В.И. Варфоломеев

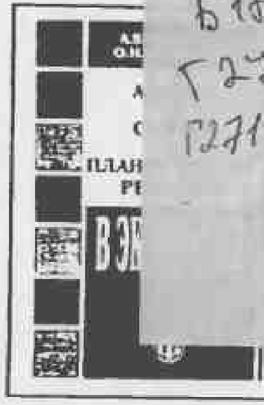
Алгоритмическое моделирование элементов экономических систем

ПРАКТИКУМ

"

А

Анали... ний



...сан на ...вне. Он ...овение ...одов во ... причем ...нтации. ...и, где ...е, что в ... сфере ...х реше- ...менные ...ческие ...стемы, ... плани- рования и синтеза экономических решений в условиях неопределенности. Рассмотрены методологические подходы использования этих методов и систем для решения широкого спектра прикладных ...омических задач.

...циальности ...смежным ...ся и

В.И.Варфоломеев

Алгоритмическое моделирование элементов экономических систем

ПРАКТИКУМ

Рекомендовано
Министерством образования
Российской Федерации
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальности
"Математические методы и исследование
операций в экономике"

МОСКВА
"ФИНАНСЫ И СТАТИСТИКА"
2000

УДК 519.86 – 37(076.5)
ББК 65в6я73
В18

РЕЦЕНЗЕНТЫ:
кафедра информационно-вычислительных систем
Московского университета потребительской кооперации
и доктор технических наук, профессор
С. В. Черемных

Варфоломеев В. И.

В18 Алгоритмическое моделирование элементов экономических систем: Практикум. Учеб. пособие. – М.: Финансы и статистика, 2000. – 208 с: ил.

ISBN 5-279-02123-7.

Рассматриваются основы алгоритмического моделирования и подробно описываются процессы разработки ряда алгоритмических моделей экономических систем. Приводятся сведения из теории вероятностей, математической статистики и теории эффективности, которые необходимо знать для понимания содержания описанных алгоритмических моделей. В приложениях помещены программы рассмотренных имитационных моделей экономических систем, составленные на алгоритмических языках Visual Basic 5.0 и Pascal.

Для студентов экономических вузов, обучающихся по специальности 071900 «Информационные системы в экономике». Может быть также полезно научным сотрудникам, занимающимся исследованиями в области экономики, менеджмента, коммерческой и консалтинговой деятельности.

В $\frac{2404000000 - 023}{010(01) - 2000}$ 162 – 99

УДК 519.86–37(076.5)
ББК 65в6я73

ISBN 5-279-02123-7

ПРЕДИСЛОВИЕ

Настоящее учебное пособие предназначено для студентов экономических вузов, обучающихся по специальности «Информационные системы в экономике». Задачей их обучения в вузе является подготовка квалифицированных специалистов, способных принимать обоснованные решения с целью повышения эффективности экономических систем различного типа.

В настоящее время широко распространено участие в процессе выработки экономических решений специальных консультационных, или консалтинговых фирм.

По заказам предпринимателей, коммерсантов, менеджеров и т. д. консалтинговые фирмы разрабатывают математические модели реальных процессов, которые должны протекать в планируемых для создания экономических системах.

При этом для разработки моделей сложных экономических систем используются современные пакеты прикладных программ типа GPSS, Pilgrim, ReThink и т. п., позволяющие создавать многоуровневые структурные (функциональные) модели исследуемых объектов в графических терминах.

Однако для изучения упомянутых пакетов прикладных программ студентами требуется дополнительное учебное время, которым обычно современный экономический вуз не располагает. В то же время для разработки и создания сравнительно простых моделей экономических систем вполне пригоден метод алгоритмического статистического моделирования. Для использования этого метода достаточно знания одного из универсальных языков программирования типа Pascal, Quick Basic или Visual Basic.

Книга состоит из двух частей и приложений.

В первой части рассматриваются основы теории алгоритмического моделирования сложных систем и приводятся подробные описания семи алгоритмических статистических моделей простых экономических систем. Постановка задач для части из них заимствована из книги Т. Нейлора «Машинные имитационные эксперименты с моделями экономических систем» [6]. Остальные модели целиком разработаны автором.

В конце каждого раздела приводятся основные и дополнительные задания для самостоятельной работы. При выполнении основного задания по каждому разделу читатель должен использовать приведенное в Приложении описание Visual Basic-программы для создания машинной модели на персональном компьютере. При выполнении дополнительных заданий отлаженная программа используется для проведения исследования закономерностей функционирования моделируемой системы.

Материал, помещенный в первой части книги, рассчитан на читателя, который знаком с основными положениями теории вероятностей, математической статистики и теории эффективности.

Для остальных читателей во второй части приведен справочный материал по этим дисциплинам, который может потребоваться для понимания содержания алгоритмических моделей.

В приложениях даны программы для всех описанных моделей на языках Visual Basic 5.0 и Pascal.

