполнение всех заказов.

Шаги 1-5 повторяются многократно (например, 1000 раз). Выбирается вариант распределения заказов, для которого общие затраты минимальны.

В табл.4.4 приведен пример десяти испытаний данного алгоритма.

Таблица 4.4

Номер испытания	$R_1$	Заказ для П1	$R_2$	Заказ для П2	$R_3$	Заказ для П3	Заказ для П4	Затраты
1	0,0795	1	0,3780	3	0,0593	2	4	$5+7+12+10=34$
2	0,7602	4	0,2847	1	0,8197	3	2	$7+8+10+12=37$
3	0,6133	3	0,5766	2	0,9595	4	1	$12+9+13+15=49$
4	0,0981	1	0,2410	2	0,5962	4	3	$5+9+13+9=36$
5	0,2978	2	0,6458	3	0,9762	4	1	$8+7+13+15=43$
6	0,0523	1	0,4523	3	0,8153	4	2	$5+7+13+12=37$
7	0,4286	2	0,8400	4	0,8754	3	1	$8+14+10+15=47$
8	0,5900	3	0,3421	2	0,7919	4	1	$12+9+13+15=49$
9	0,1204	1	0,7492	4	0,3792	2	3	$5+14+12+9=40$
10	0,7198	3	0,4822	2	0,8215	4	1	$12+9+13+15=49$

Здесь, например, во втором испытании  $0,75 < R_1 < 1$  (предприятию П1 выделяется четвертый заказ);  $0 < R_2 < 0,33$  (предприятию П2 выделяется первый заказ);  $0,5 < R_3 < 1$  (предприятию П3 выделяется третий заказ); предприятию П4 остается второй заказ.

Таким образом, по результатам десяти испытаний наиболее эффективным решением является следующее: выделить предприятию П1 первый заказ, П2 - третий, П3 - второй, П4 - четвертый. Общие затраты на выполнение всех заказов составят 34 ден.ед.

Приведем программную реализацию алгоритма решения данной задачи на языке VBA.

Основные константы и переменные программы: kol - количество предприятий (и заказов); mз - матрица затрат (табл.4.3); spis\_zakaz - список заказов, которые еще не распределены (сначала в нем находятся все заказы); nerasp - количество заказов, которые еще не распределены; pazn - массив для хранения разыгрываемых вариантов решений (элементы этого массива - номера заказов, выделяемых предприятиям); sum\_zatr - суммарные затраты, соответствующие разыгранному варианту; opt\_pazn - массив для хранения оптимального варианта распределения заказов; min\_zatr - минимальные затраты (соответствующие оптимальному

решению); interval - величина интервала,  $a$  и  $b$  - границы интервала (см. шаги 1,2,3 алгоритма).

Sub primer42()

Dim mz(1 To 4, 1 To 4), opt\_nazn(1 To 4), nazn(1 To 4), spis\_zakaz(1 To 4)

n = 100: kol = 4

mz(1, 1) = 5: mz(1, 2) = 8: mz(1, 3) = 12: mz(1, 4) = 7

mz(2, 1) = 8: mz(2, 2) = 9: mz(2, 3) = 7: mz(2, 4) = 14

mz(3, 1) = 8: mz(3, 2) = 12: mz(3, 3) = 10: mz(3, 4) = 13

mz(4, 1) = 15: mz(4, 2) = 12: mz(4, 3) = 9: mz(4, 4) = 10

min\_zatr = 1000

For i = 1 To n

nerasp = kol

For k = 1 To kol

spis\_zakaz(k) = k 'Составление списка, первоначально содержащего номера всех заказов

Next k

For j = 1 To kol - 1

interval = 1 / nerasp : r = Rnd 'Вычисление длины интервала и розыгрыш СРРЧ

For k = 1 To nerasp

a = (k - 1) \* interval : b = k \* interval

If (r > a) And (r <= b) Then 'Определение интервала, в который попадает СРРЧ

nazn(j) = spis\_zakaz(k) 'Выделение заказа j-му предприятию

If k < nerasp Then

For jj = k To nerasp - 1

spis\_zakaz(jj) = spis\_zakaz(jj + 1) 'Исключение выделенного заказа из списка заказов

Next jj 'путем сдвига этого списка

End If

For jj = nerasp To kol

spis\_zakaz(jj) = 0

Next jj

nerasp = nerasp - 1 'Количество нераспределенных заказов уменьшилось на 1

GoTo met

End If

Next k

met: Next j

nazn(kol) = spis\_zakaz(1) 'Последнему предприятию выделяется оставшийся заказ

sum\_zatr = 0

For j = 1 To kol

k = nazn(j)

sum\_zatr = sum\_zatr + mz(j, k) 'Вычисление затрат в соответствии с распределением заказов

Next j

If sum\_zatr < min\_zatr Then 'Запоминается лучший вариант из рассмотренных

For j = 1 To kol

opt\_nazn(j) = nazn(j)

Next j

min\_zatr = sum\_zatr

End If

Next i

MsgBox ("Оптимальное решение")

For j = 1 To kol

MsgBox ("Предприятие " + Str(j) + " выполняет заказ " + Str(opt\_nazn(j)))

Next j

MsgBox ("Минимальные затраты: " + Str(min\_zatr) + " ден.ед.")

End Sub

Примечание. Полученное решение близко к точному оптимальному решению задачи. Оптимальное решение, найденное специальным методом для решения задачи о назначениях (метод Мака), следующее: предприятию П1 выделяется четвертый заказ, П2 – второй, П3 – первый, П4 – третий. Общие затраты при этом составят 33 ден.ед. При использовании программной реализации предложенного алгоритма (см. выше), где выполняется 100 испытаний, определяется и точное оптимальное решение.

По результатам выполнения программы получено следующее решение: предприятию П1 выделяется четвертый заказ, П2 – второй, П3 – первый, П4 – третий. Общие затраты при этом составят 33 ден.ед. Это решение совпадает с точным решением, найденным методом Мака.

**Пример 4.3.** Между тремя предприятиями (П1,П2,П3) распределяется сумма в размере 5 млн ден.ед. Средства выделяются в размерах, кратных 1 млн ден.ед. Прибыль, которая может быть получена предприятиями в зависимости от выделенных средств, приведена в табл.4.5.

Таблица 4.5

Вложенные средства, млн ден.ед.		1	2	3	4	5
Прибыль предприятия, млн ден.ед.	П1	2	4	7	9	10
	П2	2	3	6	8	11
	П3	3	4	5	9	11

Например, если предприятию П1 будет выделено 2 млн ден.ед., то его прибыль от использования этих средств составит 4 млн ден.ед.

Требуется распределить имеющуюся сумму таким образом, чтобы общая прибыль, полученная предприятиями, была максимальной.

Для такой задачи можно найти точное решение, используя метод динамического программирования. Однако при большой размерности задачи, т.е. при большом количестве предприятий и большом объеме распределяемых средств, применение этого метода становится достаточно сложным.

Решим данную задачу, используя метод Монте-Карло. Для случайного выбора величины средств, выделяемых предприятию, будет использоваться формула (3.1).

Алгоритм решения рассматриваемой задачи на основе метода Монте-Карло реализуется в следующем порядке.

1. Разыгрывается (т.е. случайным образом выбирается) величина средств, выделяемых первому предприятию. Для этого разыгрывается случайное число  $R_1$  и используется формула (3.1) при  $A=0, B=5: S_1=[0+(5-0+1) \cdot R_1]$ .