В разработке моделей экономических систем должны участвовать как минимум два специалиста. Первый из них – это предприниматель или менеджер, который заинтересован в создании математической модели некоторой еще не существующей экономической системы с целью получения максимально возможной прибыли. Он должен представлять, какие переменные являются независимыми (входными) и какие зависимыми (выходными), какие факторы влияют на процесс, протекающий в экономической системе, и какие из них являются в той или иной степени неопределенными (неизвестными). Предприниматель должен выбрать показатель, по которому он собирается оценивать эффективность будущей экономической системы, и критерий для выбора альтернативных вариантов ее структуры.

Второй разработчик – это специалист в области математического моделирования сложных систем (необязательно экономических). Он должен, используя информацию, полученную от предпринимателя, составить формализованное описание алгоритмической модели, разработать алгоритм и программу, произвести отладку и испытания модели и совместно с предпринимателем произвести машинный эксперимент с экономической моделью с целью установления оптимальных параметров системы, обеспечивающих максимум (или минимум) выбранного показателя эффективности.

Реально на практике такой диалог ведется обычно между предпринимателем и консультантом консалтинговой фирмы.

Студент, обучающийся по специальности «Информационные системы в экономике», должен приобрести навыки работы в качестве обоих упомянутых лиц. Это означает, что в процессе работы по созданию модели экономической системы он должен действовать то как предприниматель, то как специалист по моделированию. Иначе говоря, он должен постоянно вести диалог сам с собой, становясь то в позицию предпринимателя, то в позицию разработчика модели.

Поэтому описание каждой алгоритмической модели в книге начинается с диалога между предпринимателем и консультантом. В результате этого диалога появляется концептуальная модель экономической системы. Таким путем автор попытался облегчить читателям понимание того, как вырабатывается исходная информация для алгоритмической модели.

ВВЕДЕНИЕ

Лица, ответственные за принятие решений, касающихся проектирования и создания экономических систем, могут оценить их эффективность одним из трех следующих способов.

Во-первых, есть возможность (по крайней мере теоретическая) проводить управляемые эксперименты с экономической системой фирмы, отрасли или страны. Однако принятие неоптимальных решений может причинить ущерб экономической системе. При этом чем больше масштаб системы, тем ощутимее убытки. Тем не менее на практике такие эксперименты нередко производились и производятся (в некоторых странах) с неизменно отрицательным результатом.

Даже в случае оптимальных решений, касающихся, например, управления деятельностью фирмы, при проведении натурных экспериментов трудно сохранить постоянство факторов и условий, влияющих на результат, а следовательно, сложно обеспечить надежную оценку различных экономических решений.

Во-вторых, если есть данные о развитии экономической системы за некоторый период времени в прошлом, то можно провести мысленный эксперимент на этих данных. Однако для этого нужно знать точно, какие изменения каких входных переменных привели к наблюдаемому изменению выходных переменных, характеризующих эффективность экономической системы. Иногда причинами изменений могут оказаться случайные возмущения, или так называемый «шум». Поэтому нельзя слишком доверять оценкам экономических решений, полученным на основе данных о развитии системы в прошлом.

В-третьих, можно построить математическую модель рассматриваемой системы, связывающую входные (независимые) переменные с выходными (зависимыми) переменными, а также с экономической стратегией, т. е. со способом управления экономической системой. Если есть основания для того, чтобы считать разработанную математическую модель ***адекватной*** рассматриваемой экономической системе, то с помощью модели можно производить расчеты или машинные эксперименты (если модель реализована на ЭВМ). По результатам этих экспериментов можно выработать рекомендации по повышению эффективности существующей или проектируемой экономической системы.

Условием для разработки модели является наличие так называемой ***информационной достаточности.*** Это означает, что разработчик должен иметь достаточное представление о том, что является входными и выходными переменными в исследуемой системе и какие факторы оказывают влияние на процесс ее функционирования. Если уровень информационной достаточности невысок, то создать модель, с помощью которой можно получать новые знания об объекте-оригинале, невозможно. Если же уровень информационной достаточности велик, т. е. система уже хорошо изучена, то вопрос о создании модели теряет смысл, так как новых знаний она также не даст.

Следовательно, разрабатывать модель имеет смысл только в том случае, если объект-оригинал еще недостаточно изучен или вообще не существует в природе и только проектируется.

Если объект-оригинал существует, то модель считается ***адекватной*** ему в том случае, если зависимость выходных переменных от входных параметров в модели и в объекте-оригинале практически совпадает. При упрощении моделей степень адекватности снижается.

Если же объекта-оригинала еще не существует, то модель считается ***адекватной*** объекту-оригиналу, если она с достаточной степенью приближения на уровне понимания моделируемого процесса исследователем отражает закономерности процесса функционирования реальной системы. Залогом адекватности в этом случае является полнота описания моделируемого процесса, т. е. учет всех факторов, поддающихся формализации.

Ярким примером успешного решения задачи моделирования процесса, который невозможно осуществить на практике, является разработка вычислительным центром АН СССР в 1985 г. под руководством академика Н. Н. Моисеева модели ядерной войны, получившей название «Гея». С помощью этой модели были строго научно оценены катастрофичные для человека и всего живого на Земле последствия, к которым привела бы ядерная война. Опубликование результатов исследований внесло важный вклад в ослабление ядерной угрозы.

Существует множество различных типов моделей: физические, аналоговые, интуитивные и т. д. Особое место среди них занимают математические модели, которые, по мнению академика А.А. Самарского, «являются самым большим достижением научно-технической революции XX века». Математические модели делятся на две группы: ***аналитические*** и ***алгоритмические*** (которые иногда называют ***имитационными***).

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Не составляет исключения и экономическая деятельность. Однако заметных успехов в области моделирования экономических процессов до сих пор не наблюдается.

На наш взгляд, это обстоятельство объясняется следующими причинами.

1. Экономические процессы происходят в значительной мере стихийно, неуправляемо. Они плохо поддаются попыткам волевого управления со стороны политических, государственных и хозяйственных руководителей отдельных отраслей и экономики страны в целом. По этой причине экономические системы плохо поддаются изучению и формализованному описанию.

2. Специалисты в области экономики, как правило, имеют слабую математическую подготовку вообще и по вопросам математического моделирования в частности. Большинство из них не умеет формально описывать (формализовывать) наблюдаемые экономические процессы. Это, в свою очередь, не позволяет установить, адекватна ли та или иная математическая модель рассматриваемой экономической системе.

3. Специалисты в области математического моделирования, не имея в своем распоряжении формализованного описания экономического процесса, не могут создать адекватную ему математическую модель.

Существующие математические модели, которые принято называть ***моделями экономических систем,*** можно условно разделить на три группы.

К ***первой группе*** можно отнести модели, достаточно точно отражающие какую-либо одну сторону определенного экономического процесса, происходящего в системе сравнительно малого масштаба. С точки зрения математики они представляют собой весьма простые соотношения между двумя-тремя переменными. Обычно это алгебраические уравнения 2-й или 3-й степени, в крайнем случае система алгебраических уравнений, требующая для решения применения метода итераций (последовательных приближений). Они находят применение на практике, но не представляют интереса с точки зрения специалистов в области математического моделирования.

Ко ***второй группе*** можно отнести модели, которые описывают реальные процессы, протекающие в экономических системах малого и среднего масштаба, подверженные воздействию случайных и неопределенных факторов. Разработка таких моделей требует принятия допущений, позволяющих разрешить неопределенности. Например, требуется задать распределения случайных величин, относящихся к входным переменным. Эта искусственная операция в известной степени порождает сомнение в достоверности результатов моделирования. Однако другого способа создания математической модели не существует.

Среди моделей этой группы наибольшее распространение получили модели так называемых систем массового обслуживания. Существуют две разновидности этих моделей: аналитические и алгоритмические. Аналитические модели не учитывают действие случайных факторов и поэтому могут использоваться только как модели первого приближения. С помощью алгоритмических моделей исследуемый процесс может быть описан с любой степенью точности на уровне его понимания постановщиком задачи.

К *третьей группе* относятся модели больших и очень больших (макроэкономических) систем: крупных торговых и промышленных предприятий и объединений, отраслей народного хозяйства и экономики страны в целом. Создание математической модели экономической системы такого масштаба представляет собой сложную научную проблему, решение которой под силу лишь крупному научно-исследовательскому учреждению.

Экономические модели, рассматриваемые в данной книге, относятся ко второй группе. Учебное пособие предназначено для обучения студентов экономических вузов по курсу «Моделирование экономических процессов». Главной задачей этого курса является выработка у обучаемых умения формально описывать те или иные экономические процессы, а также разрабатывать алгоритмы и программы и с их помощью проводить машинные эксперименты с целью исследования закономерностей функционирования исследуемых систем.

В учебном пособии не ставилась задача охватить все классы экономических моделей. Приведенная классификация (с. 49) имеет условный характер и не претендует на полноту и законченность.

Рассмотренные в пособии экономические модели подобраны по признаку используемого аппарата алгоритмического моделирования. В основном это модели систем массового обслуживания различных модификаций. Поэтому каждая модель может иметь два наименования. Первое из них характеризует класс алгоритмической модели, а второе – один из возможных объектов-оригиналов, свойства которого могут изучаться с помощью данной модели. Как правило, таких объектов существует несколько. Именно здесь проявляется универсальность методов математического моделирования сложных систем. Одна и та же модель может использоваться для изучения свойств совершенно разных по физической природе реальных систем.

ЧАСТЬ 1. АЛГОРИТМИЧЕСКИЕ МОДЕЛИ ЭКОНОМИЧЕСКИХ СИСТЕМ

1. ОСНОВЫ АЛГОРИТМИЧЕСКОГО СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

1.1. ПОНЯТИЕ МОДЕЛИ

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Между тем общепризнанного определения понятия модели не существует.