2. Если первому предприятию выделены не все средства (т.е.  $S_1 < 5$ ), то разыгрывается величина средств, выделяемых второму предприятию. Для этого разыгрывается СРРЧ  $R_2$  и используется формула (3.1) при  $A=0, B=5-S_1, S_2=[0+(5-S_1-0+1) \cdot R_2]$ .

3. Оставшиеся средства (если они есть) выделяются третьему предприятию:  $S_3=5-S_1-S_2$ .

4. По величинам выделенных средств ( $S_1, S_2, S_3$ ) и по соответствующим значениям прибыли (табл.4.5) определяется суммарная прибыль предприятий.

Шаги 1-4 повторяются многократно (например, 1000 раз). Выбирается вариант распределения средств, для которого значение суммарной прибыли максимально.

Приведем пример десяти испытаний данного алгоритма (табл.4.6).

Таблица 4.6

Номер испытания	$R_1$	$S_1$	$R_2$	$S_2$	$S_3$	Прибыль
1	0,0795	0	0,3780	2	3	0+3+5=8
2	0,0593	0	0,7602	4	1	0+8+3=11
3	0,2847	1	0,8197	4	0	2+8+0=10
4	0,6133	3	0,5766	1	1	7+2+3=12
5	0,9595	5	-	0	0	10+0+0=10
6	0,0981	0	0,2410	1	4	0+2+9=11
7	0,5962	3	0,2978	0	2	7+0+4=11
8	0,6458	3	0,9762	2	0	7+3+0=10
9	0,0523	0	0,4523	2	3	0+3+5=8
10	0,8153	4	0,4286	0	1	9+0+3=12

Здесь, например, в первом испытании  $S_1=[0+(5-0+1) \cdot 0,0795]=0$ ,  $S_2=[0+(5-0+1) \cdot 0,3780]=2$ ,  $S_3=5-0-2=3$ . В этом испытании предприятию П1 средства не выделяются, предприятию П2 выделяется 2 млн ден.ед., предприятию П3 – оставшиеся 3 млн ден.ед. Из табл.4.6 видно, что при таком распределении средств предприятие П1 не получит прибыли, П2 – получит прибыль в размере 3 млн ден.ед., П3 – прибыль в размере 5 млн ден.ед. Таким образом, общая прибыль предприятий составит 8 млн ден.ед.

Как видно из табл.4.6, по результатам десяти испытаний найдены два варианта решения, обеспечивающие максимальную прибыль. Первый вариант - выделить предприятию П1 3 млн ден.ед., П2 - 1 млн, П3 - 1 млн. Второй вариант – выделить предприятию П1 4 млн ден.ед., П3 - 1 млн (предприятию П2 средства не выделяются). Общая прибыль предприятий (для любого из этих вариантов решения) составит 12 млн ден.ед.

Примечание. Это решение совпадает с точным оптимальным решением, найденным методом динамического программирования.

Приведем программную реализацию рассмотренного алгоритма на языке VBA. Основные константы и переменные программы: kol - количество предприятий; pp - прибыль предприятий, которую они могут получить в зависимости от выделенных средств (табл.4.5); s - массив для разыгрываемых вариантов решений (элементы этого массива s(1), s(2), s(3)- величины  $S_1$ ,  $S_2$ ,  $S_3$  соответственно); prtb - суммарная прибыль предприятий, соответствующая разыгранному варианту; s\_opt - массив для хранения оптимального варианта распределения средств; max\_prtb - максимальная прибыль (соответствующая оптимальному решению); a и b - имеют тот же смысл, что в формуле (3.1): a=0, b – имеющаяся (подлежащая распределению) сумма.

Sub prmer43()

‘Распределение средств между предприятиями

Dim pp(1 To 3, 1 To 5), s(1 To 3), s\_opt(1 To 3)

n = 1000

```

kol = 3: sredstva = 5
pp(1, 1) = 2: pp(1, 2) = 4: pp(1, 3) = 7: pp(1, 4) = 9: pp(1, 5) = 10
pp(2, 1) = 2: pp(2, 2) = 3: pp(2, 3) = 6: pp(2, 4) = 8: pp(2, 5) = 11
pp(3, 1) = 3: pp(3, 2) = 4: pp(3, 3) = 5: pp(3, 4) = 9: pp(3, 5) = 11
max_prib = 0
For i = 1 To n
a = 0: b = sredstva: prib = 0
For j = 1 To kol
If b > 0 Then
s(j) = Int(a + (b - a + 1) * Rnd)
b = b - s(j)
Else
s(j) = 0
End If
Next j
For j = 1 To kol
k = s(j)
If k > 0 Then prib = prib + pp(j, k)
Next j
If prib > max_prib Then
For j = 1 To kol
s_opt(j) = s(j)
Next j
max_prib = prib
End If
Next i
MsgBox ("Распределение средств: ")
For j = 1 To kol
MsgBox ("Предприятие " + Str(j) + ": " + Str(s_opt(j)) + " млн ден.ед.")
Next j
MsgBox ("Максимальная прибыль: " + Str(max_prib))
End Sub

```

**Пример 4.4.** Два речных судна (С1 и С2) используются для перевозки грузов в два населенных пункта (П1 и П2). В течение года в населенный пункт П1 необходимо доставить не менее 3 тыс.т груза, в пункт П2 – не менее 4 тыс.т. Судно С1 перевозит за один рейс 12 т груза, судно С2 – 15 т. Затраты, связанные с одним рейсом судна С1 в пункт П1, составляют 400 ден.ед., судна С1 в пункт П2 – 150 ден.ед., судна С2 в пункт П1 – 500 ден.ед., судна С2 в пункт П2 – 300 ден.ед. Каждое судно может выполнить в течение года не более 300 рейсов.

Требуется составить оптимальный план перевозок, т.е. определить, сколько рейсов должно выполнить каждое судно в каждый населенный пункт, чтобы доставить необходимое количество груза с минимальными затратами.

Решим данную задачу, используя метод Монте-Карло. Для выбора количества рейсов каждого судна в каждый населенный пункт будет применяться формула (3.1).

Алгоритм решения рассматриваемой задачи на основе метода Монте-Карло реализуется в следующем порядке.

1. Разыгрывается количество рейсов судна С1 в пункт П1. Для этого разыгрывается случайное число  $R_1$  и используется формула (3.1) при  $A=0$ ,  $B=300$ :  $N_{C1П1}=[0+(300-0+1) \cdot R_1]$ .

2. Разыгрывается количество рейсов судна С1 в пункт П2. Для этого разыгрывается случайное число  $R_2$  и используется формула (3.1) при  $A=0$ ,  $B=300-N_{C1П1}$ :  $N_{C1П2}=[0+(300-N_{C1П1}-0+1) \cdot R_2]$ .

3. Разыгрывается количество рейсов судна С2 в пункт А. Для этого разыгрывается случайное число  $R_3$  и используется формула (3.1) при  $A=0$ ,  $B=300$ :  $N_{C2П1}=[0+(300-0+1) \cdot R_3]$ .

4. Разыгрывается количество рейсов судна С2 в пункт В. Для этого разыгрывается случайное число  $R_4$  и используется формула (3.1) при  $A=0$ ,  $B=300-N_{C2П1}$ :  $N_{C2П2}=[0+(300-N_{C2П1}-0+1) \cdot R_4]$ .

5. Вычисляется количество груза, перевозимого в пункты П1 и П2 при выбранном плане перевозок ( $G_{П1}$  и  $G_{П2}$ ). Если  $G_{П1} \geq 3000$  и  $G_{П2} \geq 4000$ , то выбранный план перевозок является допустимым и для него вычисляются затраты.

Шаги 1-5 выполняются многократно (например, 5 млн раз). Выбирается план перевозок с минимальными затратами.

Приведем пример одного испытания данного алгоритма.

Разыграем количество рейсов судна С1 в пункт П1. Пусть разыграно СРРЧ  $R_1=0,0795$ , тогда

$N_{C1П1}=[0+(300-0+1) \cdot 0,0795]=23$  рейса. Так как судно С1 может выполнить не более 300 рейсов в год, в пункт П2 оно может выполнить не более  $300-23=277$  рейсов. Разыграем количество рейсов судна С1 в пункт П2: пусть  $R_2=0,378$ , тогда  $N_{C1П2}=[0+(277-0+1) \cdot 0,378]=105$  рейсов.

Разыграем количество рейсов судна С2 в пункт П1. Пусть разыграно СРРЧ  $R_3=0,0593$ , тогда

$N_{C2П1}=[0+(300-0+1) \cdot 0,0593]=17$  рейсов. Разыграем количество рейсов судна С2 в пункт П2: пусть  $R_4=0,7602$ , тогда  $N_{C2П2}=[0+(283-0+1) \cdot 0,7602]=215$  рейсов.

Вычислим количество груза, доставляемого в каждый из пунктов. В пункт П1 доставляется  $12 \cdot 23 + 15 \cdot 17 = 531$  т груза, в пункт П2 -  $12 \cdot 105 + 215 \cdot 17 = 4915$  т. Так как в пункт П1 доставляется менее 3000 т груза, полученный план перевозок недопустим.

Примечание. Для решения данной задачи требуется большое количество испытаний алгоритма, так как очень многие из разыгранных вариантов решения оказываются недопустимыми (не удовлетворяют ограничениям на количество перевозимых грузов).

Приведем программную реализацию рассмотренного алгоритма.

```
Sub primer44()
```

```
'Составление плана перевозок в два населенных пункта
```

```
n = 5000000 'Количество испытаний
```

```
potr_p1 = 3000: potr_p2 = 4000 'Потребности населенных пунктов
```

```
gc1 = 12: gc2 = 15 'Вместимость судов
```

```
zc1p1 = 400: zc1p2 = 150: zc2p1 = 500: zc2p2 = 300 'Затраты на один рейс
```

```
minzatr = 1000000
```

```
For i = 1 To n
```

```
reis = 300
```

```
r = Rnd: nc1p1 = Int((reis + 1) * r): reis = reis - nc1p1 'Розыгрыш количества рейсов для С1
```

```
r = Rnd: nc1p2 = Int((reis + 1) * r)
```

```
reis = 300
```

```

r = Rnd: nc2p1 = Int((reis + 1) * r): reis = reis - nc2p1 'Розыгрыш количества рейсов для С2
r = Rnd: nc2p2 = Int((reis + 1) * r)
zatr = nc1p1 * zc1p1 + nc1p2 * zc1p2 + nc2p1 * zc2p1 + nc2p2 * zc2p2 'Затраты на перевозки
gruzp1 = nc1p1 * gc1 + nc2p1 * gc2:рузp2 = nc1p2 * gc1 + nc2p2 * gc2 'Количество груза,
'доставляемого в П1 и П2
If (gruzp1 >= potr_p1) And (gruzp2 >= potr_p2) And (zatr < minzatr) Then
minzatr = zatr 'Запоминается лучший вариант из рассмотренных
nc1p1opt = nc1p1: nc1p2opt = nc1p2: nc2p1opt = nc2p1: nc2p2opt = nc2p2
End If
Next i
MsgBox ("Судно С1: П1 - " + Str(nc1p1opt) + " П2 - " + Str(nc1p2opt))
MsgBox ("Судно С2: П1 - " + Str(nc2p1opt) + " П2 - " + Str(nc2p2opt))
MsgBox ("Затраты: " + Str(minzatr))
End Sub

```

По результатам выполнения программы получено следующее решение: судно С1 выполняет 4 рейса в пункт П1 и 289 рейсов в пункт П2; судно С2 выполняет 197 рейсов в пункт П1 и 36 рейсов в пункт П2. Затраты на перевозки составят 154 250 ден.ед.

**Пример 4.5.** Крупное предприятие предполагает построить несколько жилых домов для своих сотрудников. Всего на строительство выделено 10 млн ден.ед. Имеются три проекта жилых домов (ПР1, ПР2, ПР3). Их характеристики приведены в табл.4.7.

Таблица 4.7

Проект	ПР1	ПР2	ПР3
Жилая площадь, тыс м <sup>2</sup>	2,5	2	4
Стоимость, млн ден.ед.	2	1,5	2,8

Требуется определить, сколько домов следует построить по каждому из проектов, чтобы их общая жилая площадь была максимальной.

Алгоритм решения такой задачи на основе метода Монте-Карло реализуется в следующем порядке.

1. Определяется максимальное количество домов, которое можно построить по каждому проекту в соответствии с выделенными средствами:  $N_1 = \lfloor 10/2 \rfloor = 5$ ,  $N_2 = \lfloor 10/1,5 \rfloor = 6$ ,  $N_3 = \lfloor 10/2,8 \rfloor = 3$ .

2. Разыгрывается количество домов, которые следует построить по проекту ПР1 ( $N_{ПР1}$ ). Для этого используется формула (3.1) при  $A=0$ ,  $B=N_1=5$  (так как 5 - максимально возможное количество домов по данному проекту).

3. Разыгрывается количество домов, которые следует построить по проекту ПР2 ( $N_{ПР2}$ ). Для этого используется формула (3.1) при  $A=0$ ,  $B=6$ .

4. Разыгрывается количество домов, которые следует построить по проекту ПР3 ( $N_{ПР3}$ ). Для этого используется формула (3.1) при  $A=0$ ,  $B=3$ .

5. Находятся общие затраты на постройку всех домов, выбранных на шагах 2-4. Если эти затраты не превышают 10 млн ден.ед., то выбранный план строительства является допустимым; вычисляется общая жилая площадь домов, выбранных на шагах 2-4.

Шаги 2-5 повторяются многократно. Выбирается план строительства, для которого общая жилая площадь построенных домов максимальна.

Приведем пример одного испытания данного алгоритма. Разыграем количество домов, которые следует построить по проекту ПР1. Пусть разыграно СРРЧ  $R_1=0,0795$ , тогда  $N_{\text{ПР1}}=[0+(5-0+1)\cdot 0,0795]=0$ , т.е. предлагается не строить дома по проекту ПР1. Аналогично разыграем количество домов для постройки по проектам ПР2 и ПР3. Пусть  $R_2=0,378$ , тогда  $N_{\text{ПР2}}=[0+(6-0+1)\cdot 0,3780]=2$  дома. Пусть  $R_3=0,0593$ , тогда  $N_{\text{ПР3}}=[0+(3-0+1)\cdot 0,0593]=0$ . Затраты на строительство составят  $0\cdot 2+2\cdot 1,5+0\cdot 2,8=3$  млн ден.ед. Так как затраты не превышают выделенных средств (10 млн ден.ед.), план строительства является допустимым. Жилая площадь домов, построенных по этому плану, составит  $0\cdot 2,5+2\cdot 2+0\cdot 4=4$  тыс м<sup>2</sup>.

Примечание. Как и в предыдущем примере, для решения данной задачи требуется большое количество испытаний, так как значительная часть разыгрываемых вариантов решений оказываются недопустимыми.