На наш взгляд, заслуживает предпочтения следующее определение:

***модель** – объект любой природы, который создается исследователем с целью получения новых знаний об объекте-оригинале и отражает только существенные (с точки зрения разработчика) свойства оригинала.*

Анализируя содержание этого определения, можно сделать следующие выводы:

1) любая модель субъективна, она несет на себе печать индивидуальности исследователя;

2) любая модель *гомоморфна*, т. е. в ней отражаются не все, а только существенные свойства объекта-оригинала;

3) возможно существование множества моделей одного и того же объекта-оригинала, отличающихся целями исследования и степенью адекватности.

Модель считается *адекватной* объекту-оригиналу, если она с достаточной степенью приближения на уровне понимания моделируемого процесса исследователем отражает закономерности процесса функционирования реальной системы во внешней среде.

1.2. КЛАССИФИКАЦИЯ МОДЕЛЕЙ

По форме представления объектов модели можно разделить на две группы: ***материальные*** и ***идеальные.*** Материальные модели, в свою очередь, делятся на ***физические*** и ***аналоговые.*** В ***физических моделях*** обеспечивается аналогия физической природы и модели (примером может служить аэродинамическая труба). В ***аналоговых моделях*** добиваются сходства процессов, протекающих в оригинале и модели (так с помощью гидроинтегратора моделируется передача тепла).

Идеальные модели можно разделить на ***знаковые*** (семиотические) и ***интуитивные*** (мысленные). ***Знаковые модели*** делятся на логические, геометрические и математические.

Математические модели можно разделить на ***аналитические, алгоритмические*** и ***комбинированные.***

Для ***аналитического*** моделирования характерно то, что для описания процессов функционирования системы используются системы алгебраических, дифференциальных, интегральных или конечно-разностных уравнений. Аналитическая модель может быть исследована следующими методами:

а) аналитическим, когда стремятся получить в общем виде явные зависимости для искомых характеристик;

б) численным, когда, не умея решать уравнения в общем виде, стремятся получить числовые результаты при конкретных начальных данных;

в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

Желая использовать аналитический метод, часто идут на существенные упрощения первоначальной модели, чтобы иметь возможность изучить хотя бы общие свойства системы. Аналитические модели бывают ***детерминированные*** и ***статистические.*** Численный метод проведения аналитических расчетов с помощью датчиков случайных чисел получил название метода статистических испытаний, или метода Монте-Карло.

При ***алгоритмическом моделировании*** описывается процесс функционирования системы во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени. ***Алгоритмические*** модели также могут быть детерминированными и статистическими. В последнем случае в модели с помощью датчиков случайных чисел имитируется действие неопределенных и случайных факторов. Такой метод моделирования получил название метода статистического моделирования. В настоящее время этот метод считается наиболее эффективным методом исследования сложных систем, а часто и единственным практически доступным методом получения информации о поведении гипотетической системы на этапе ее проектирования.

Комбинированное моделирование позволяет объединить достоинства аналитического и алгоритмического моделирования. При построении комбинированных моделей производится предварительная декомпозиция процесса функционирования модели на составляющие подпроцессы. Для тех из них, где это возможно, используются аналитические модели, а для остальных процессов строятся алгоритмические модели.

1.3. ПОСЛЕДОВАТЕЛЬНОСТЬ РАЗРАБОТКИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

В процесс разработки и машинной реализации математической модели входят следующие этапы:

- построение концептуальной модели;
- разработка алгоритма модели системы;
- разработка программы модели системы;
- проведение машинных экспериментов с моделью системы.

1.3.1. Построение концептуальной модели

Построение концептуальной модели включает следующие подэтапы:

- постановку задачи моделирования;

• определение требований к исходной информации и ее сбор;
• выдвижение гипотез и предположений;
• определение параметров и переменных модели;
• обоснование выбора показателей и критериев эффективности системы;
• составление содержательного описания модели.

При постановке задачи моделирования дается четкая формулировка целей и задач исследования реальной системы, обосновывается необходимость машинного моделирования, выбирается методика решения задачи с учетом имеющихся ресурсов, определяется возможность разделения задачи на подзадачи.

При сборе необходимой исходной информации необходимо помнить, что именно от качества исходной информации об объекте моделирования зависит как адекватность модели, так и достоверность результатов моделирования.

Гипотезы при построении модели системы служат для заполнения «пробелов» в понимании задачи исследователем. Предположения дают возможность провести упрощение модели. В процессе работы с моделью системы возможно многократное возвращение к этому подэтапу в зависимости от полученных результатов моделирования и новой информации об объекте.

При определении параметров и переменных составляется перечень входных, выходных и управляющих переменных, а также внешних и внутренних параметров системы.

Выбранные показатели и критерии эффективности системы должны отражать цель функционирования системы и представлять собой функции переменных и параметров системы.

Разработка концептуальной модели завершается составлением содержательного описания, которое используется как основной документ, характеризующий результаты работы на первом этапе.

1.3.2. Разработка алгоритма модели

Разработка алгоритма модели включает следующие подэтапы:
• построение логической схемы алгоритма;
• получение математических соотношений;
• проверку достоверности алгоритма.

Вначале создается укрупненная (обобщенная) схема моделирующего алгоритма, которая задает общий порядок действий при моделировании исследуемого процесса. Затем разрабатывается детальная схема, каждый элемент которой впоследствии превращается в оператор программы.

Для комбинированных моделей разрабатывается аналитическая часть в виде явных функций и имитационная часть в виде моделирующего алгоритма.

Проверка достоверности алгоритма должна дать ответ на вопрос, насколько алгоритм отражает замысел моделирования, сформулированный на этапе разработки концептуальной модели.

1.3.3. Разработка программы

Разработка программы для ЭВМ включает следующие подэтапы:
• выбор вычислительных средств;
• проведение программирования;
• проверку достоверности программы.

Прежде всего выбираются тип ЭВМ (компьютера) и язык программирования. Создание программы по детально разработанному алгоритму может осуществить программист без участия и помощи разработчика модели.

После составления программы производится проверка ее достоверности на контрольном примере. На этом подэтапе необходимо оценить затраты машинного времени для расчета одной реализации моделируемого процесса, что позволит разработчику модели правильно сформулировать требования к точности и достоверности результатов моделирования.

1.3.4. Проведение машинных экспериментов с моделью системы

На этом этапе проводятся серийные расчеты по составленной и отлаженной программе. Этап включает следующие подэтапы:
• планирование машинного эксперимента;
• проведение рабочих расчетов;

• представление результатов моделирования;

• интерпретацию результатов моделирования;

• выдачу рекомендаций по оптимизации режима работы реальной системы.

Перед проведением рабочих расчетов на ЭВМ должен быть составлен план проведения эксперимента с указанием комбинаций переменных и параметров, для которых должно проводиться моделирование системы. Задача заключается в разработке оптимального плана эксперимента, реализация которого позволяет при сравнительно небольшом числе испытаний модели получить достоверные данные о закономерностях функционирования системы.

Результаты моделирования могут быть представлены в виде таблиц, графиков, диаграмм, схем и т. п. В большинстве случаев наиболее простой формой считаются таблицы, хотя графики более наглядно иллюстрируют результаты моделирования системы. Целесообразно предусмотреть вывод результатов на экран дисплея и на принтер.

Интерпретация результатов моделирования имеет целью переход от информации, полученной в результате машинного эксперимента с моделью, к выводам, касающимся процесса функционирования объекта-оригинала.

На основании анализа результатов моделирования принимается решение о том, при каких условиях система будет функционировать с наибольшей эффективностью.

1.4. ТИПОВЫЕ МАТЕМАТИЧЕСКИЕ СХЕМЫ

В процессе создания математической модели, реализуемой на ЭВМ, происходит переход от содержательного описания к формальному алгоритму. Промежуточным звеном между ними может служить ***математическая схема.***

Существует ряд типовых математических схем, которые могут лечь в основу разрабатываемого конкретного моделирующего алгоритма.

К ним относятся следующие схемы (модели):

• непрерывно-детерминированные модели (D-схемы);

• дискретно-детерминированные модели (F-схемы);

• дискретно-стохастические модели (P-схемы);

• непрерывно-стохастические модели (Q-схемы).

К ***непрерывно-детерминированным моделям*** относятся модели, описываемые системами обыкновенных дифференциальных уравнений или уравнений в частных производных. В качестве независимой переменной, от которой зависят неизвестные искомые функции, обычно служит время. Тогда вектор-функция искомых переменных будет непрерывной. Математические схемы такого вида отражают динамику изучаемой системы и поэтому называются D-схемами (англ. dynamic).

К ***дискретно-детерминированным моделям*** относятся так называемые конечные автоматы. Автомат можно представить как некоторое устройство, на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния. У конечного автомата множество входных сигналов и внутренних состояний является конечным множеством. Название F-схема происходит от английских слов finite automata.

К ***дискретно-стохастическим моделям*** относятся вероятностные (стохастические) автоматы или по-английски probabilistic automat. Отсюда название – P-схема. В общем виде вероятностный автомат можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано стохастически.

Примером типовой схемы ***непрерывно-стохастического типа*** может служить схема ***системы массового обслуживания*** (СМО) или по-английски queueing system. Отсюда название – Q-схема.

В качестве процесса обслуживания в СМО могут быть представлены различные по физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки товаров, потоки продукции, потоки деталей, потоки клиентов и т. п.

Для любой системы массового обслуживания характерно наличие трех отличительных свойств:

- объектов, у которых может возникнуть потребность в удовлетворении некоторых заявок;
- агрегатов, предназначенных для удовлетворения заявок на обслуживание;
- специальной организации приема в систему заявок и их обслуживания.

Схема системы массового обслуживания показана на рис. 1.1.

Рис. 1.1. Схема системы массового обслуживания

Совокупность заявок рассматривают как ***поток событий,*** т. е. последовательность событий, происходящих в случайные моменты времени. Время обслуживания заявки также считается случайной величиной.

Из-за совместного действия этих двух случайных факторов количество обслуженных заявок в заданном интервале времени является величиной случайной.

Исследование моделей СМО ставит целью установление параметров случайных величин, характеризующих процесс обслуживания заявок.