Sub primer45()

‘Составление оптимального плана строительства

$n1 = 5: n2 = 6: n3 = 3$

$n = 5000000$  ‘Количество испытаний

$zatrmax = 10$

$plmax = 0$

For  $i = 1$  To  $n$

$r1 = \text{Rnd}: npr1 = \text{Int}((n1 + 1) * r1)$  ‘Розыгрыш количества домов по каждому из проектов

$r2 = \text{Rnd}: npr2 = \text{Int}((n2 + 1) * r2)$

$r3 = \text{Rnd}: npr3 = \text{Int}((n3 + 1) * r3)$

$pl = 2.5 * npr1 + 2 * npr2 + 4 * npr3$  ‘Вычисление жилой площади

$zatr = 2 * npr1 + 1.5 * npr2 + 2.8 * npr3$  ‘Вычисление затрат

If ( $zatr \leq zatrmax$ ) And ( $pl > plmax$ ) Then ‘Запоминается лучший вариант из рассмотренных

$plmax = pl: zatropt = zatr: npr1opt = npr1: npr2opt = npr2: npr3opt = npr3$

End If

Next  $i$

MsgBox ("Построить по проекту 1: " + Str( $npr1opt$ ))

MsgBox ("Построить по проекту 2: " + Str( $npr2opt$ ))

MsgBox ("Построить по проекту 3: " + Str( $npr3opt$ ))

MsgBox ("Жилая площадь: " + Str( $plmax$ ) + " Затраты: " + Str( $zatropt$ ))

End Sub

По результатам выполнения программы получено следующее решение: по проекту ПР1 строить дома не следует, по проекту ПР2 – построить один дом, по проекту ПР3 - три дома. Общая жилая площадь построенных домов – 14 тыс м<sup>2</sup>. Затраты на строительство составят 9,9 млн ден.ед.

**Пример 4.6.** Для транспортировки некоторого химиката требуется изготовить контейнеры. Требования к контейнерам следующие: 1) емкость контейнера - 6 м<sup>3</sup>; 2) высота может составлять от 1 до 3 м; 3) основание контейнера должно быть квадратным. Дно и стенки контейнера, непосредственно соприкасающиеся с химикатом, должны быть изготовлены из более стойкого материала, чем крышка контейнера. Цена материала дна и стенок контейнера – 6 ден.ед./м<sup>2</sup>, цена материала крышки - 4 ден.ед./м<sup>2</sup>. Требуется найти габаритные размеры контейнера (размеры основания и высоту), при которых его стоимость будет минимальной.

Обозначим высоту контейнера как  $H$ , размеры его основания (длину и ширину) - как  $L$ , стоимость материала для контейнера – как  $C$ . Тогда для данной задачи можно построить следующую математическую модель:

$$H \geq 1;$$

$$H \leq 3;$$

$$L = \sqrt{6/H};$$

$$C = 6 \cdot L^2 + 24 \cdot H \cdot L + 4 \cdot L^2 \rightarrow \min.$$

Здесь первое и второе ограничения устанавливают, что высота контейнера должна составлять от 1 до 3 м; третье ограничение устанавливает размер основания (из условия, что емкость контейнера равна 6 м<sup>3</sup>). Целевая функция  $C$  выражает стоимость контейнера (первое слагаемое - стоимость материала основания, второе - стоимость материала стенок, третье - крышки). Данная задача представляет собой задачу нелинейного программирования (в ней нелинейными являются целевая функция и одно из ограничений). Точных методов для решения таких задач нет. Найдем приближенное решение, используя метод Монте-Карло.

1. Определяется (выбирается случайным образом) высота контейнера. Для этого разыгрывается СРРЧ  $R$ . Так как высота контейнера должна принимать значение в заданном диапазоне (от 1 до 3 м), для ее розыгрыша воспользуемся формулой (3.11) при  $a=1$ ,  $b=3$ :  $H=1+(3-1)R$ .

2. Находится размер основания контейнера:  $L = \sqrt{6/H}$ .

3. Для выбранных размеров контейнера ( $H$  и  $L$ ) находится стоимость материала:  $C = 6 \cdot L^2 + 24 \cdot H \cdot L + 4 \cdot L^2$ .

Шаги 1-3 повторяются многократно. Выбираются размеры контейнера, при которых стоимость материала минимальна.

В табл.4.8 приводится пример десяти испытаний данного алгоритма.

Таблица 4.8

Номер испытания	R	H	L	C
1	0,0795	1,16	2,28	115,06
2	0,3780	1,76	1,85	112,07
3	0,0593	1,12	2,32	115,81
4	0,7602	2,52	1,54	117,14
5	0,2847	1,57	1,96	111,88
6	0,8197	2,64	1,51	118,24
7	0,6133	2,23	1,64	114,67
8	0,5766	2,15	1,67	114,13
9	0,9595	2,92	1,43	120,99
10	0,0981	1,20	2,24	114,46

Таким образом, по результатам десяти испытаний минимальная стоимость контейнера обеспечивается при следующих размерах: высота - 1,57 м, размер основания - 1,96 м. Таким образом, следует выпускать контейнеры размерами 1,96 x 1,96 x 1,57 м. Стоимость материала для такого контейнера составит 111,88 ден.ед.

Приведем программную реализацию алгоритма решения задачи.

Sub primer46()

‘Определение оптимальных размеров контейнера

hmin = 1: hmax = 3

n = 5000000 ‘Количество испытаний

cmin = 1000

For i = 1 To n

r = Rnd

h = hmin + (hmax - hmin) \* r

‘Розыгрыш высоты

l = Sqr(6 / h)

‘Вычисление размера основания

c = 6 \* l ^ 2 + 24 \* h \* l + 4 \* l ^ 2

‘Вычисление стоимости

```
If c < cmin Then                                ‘Запоминается лучший вариант из рассмотренных
cmin = c : lopt = l : hopt = h
End If
Next i
msgbox(“Оптимальные размеры:
“+Str(Round(lopt,2))+”x”+Str(Round(lopt,2))+”x”+Str(Round(hopt,2)))
msgbox(“Стоимость: “+Str(Round(cmin,2)))
End Sub
```

По результатам выполнения программы получено следующее решение: минимальная стоимость контейнера обеспечивается при размерах 1,93 x 1,93 x 1,61 м. Стоимость материала для такого контейнера составит 111,86 ден.ед.

Примечание. Это решение совпадает с точным оптимальным решением, найденным методом нелинейного программирования. Следует обратить внимание, что решение, найденное по результатам всего десяти испытаний (табл.4.8), практически совпадает с оптимальным. Алгоритм решения данной задачи на основе метода Монте-Карло значительно проще, чем алгоритм решения на основе методов нелинейного программирования.

## ЛИТЕРАТУРА

### *Основная*

1. Таха Х. Введение в исследование операций. М.: Издательский дом “Вильямс”, 2001. – 912 с.
2. Харин Ю.С., Малюгин В.И., Кирлица В.П. и др. Основы имитационного и статистического моделирования. Мн.: Дизайн ПРО, 1997. – 288 с.
3. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Высш. шк., 2001. – 342 с.
4. Советов Б.Я., Яковлев С.А. Моделирование систем. Практикум. М.: Высш. шк., 1999. – 224 с.
5. Кнут Д.Э. Искусство программирования. В 3 т. Том 2. Получисленные алгоритмы. М.: Издательский дом “Вильямс”, 2003. – 832 с.
6. Соболев И.М. Метод Монте-Карло. М.: Наука, 1985. – 80 с.
7. Эддоус М., Стэнфилд Р. Методы принятия решений. М.: Юнити, 1997. – 590 с.
8. Большаков А.С. Моделирование в менеджменте. М.: Филинь, 2000. – 464 с.
9. Рыжиков Ю.И. Имитационное моделирование. Теории и технологии. СПб.: КОРОНА принт, 2004. – 384 с.