Существует несколько разновидностей СМО:

1) по числу каналов обслуживания СМО делятся на одноканальные и многоканальные;

2) по числу фаз (последовательно соединенных агрегатов) СМО делятся на однофазные и многофазные;

3) по наличию обратной связи СМО делятся на разомкнутые (с бесконечным числом заявок) и замкнутые (с конечным числом заявок);

4) по наличию очереди СМО делятся на системы без очередей (с потерями заявок), системы с неограниченным ожиданием (по времени или длине очереди) и системы с ограниченным ожиданием (по времени или длине очереди);

5) по принципу формирования очередей СМО делятся на системы с общей очередью и системы с несколькими очередями;

6) по наличию отказов СМО делятся на системы с отказами и системы без отказов;

7) по виду приоритета СМО делятся на системы со статическим приоритетом (обслуживание в порядке поступления заявок) и системы с динамическим приоритетом, который, в свою очередь, имеет три разновидности:

- ***относительный приоритет*** (заявка высокого приоритета ожидает окончания обслуживания заявки с более низким приоритетом);
- ***абсолютный приоритет*** (заявка высокого приоритета при поступлении немедленно вытесняет заявку с более низким приоритетом);
- ***смешанный приоритет*** (если заявка с низшим приоритетом обслуживалась в течение времени, меньше критического, то используется абсолютный приоритет, в противном случае используется относительный приоритет).

1.5. ДАТЧИКИ СЛУЧАЙНЫХ ЧИСЕЛ С РАВНОМЕРНЫМ РАСПРЕДЕЛЕНИЕМ

Алгоритмическое моделирование – это численный метод исследования систем и процессов с помощью моделирующего алгоритма.

Каждый раз, когда на ход моделируемого процесса оказывает влияние случайный фактор, его действие имитируется с помощью специально организованного розыгрыша (жребия). Таким образом строится одна случайная реализация моделируемого явления, представляющая собой как бы один результат опыта. По одному опыту, конечно, нельзя судить о закономерностях изучаемого процесса. Но при большом числе реализаций средние характеристики, вырабатываемые моделью, приобретают свойство устойчивости, которое усиливается с увеличением числа реализаций.

Бросание жребия можно осуществить вручную (выбором из таблицы случайных чисел), но удобнее это делать с помощью специальных программ, входящих в состав программного обеспечения ЭВМ. Такие программы называют ***датчиками*** или ***генераторами*** случайных чисел.

В трансляторах почти всех алгоритмических языков имеются стандартные процедуры или функции, которые генерируют случайные (точнее, псевдослучайные) величины с равномерным распределением.

В трансляторе языка Visual Basic имеется стандартная функция *RND*, возвращающая случайные вещественные числа одинарной точности в интервале (0,1).

Обращение к этой функции может иметь вид: $z = RND$, где z – возможное значение случайной величины, равномерно распределенной в интервале (0,1).

1.6. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ СОБЫТИЙ

1.6.1. Моделирование простого события

Пусть имеется событие A, вероятность наступления которого равна P_A. Требуется выработать правило, при многократном использовании которого частота появления события стремилась бы к его вероятности. Выберем с помощью датчика случайных чисел, равномерно распределенных в интервале (0,1) некоторое число z и определим вероятность того, что $z < P_A$. Для случайной величины z с равномерным распределением справедлива следующая зависимость:

$$P(z < P_A) = \int_0^{PA} f(x)dx = P_A.$$

Таким образом, вероятность попадания случайной величины в интервал $(0,P_A)$ равна величине P_A. Поэтому если при розыгрыше число z попало в этот интервал, то следует считать, что событие A произошло. Противоположное событие (не A) произойдет с вероятностью $(1-P_A)$ в том случае, если $z \geq P_A$.

Процедура моделирования простого события в имитационной модели описывается алгоритмом, схема которого показана на рис. 1.2.

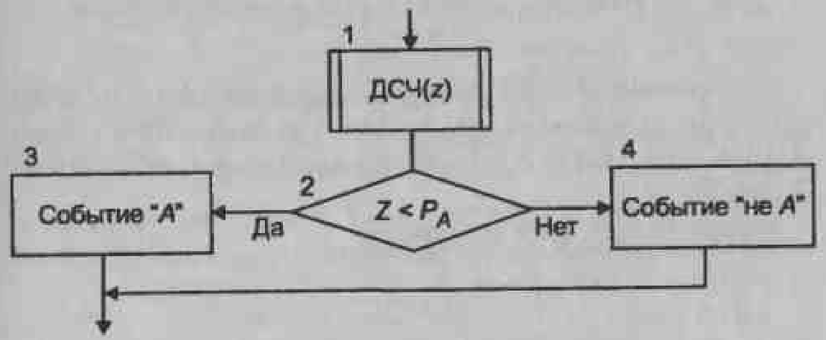


Рис. 1.2. Моделирование простого события

Оператор 1 обращается к датчику случайных чисел, генерирующему случайную величину z. Оператор 2 проверяет условие $z < P_A$. Если оно выполняется, считается, что произошло событие A. В противном случае считается, что произошло противоположное событие (не A).