### *Дополнительная*

10. Томашевский В.Н., Жданова Е.Г. Имитационное моделирование в среде GPSS. М.: Бестселлер, 2003. – 416 с.
11. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004. – 320 с.
12. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II. М.: Мир, 1987. – 646 с.
13. Марков А.А. Моделирование информационно-вычислительных процессов. М.: Изд-во МГТУ им. Н.Э. Баумана, 1999. – 360 с.
14. Гулятьев А.К. MATLAB 5.3. Имитационное моделирование в среде Windows. СПб.: КОРОНА принт, 2001. – 400 с.
15. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. М.: Наука, 1988. – 480 с.
16. Смородинский С.С., Батин Н.В. Оптимизация решений на основе методов и моделей математического программирования: Учеб. пособие по курсу "Системный анализ и исследование операций" для студентов специальности АСОИ. Мн.: БГУИР, 2003. – 136 с.

# ПРИЛОЖЕНИЕ 1

## СЛУЧАЙНЫЕ РАВНОМЕРНО РАСПРЕДЕЛЕННЫЕ ЧИСЛА

0,0795	0,3780	0,0593	0,7602	0,2847	0,8197	0,6133	0,5766	0,9595	0,0981
0,2410	0,5962	0,2978	0,6458	0,9762	0,0523	0,4523	0,8153	0,4286	0,8400
0,8754	0,5900	0,3421	0,7919	0,1204	0,7492	0,3792	0,7198	0,4822	0,8215
0,6058	0,5321	0,8064	0,0540	0,9046	0,5778	0,0043	0,6218	0,0563	0,5934
0,3716	0,3549	0,0112	0,8438	0,3233	0,5052	0,7647	0,2125	0,6382	0,5239
0,0644	0,7612	0,5707	0,2759	0,8396	0,9127	0,7652	0,5576	0,2914	0,5682
0,7195	0,1977	0,2245	0,3979	0,0652	0,0136	0,2148	0,8663	0,7802	0,9190
0,5994	0,8691	0,0564	0,9681	0,7945	0,1137	0,3537	0,8137	0,8484	0,4385
0,9561	0,2147	0,9299	0,1944	0,0147	0,1935	0,1453	0,2754	0,0517	0,7645
0,4403	0,1689	0,0453	0,1344	0,5162	0,0540	0,2084	0,3725	0,3741	0,2042
0,2058	0,0720	0,2383	0,7248	0,2661	0,1888	0,8946	0,1636	0,9949	0,3229
0,5869	0,5937	0,3465	0,9940	0,1708	0,3062	0,1309	0,6818	0,9790	0,7400
0,3931	0,0718	0,7680	0,9890	0,0435	0,9811	0,5677	0,5516	0,0583	0,5912
0,7759	0,5792	0,0555	0,6824	0,2781	0,5160	0,6977	0,1059	0,5570	0,6978
0,8090	0,4302	0,6869	0,4293	0,9700	0,2145	0,2849	0,8945	0,7895	0,3605
0,4983	0,8773	0,4004	0,2839	0,1111	0,6815	0,6258	0,8406	0,8576	0,4549
0,4122	0,4560	0,2445	0,9876	0,3876	0,1894	0,1342	0,6081	0,1433	0,6191
0,3483	0,5023	0,4718	0,5838	0,5971	0,1554	0,5865	0,8211	0,5267	0,0068
0,3178	0,5673	0,8289	0,7917	0,3653	0,8804	0,0996	0,4143	0,7519	0,4857
0,1757	0,2741	0,5951	0,0243	0,4258	0,4118	0,9968	0,7917	0,6760	0,7910
0,6016	0,2608	0,2722	0,4435	0,3890	0,8999	0,8378	0,0933	0,0203	0,5364
0,5291	0,7498	0,4549	0,8937	0,9365	0,1678	0,6006	0,1885	0,5667	0,5623
0,4291	0,9307	0,8907	0,4776	0,1691	0,6589	0,9766	0,8555	0,2026	0,5771
0,9220	0,9723	0,3733	0,3401	0,8118	0,0978	0,0913	0,7856	0,6641	0,5063
0,3362	0,8619	0,7735	0,7002	0,4086	0,6974	0,6014	0,3553	0,5268	0,4749
0,2861	0,1074	0,8348	0,1517	0,0206	0,2963	0,6904	0,0079	0,5070	0,3674
0,7993	0,6758	0,1273	0,5915	0,0321	0,0516	0,6761	0,0542	0,5205	0,2080
0,2886	0,2257	0,8240	0,6699	0,2291	0,5493	0,5030	0,6703	0,3374	0,8210
0,1792	0,4387	0,6405	0,7603	0,1866	0,4258	0,3329	0,2838	0,8122	0,2138
0,6629	0,2038	0,0407	0,4089	0,0196	0,8947	0,4352	0,1265	0,2156	0,4300

## ПРИЛОЖЕНИЕ 2

### РАБОТА С СИСТЕМОЙ ПРОГРАММИРОВАНИЯ VISUAL BASIC FOR APPLICATIONS

Рассмотрим основные этапы подготовки программ (макросов) в среде системы программирования VBA (Visual Basic for Applications). Язык программирования VBA является диалектом универсального языка программирования Visual Basic. Язык VBA встроен во все приложения пакета Microsoft Office (текстовый редактор Word, табличный процессор Excel и т.д.).

1. Загрузить Word или Excel. Создать файл. Сохранить его под каким-либо именем.
  2. Выбрать команду **Сервис – Макрос – Макросы**.
  3. В появившемся окне **Макрос** в поле **Макросы из** (Word) или **Находится в** (Excel) выбрать имя файла, созданного на шаге 1.
  4. В поле **Имя** (Word) или **Имя макроса** (Excel) указать имя создаваемой программы.
  5. Нажать кнопку **Создать**. На экран выводится окно системы программирования VBA.
  6. Ввести исходный текст программы.
  7. Сохранить файл с созданной программой, используя команду **Файл – Сохранить** (File - Save).
  8. Чтобы запустить программу на выполнение, выбрать команду **Запуск – Запуск подпрограммы** (Run – Run Sub/UserForm).
- При работе с программами, представляющими собой имитационные модели, обычно желательно, чтобы в каждом сеансе работы с программой (т.е. при каждом ее запуске на выполнение) использовались одни и те же случайные числа (СРРЧ). Для этого необходимо *перед каждым запуском программы* выполнять команду **Запуск – Сброс** (Run – Reset).
9. По окончании работы с программой закрыть окно VBA.
  10. Выйти из Word или Excel.

Чтобы воспользоваться программой, созданной ранее, необходимо выполнить следующее.

1. Загрузить файл Word или Excel, в котором была подготовлена программа.
2. Выбрать команду **Сервис – Макрос – Макросы**.
3. В появившемся окне **Макрос** в поле **Макросы из** (Word) или **Находится в** (Excel) выбрать имя загруженного файла.
4. В поле **Имя** (Word) или **Имя макроса** (Excel) выбрать имя программы, которую требуется запустить на выполнение или изменить.
5. Нажать кнопку **Изменить**. На экран выводится окно системы программирования VBA с текстом программы. После этого можно внести в программу желаемые изменения или запустить ее на выполнение.

## ПРИЛОЖЕНИЕ 3

### НЕКОТОРЫЕ СВЕДЕНИЯ О ВЕРОЯТНОСТНЫХ ХАРАКТЕРИСТИКАХ СЛУЧАЙНЫХ ВЕЛИЧИН

**Дискретные случайные величины** задаются перечислением их возможных значений  $x_1, x_2, \dots, x_n$  и вероятностей этих значений  $P_1, P_2, \dots, P_n$ . Если количество возможных значений случайной величины  $n$  достаточно велико (или бесконечно), то значения случайной величины и их вероятности задаются формулами (законом распределения случайной величины).