1.6.2. Моделирование полной группы несовместных событий

Пусть имеется полная группа несовместных событий (ПГНС) $A_1, A_2, ..., A_k$ с вероятностями $P_1, P_2, ..., P_k$. При этом выполняется условие

$$\sum_{i=1}^{k} P_i = 1.$$

Разделим интервал (0,1) на k отрезков, длины которых составляют P_1; P_2; ..., P_k (рис. 1.3).

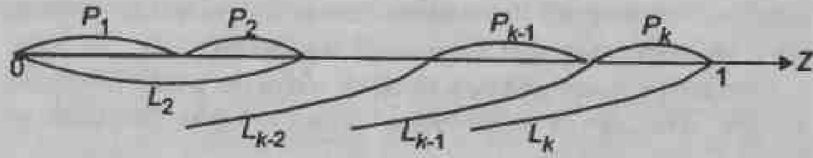


Рис. 1.3. Моделирование полной группы несовместных событий

Если случайное число z, генерированное датчиком случайных чисел с равномерным распределением в интервале (0,1), попало, например, на участок P_{k-1}, то это должно означать, что произошло событие A_{k-1}.

Действительно, если обозначить

$$L_j = \sum_{i=1}^{j} P_i,$$

то окажется справедливым выражение

$$P(L_{k-2} < z < L_{k-1}) = \int_{L_{k-2}}^{L_{k-1}} 1 \cdot dz = P_{k-1}.$$

Следовательно, произойдет событие, которое имеет вероятность P_{k-1}.

Процедура моделирования полной группы несовместных событий описывается алгоритмом, схема которого показана на рис. 1.4.

Оператор 1 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1). Условный оператор 2 проверяет условие попадания случайной величины z в интервал (0, L_1). Если это условие выполняется, то считается, что произошло событие A_1. Если условие в операторе 2 не выполняется, то алгоритм осуществляет проверку условий попадания случайной величины в другие интервалы. Одно из событий A_1, A_2, ... , A_k обязательно произойдет.

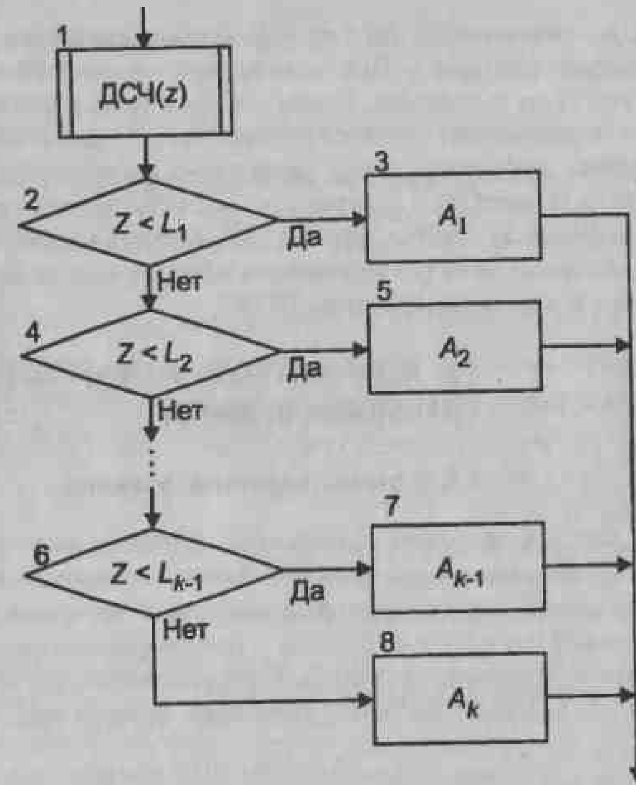


Рис. 1.4. Схема алгоритма моделирования ПГНС

1.7. МОДЕЛИРОВАНИЕ ДИСКРЕТНОЙ СЛУЧАЙНОЙ ВЕЛИЧИНЫ

Дискретная случайная величина может быть задана табличной зависимостью:

X	x_1	x_2	...	x_n
P	p_1	p_2	...	p_n

Здесь p_j – вероятность того, что дискретная случайная величина X примет значение x_j. При этом $p_1+p_2+...+p_n = 1$. Разделим интервал (0,1) на n отрезков, длины которых пропорциональны заданным вероятностям. Если случайное число z, вырабатываемое датчиком случайных чисел, равномерно распределенных в интервале (0,1), попадет в интервал p_k, то случайная величина X примет значение x_k. Таким образом, при моделировании дискретных случайных величин фактически используется та же процедура, что и при моделировании ПГНС.

1.8. МОДЕЛИРОВАНИЕ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

1.8.1. Метод обратной функции

Пусть имеется некоторая непрерывная случайная величина x, заданная функцией распределения $F(x)$. Можно доказать, что значения этой функции равномерно распределены в интервале (0,1). Поэтому между случайной величиной z, равномерно распределенной в том же интервале, и функцией распределения случайной величины x существует взаимно однозначное соответствие, т. е.

$$z = F(x). \tag{1.1}$$

Отсюда следует, что

$$x = F^{-1}(z), \tag{1.2}$$

где F^{-1} – обратная функция.