Для описания **непрерывных случайных величин** обычно используются две функции: функция распределения  $F(x)$  и плотность распределения (или плотность вероятности)  $f(x)$ .

Примечание. Случайные величины обычно обозначаются заглавными буквами (например  $X$ ), а их конкретные значения - такими же строчными буквами ( $x$ ).

**Функция распределения  $F(x)$**  случайной величины  $X$  имеет следующий смысл:

$$F(x) = P(X < x). \quad (\text{П3.1})$$

Таким образом, значение функции распределения  $F(x)$  от конкретного числа  $x$  – это вероятность того, что случайная величина  $X$  примет значение меньше  $x$ .

**Плотность распределения  $f(x)$**  и функция распределения  $F(x)$  связаны следующими формулами:

$$f(x) = \frac{dF(x)}{dx}; \quad (\text{П3.2})$$

$$F(x) = \int_{-\infty}^x f(x) dx. \quad (\text{П3.3})$$

Примечание. Функция распределения  $F(x)$  может быть задана как для непрерывной, так и для дискретной случайной величины  $X$ . Плотность распределения обычно задается для непрерывных случайных величин.

Зная функцию распределения или плотность распределения случайной величины  $X$ , можно вычислить вероятность попадания этой величины в любой диапазон значений  $[a; b)$ . Для этого используются следующие формулы:

$$P(a \leq X < b) = F(b) - F(a), \quad (\text{П3.4})$$

или

$$P(a \leq X < b) = \int_a^b f(x) dx. \quad (\text{П3.5})$$

Таким образом, вероятность попадания случайной величины  $X$  в некоторый диапазон  $[a; b)$  равна разности значений функции распределения на

границах этого диапазона, или интегралу от плотности распределения в этом диапазоне.

**Пример ПЗ.1.** В условиях примера 3.2, используя функцию распределения времени внесения добавок, найти вероятность того, что время внесения добавок составит: а) менее 9 минут; б) от 5 до 9 минут; в) более 8 минут.

Приведем еще раз функцию распределения времени внесения добавок из примера 3.2:

$$F(x) = \begin{cases} 0, & x < 3; \\ 0,1x - 0,3, & 3 \leq x < 6; \\ 0,15x - 0,6, & 6 \leq x < 10; \\ 0,05x + 0,4, & 10 \leq x < 12; \\ 1, & x \geq 12; \end{cases}$$

а) вероятность того, что время внесения добавок составит менее 9 минут, найдем по формуле (ПЗ.1):

$$P(X < 9) = F(9) = 0,15 \cdot 9 - 0,6 = 0,75;$$

б) вероятность того, что время внесения добавок составит от 5 до 9 минут, найдем по формуле (ПЗ.4):

$$P(5 \leq X < 9) = F(9) - F(5) = (0,15 \cdot 9 - 0,6) - (0,1 \cdot 5 - 0,3) = 0,55;$$

в) вероятность того, что время внесения добавок составит более 8 минут, найдем через вероятность противоположного события (т.е. того, что время внесения добавок составит менее 8 минут), используя формулу (ПЗ.1):

$$P(X \geq 8) = 1 - P(X < 8) = 1 - F(8) = 1 - (0,15 \cdot 8 - 0,6) = 0,4.$$

**Пример ПЗ.2.** Найти те же вероятности, что и в примере ПЗ.1, используя плотность распределения времени внесения добавок.

Приведем еще раз плотность распределения времени внесения добавок из примера 3.2:

$$f(x) = \begin{cases} 0, & x < 3; \\ 0,1, & 3 \leq x < 6; \\ 0,15, & 6 \leq x < 10; \\ 0,05, & 10 \leq x < 12; \\ 0, & x \geq 12. \end{cases}$$

Для вычисления вероятностей воспользуемся формулой (ПЗ.5):

а) вероятность того, что время внесения добавок составит менее 9 минут:

$$P(X < 9) = \int_{-\infty}^9 f(x) dx = \int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^9 0,15 dx = 0,75;$$

б) вероятность того, что время внесения добавок составит от 5 до 9 минут:

$$P(5 \leq X < 9) = \int_5^9 f(x) dx = \int_5^6 0,1 dx + \int_6^9 0,15 dx = 0,55;$$

в) вероятность того, что время внесения добавок составит более 8 минут:

$$P(X \geq 8) = \int_8^{\infty} f(x) dx = \int_8^{10} 0,15 dx + \int_{10}^{12} 0,05 dx + \int_{12}^{\infty} 0 dx = 0,4.$$

Кроме того, в практических задачах обычно используются **числовые характеристики** случайных величин. Чаще всего требуется использовать математическое ожидание, дисперсию, стандартное отклонение, коэффициент вариации.

**Математическое ожидание** характеризует среднее значение случайной величины. Оно вычисляется по следующей формуле:

- для дискретных случайных величин:

$$M[X] = \sum_{i=1}^n x_i P_i, \quad (\text{ПЗ.6})$$

где  $x_i$  – возможные значения дискретной случайной величины;

$P_i$  – вероятности этих значений;

$n$  – количество возможных значений дискретной случайной величины;

- для непрерывных случайных величин:

$$M[X] = \int_{-\infty}^{\infty} x \cdot f(x) dx, \quad (\text{ПЗ.7})$$

где  $f(x)$  – плотность распределения непрерывной случайной величины.

Дисперсия, стандартное отклонение и коэффициент вариации случайной величины характеризуют разброс ее значений, т.е. отклонение конкретных значений случайной величины от ее среднего значения (математического ожидания).

**Дисперсия** вычисляется следующим образом:

- для дискретных случайных величин:

$$D[X] = \sum_{i=1}^n (x_i - M[X])^2 P_i, \quad (\text{ПЗ.8})$$

- для непрерывных случайных величин:

$$D[X] = \int_{-\infty}^{\infty} (x - M[X])^2 f(x) dx, \quad (\text{ПЗ.9})$$

где  $M[X]$  – математическое ожидание случайной величины.

На практике во многих случаях дисперсия вычисляется по следующей формуле:

$$D[X] = M[X^2] - (M[X])^2, \quad (\text{ПЗ.10})$$

где  $M[X^2]$  – математическое ожидание квадрата случайной величины (второй начальный момент случайной величины), вычисляемое следующим образом:

- для дискретных случайных величин:

$$M[X^2] = \sum_{i=1}^n x_i^2 P_i, \quad (\text{ПЗ.11})$$

- для непрерывных случайных величин:

$$M[X^2] = \int_{-\infty}^{\infty} x^2 \cdot f(x) dx. \quad (\text{ПЗ.12})$$

Стандартное отклонение  $\sigma$  и коэффициент вариации  $\varepsilon$  вычисляются по следующим формулам:

$$\sigma = \sqrt{D[X]}; \quad (\text{ПЗ.13})$$

$$\varepsilon = \frac{\sigma}{M[X]}. \quad (\text{ПЗ.14})$$

**Пример ПЗ.3.** В условиях примера 2.2 найти математическое ожидание прибыли от ремонта одного изделия.

В примере 2.2 прибыль от ремонта одного изделия представляет собой дискретную случайную величину, которая может принимать значения 35 (с вероятностью 0,2), 60 (с вероятностью 0,4), 40 (с вероятностью 0,15) или 25 (с вероятностью 0,25). Для вычисления математического ожидания воспользуемся формулой (ПЗ.6):

$$M[X] = 35 \cdot 0,2 + 60 \cdot 0,4 + 40 \cdot 0,15 + 25 \cdot 0,25 = 43,25 \text{ ден.ед.}$$

Таким образом, математическое ожидание прибыли (т.е. *средняя* прибыль) от ремонта одного изделия составляет 43,25 ден.ед.

**Пример ПЗ.4.** В условиях примера 3.3 для времени внесения добавок найти математическое ожидание, дисперсию, стандартное отклонение и коэффициент вариации.

Найдем математическое ожидание по формуле (ПЗ.7):

$$\begin{aligned} M[X] &= \int_{-\infty}^3 0 x dx + \int_3^6 0,1x dx + \int_6^{10} 0,15x dx + \int_{10}^{12} 0,05x dx + \int_{12}^{\infty} 0x dx = \\ &= 0 + 0,1 \frac{x^2}{2} \Big|_3^6 + 0,15 \frac{x^2}{2} \Big|_6^{10} + 0,05 \frac{x^2}{2} \Big|_{10}^{12} + 0 = 7,25 \text{ мин.} \end{aligned}$$

Таким образом, *среднее* время внесения добавок при изготовлении пластмассовой плиты составляет 7,25 мин (хотя это время различно при изготовлении каждой конкретной плиты).

Чтобы вычислить дисперсию, воспользуемся формулой (ПЗ.10). Для этого сначала найдем второй начальный момент по формуле (ПЗ.12):

$$\begin{aligned} M[X^2] &= \int_{-\infty}^3 0 x^2 dx + \int_3^6 0,1x^2 dx + \int_6^{10} 0,15x^2 dx + \int_{10}^{12} 0,05x^2 dx + \int_{12}^{\infty} 0x^2 dx = \\ &= 0 + 0,1 \frac{x^3}{3} \Big|_3^6 + 0,15 \frac{x^3}{3} \Big|_6^{10} + 0,05 \frac{x^3}{3} \Big|_{10}^{12} + 0 = 57,63 \text{ мин}^2. \end{aligned}$$

Вычислим дисперсию по формуле (ПЗ.10):

$$D[X] = 57,63 - 7,25^2 = 5,07 \text{ мин}^2.$$

Найдем стандартное отклонение по формуле (ПЗ.13):

$$\sigma = \sqrt{D[X]} = 2,25 \text{ мин.}$$

Это означает, что конкретные значения времени внесения добавок отклоняются от математического ожидания (т.е. от 7,25 мин) в среднем на 2,25 мин.

Найдем коэффициент вариации по формуле (ПЗ.14):

$$\varepsilon = \frac{2,25}{7,25} = 0,31.$$

Можно сказать, что отклонения отдельных значений времени внесения добавок от математического ожидания составляют в среднем 31%.

Примечание. Для математического ожидания и дисперсии во многих случаях используются обозначения, отличающиеся от  $M[X]$  и  $D[X]$ .

## ПРИЛОЖЕНИЕ 4

### АЛГОРИТМЫ ИМИТАЦИИ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

В таблице приводятся алгоритмы имитации некоторых случайных величин, часто применяемых при решении задач моделирования. Алгоритм имитации гауссовского распределения рассмотрен в п.3.2.3. Остальные алгоритмы получены на основе метода обратных функций.

Распределение	Алгоритм имитации	Примечание
Экспоненциальное	$X = -\frac{1}{\lambda} \ln R$	$\lambda = 1/\bar{X}$ , где $\bar{X}$ - математическое ожидание
Гауссовское (нормальное)	$X = m + \sigma\sqrt{2}\left(\sum_{i=1}^6 R_i - 3\right)$	$m$ - математическое ожидание, $\sigma$ - стандартное отклонение
Равномерное	$X = a + (b-a)R$	$a$ и $b$ - границы возможных значений величины
Треугольное	$X = \begin{cases} a + \sqrt{R(b-a)(c-a)}, & 0 \leq R < (c-a)/(b-a) \\ b - \sqrt{(1-R)(b-a)(b-c)}, & (c-a)/(b-a) \leq R < 1 \end{cases}$	$a$ и $b$ - границы возможных значений величины, $c$ - мода (точка максимума плотности распределения) случайной величины
Эрланга $k$ -го порядка	$X = -\frac{1}{k\lambda} \sum_{i=1}^k R_i$	$\lambda$ - параметр распределения

## ПРИЛОЖЕНИЕ 5

### ПРИМЕРЫ ЗАДАЧ, РЕШАЕМЫХ НА ОСНОВЕ МЕТОДА МОНТЕ-КАРЛО

**Задача П5.1.** Пользователи работают с информационно-справочной системой (ИСС). Количество запросов, вводимых пользователем в течение сеанса работы с ИСС, может быть различным. В 10% случаев за время сеанса пользователь вводит один запрос, в 50% случаев - два запроса, в 30% случаев - три, в 10% - четыре. Каждый запрос включает следующие действия:

- ввод запроса пользователем (от 10 до 20 секунд);
- поиск ответа в ИСС и его вывод на экран (от 5 до 10 секунд);
- обдумывание ответа пользователем (экспоненциальная случайная величина со средним значением 1 мин).

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения средней длительности сеанса работы пользователя с ИСС.

**Задача П5.2.** В службе контроля качества продукции предприятия выполняется проверка изделий и (при необходимости) их мелкий ремонт. Изделия поступают к контролеру партиями по 30 штук. Сначала контролер проверяет все изделия в партии, затем - ремонтирует те изделия, для которых это требуется.

Время, необходимое контролеру для проверки одного изделия - гауссовская случайная величина со средним значением 5 мин и стандартным отклонением 1,5 мин.

В 20% случаев в партии не оказывается ни одного изделия, требующего ремонта. В 40% случаев требуется ремонт одного изделия, в 30% случаев - двух, в 10% случаев - трех изделий. Больше количество изделий, требующих ремонта, оказывается в партии крайне редко.

Ремонт одного изделия занимает от 10 до 15 минут.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения среднего времени обработки контролером одной партии изделий.

**Задача П5.3.** Предприятие выполняет переводы с иностранных языков. Размер перевода - случайная величина, которую можно считать распределенной по экспоненциальному закону. Средний размер перевода - 5 листов.

Из всех заказов переводы с английского языка составляют 60%, с немецкого - 25%, с французского - 5%, с других языков - 10%.

Кроме того, известно, что из всех заказов 50% составляют заказы на технические переводы, 40% - экономические и юридические, 10% - художественные. Каждый заказ относится только к одному виду (другими словами, перевод не может быть, например, одновременно и техническим, и художественным).

Плата, которую берет предприятие с заказчика за один лист технического перевода, следующая: с английского языка - 10 ден.ед., с немецкого - 12 ден.ед., с французского - 14 ден.ед., с других языков - 17 ден.ед. За экономический или юридический перевод плата повышается на 10% (по сравнению с техническим), за художественный - на 25%.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения средней выручки предприятия от выполнения одного перевода.

**Задача П5.4.** Некоторый механизм, помещенный в защитный кожух, подвергается ударным нагрузкам. Количество ударов в течение рабочей смены может составлять от 40 до 100.

Ударная нагрузка может действовать под углом от 0 до 90 градусов. Сила ударной нагрузки - экспоненциальная случайная величина со средним значением 15 кг. При воздействии ударной нагрузки под углом менее 45 градусов защитный кожух может выдержать нагрузку силой до 100 кг, при угле воздействия от 45 до 90 градусов - до 70 кг.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения вероятности того, что в течение рабочей смены защитный кожух не разрушится.

**Задача П5.5.** В состав некоторых механизмов, выпускаемых предприятием, входит плита с четырьмя круглыми отверстиями и четыре вала; валы должны вставляться в отверстия. Номинальные (заданные в технической документации) размеры деталей следующие: диаметр отверстия - 10 мм, диаметр вала - 9,5 мм. Допустимым считается зазор между отверстием и валом от 0,2 до 0,8 мм. Чтобы механизм был годным, требования к размеру зазора должны выполняться для всех четырех отверстий и валов.

Из-за погрешностей в работе оборудования размеры деталей могут отклоняться от заданных. Диаметр отверстия представляет собой гауссовскую

случайную величину со средним значением 10 мм и стандартным отклонением 0,2 мм, а диаметр вала - гауссовскую случайную величину со средним значением 9,5 мм и стандартным отклонением 0,1 мм.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения вероятности выпуска годного изделия.

**Задача П5.6.** Торговое предприятие ежедневно закупает у издательства 2000 экземпляров некоторой газеты по цене 7 ден.ед. и продает по 10 ден.ед. Из опыта работы предприятия известно, что в утреннее время спрос на газеты обычно составляет от 800 до 1400 экземпляров. В дневное время спрос обычно составляет от 400 до 800 экземпляров. Спрос в вечернее время зависит от того, напечатаны ли в газете какие-либо материалы, вызвавшие повышенный интерес. Если в газете опубликованы такие материалы, то спрос составляет от 300 до 500 экземпляров; если таких материалов нет, спрос составляет от 100 до 300 экземпляров. Материалы, вызывающие повышенный интерес, печатаются в данной газете в среднем 2 раза в неделю.

Газеты, не распроданные в течение дня, не могут быть проданы в последующие дни.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения следующих величин: а) средняя прибыль предприятия за день; б) вероятность того, что спрос на газеты не будет удовлетворен полностью.

**Задача П5.7.** Линия электропередачи (ЛЭП) соединяет электростанцию с крупным городом. Длина ЛЭП (расстояние между городом и электростанцией) – 300 км. Средства автоматического контроля, которыми оснащена ЛЭП, позволяют в случае аварии сразу же определить место, где она произошла, и оценить ее серьезность. Примерно в 80% случаев авария оказывается обычной, в 20% случаев - серьезной. При обычной аварии для ее устранения направляется ремонтная бригада из пункта (города или электростанции), ближайшего к месту аварии. Для устранения серьезной аварии направляется специальная ремонтная бригада с электростанции, независимо от того, где произошла авария.

Скорость продвижения ремонтной бригады к месту аварии - 100 км/ч. Время, необходимое для устранения аварии - случайная величина, которую можно считать распределенной по экспоненциальному закону. Устранение обычной аварии занимает в среднем 4 часа, серьезной - 7 часов.

Требуется разработать алгоритм и программу на основе метода Монте-Карло для определения среднего времени устранения аварии.

**Задача П5.8.** Разрабатывается план производства трех деталей (Д1, Д2, Д3) на рабочий день. Предприятие имеет 20 станков типа СТ1 и 30 станков типа СТ2. На каждом из станков можно выпускать детали всех трех типов. Производительность станков (деталей/день) приведена в табл.П5.1.

**Таблица П5.1**

Станок	Производительность, деталей/день		
	Д1	Д2	Д3

СТ1	20	35	15
СТ2	15	30	45

Данные в табл.П5.1 означают, например, что если станок типа СТ1 в течение всего рабочего дня используется для производства деталей Д1, то за день на этом станке будет выпущено 20 таких деталей.

Каждый станок настраивается на выпуск какой-либо одной детали в течение всего рабочего дня (таким образом, каждый станок в течение всего рабочего дня выпускает детали только одного вида).

За рабочий день предприятие должно выпускать не менее 150 деталей Д1, не менее 100 деталей Д2 и не менее 150 деталей Д3.

Прибыль предприятия от продажи одной детали Д1 составляет 6 ден.ед., от продажи детали Д2 - 4 ден.ед., от продажи детали Д3 - 8 ден.ед.

Требуется определить, сколько станков каждого типа должны выпускать в течение рабочего дня детали каждого вида, чтобы прибыль от продажи выпускаемых деталей была максимальной. Для решения задачи разработать алгоритм и программу на основе метода Монте-Карло.

**Задача П5.9.** Для выращивания зерновых культур (озимых и яровых) может использоваться 0,8 млн га земли в климатической зоне А и 0,6 млн га – в климатической зоне В. Урожайность земель приведена в табл.П5.2.

Таблица П5.2

Зерновые культуры	Урожайность, ц/га	
	Климатическая зона А	Климатическая зона В
Озимые	20	25
Яровые	24	15

Необходимо получить не менее 20 млн центнеров озимых и не менее 6 млн центнеров яровых.

Прибыль от продажи одного центнера озимых составляет 10 ден.ед., одного центнера яровых - 8 ден.ед.

Требуется определить, какую площадь земель необходимо использовать в каждой климатической зоне для выращивания каждой из культур, чтобы прибыль от продажи урожая была максимальной. Для решения задачи разработать алгоритм и программу на основе метода Монте-Карло.

**Задача П5.10.** Фирма, владеющая двумя предприятиями (П1 и П2), имеет возможность выполнить 4 заказа. Каждый заказ должен выполняться только одним предприятием (совместная работа нескольких предприятий над одним заказом невозможна). Каждое из предприятий может выполнять несколько заказов. Затраты предприятий, связанные с выполнением заказов, и размеры платы заказчиков за их выполнение приведены в табл.П5.3.

Таблица П5.3

Заказ	1	2	3	4
Затраты предприятия П1, млн ден.ед.	4	2	6	3
Затраты предприятия П2, млн ден.ед.	7	1	2	5

Плата заказчика, млн ден.ед.	12	4	8	6
------------------------------	----	---	---	---

Данные в табл.П5.3 означают, например, что заказчик платит фирме за выполнение первого заказа 12 млн ден.ед. Если этот заказ будет выполняться первым предприятием, то затраты на его выполнение составят 4 млн ден.ед.; таким образом, прибыль фирмы составит  $12-4=8$  млн ден.ед. Если этот заказ будет выполняться вторым предприятием, то затраты составят 7 млн ден.ед. (прибыль фирмы составит  $12-7=5$  млн ден.ед.).

Требуется распределить заказы между предприятиями таким образом, чтобы прибыль фирмы была максимальной. Для решения задачи разработать алгоритм и программу на основе метода Монте-Карло.

Учебное издание

**Смородинский Сергей Степанович,**  
**Батин Николай Владимирович**

**ОПТИМИЗАЦИЯ РЕШЕНИЙ НА ОСНОВЕ КОМПЬЮТЕРНЫХ  
ИМИТАЦИОННЫХ МЕТОДОВ И МОДЕЛЕЙ**

**УЧЕБНОЕ ПОСОБИЕ**

по курсу “Моделирование систем”  
для студентов специальности

“Автоматизированные системы обработки информации”  
дневной и дистанционной форм обучения

В 2-х частях

Часть 1

Редактор и корректор Т.Н. Крюкова

---

Подписано в печать 14.10.2004.

Бумага офсетная.

Уч.-изд. л. 4,5.

Печать ризографическая.

Тираж 150 экз.

Формат 60x84 1/16.

Усл.печ. л. 4,76.

Заказ 223.

---

Издатель и полиграфическое исполнение: Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»

Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.

Лицензия на осуществление полиграфической деятельности №02330/0133108 от 30.04.2004.

220013, Минск, П. Бровки, 6