Следовательно, если уравнение (1.1) имеет аналитическое решение, то для моделирования случайной величины x можно использовать датчик случайных чисел, генерирующий величину z, и затем осуществить расчет по формуле (1.2).

1.8.2. Моделирование случайных величин с показательным распределением

Пусть имеется случайная величина x с показательным распределением. Функция распределения имеет вид:

$$F(x) = 1 - e^{-\lambda x},$$

где λ – параметр распределения.

Применив метод обратной функции, получим:

$$z = F(x) = 1 - e^{-\lambda x},$$

откуда

$$x = -\frac{1}{\lambda} Ln(1-z). \tag{1.3}$$

Учитывая, что случайная величина $(1-z)$ имеет также равномерное распределение в интервале (0,1), соотношение (1.3) можно заменить соотношением

$$x = -\frac{1}{\lambda} Ln(z).$$

1.8.3. Моделирование случайных величин с равномерным распределением

Датчик случайных чисел генерирует случайные величины с равномерным распределением в интервале (0,1). Если же нужно моделировать случайные величины с равномерным распределением в интервале (a,b), то можно воспользоваться методом обратной функции.

Для рассматриваемого случая выражение (1.1) примет вид:

$$z = F(x) = \frac{x-a}{b-a},$$

откуда $\quad x = a + z(b - a).$

На практике применяется и другой способ задания равномерного распределения. Вместо границ интервала задаются среднее значение случайной величины $x_{ср}$ и величина интервала Δx. Тогда определение возможного значения случайной величины с равномерным распределением может быть произведено по формуле

$$x = x_{ср} + \Delta x(z - 0{,}5).$$

1.8.4. Моделирование случайных величин с нормальным распределением

Метод обратной функции для нормального распределения неприменим, так как после подстановки соответствующей функции распределения выражение (1.2) не имеет аналитического решения. Поэтому в данном случае применяется другой метод.

Согласно центральной предельной теореме теории вероятностей при сложении достаточно большого числа одинаково распределенных независимых случайных чисел получается случайная величина, имеющая нормальное распределение.

Как показали исследования, уже при сложении более десяти случайных величин с равномерным распределением в интервале (0,1) получается случайная величина, которая с точностью, достаточной для большинства практических задач, может считаться распределенной нормально.

Процедура розыгрыша нормально распределенной случайной величины заключается в следующем.

1. Сложим 12 случайных величин с равномерным распределением в интервале (0,1), т. е. составим сумму

$$v = \sum_{i=1}^{12} z.$$

Использовав известные теоремы о сумме математических ожиданий и дисперсий независимых случайных величин, можно установить, что в данном случае случайная величина v имеет следующие характеристики:

математическое ожидание: $\mathrm{M(V)} = \sum\limits_{i=1}^{12} \mathrm{M(z)} = 12 \cdot \frac{1}{2} = 6;$

дисперсия: $\mathrm{D(V)} = \sum\limits_{i=1}^{12} \mathrm{D(z)} = 12 \cdot \frac{1}{12} = 1;$

среднее квадратическое отклонение: $\sigma(V) = +\sqrt{D(V)} = 1.$

2. Нормируем и центрируем случайную величину v, т. е. перейдем к величине

$$\eta = [v - M(V)] \,/\, \sigma(V) = v - 6.$$

3. От нормированной и центрированной величины η перейдем к случайной величине y с заданными параметрами $M(Y)$ и $\sigma(Y)$ по формуле

$$y = M(Y) + \sigma(Y) \cdot \eta,$$

где $M(Y)$ – известное математическое ожидание случайной величины y;
$\sigma(Y)$ – известное квадратическое отклонение случайной величины y.

1.8.5. Моделирование случайных величин с усеченным нормальным распределением

Усеченное нормальное распределение случайной величины x задается четырьмя параметрами: математическим ожиданием $M(X)$, средним квадратическим отклонением $\sigma(X)$, а также минимальным и максимальным значениями x_1 и x_2 (точками усечения).

Функция распределения случайной величины x определяется равенством

$$F(x) = \begin{cases} 0 & \text{при } x < x_1; \\ [\Phi_0(t) - \Phi_0(t_1)] \cdot A & \text{при } x_1 < x \le x_2; \\ 1 & \text{при } x > x_2, \end{cases}$$

где $A = \dfrac{1}{\Phi_0(t_2) - \Phi_0(t_1)};$

$$t = \frac{x - M(X)}{\sigma(X)}; \; t_1 = \frac{x_1 - M(X)}{\sigma(X)}; \; t_2 = \frac{x_2 - M(X)}{\sigma(X)}.$$

Существуют также формулы для расчета математического ожидания, дисперсии и среднего квадратического отклонения случайной величины x. Однако с достаточной для практики точностью при моделировании случайной величины с усеченным нормальным распределением можно обойтись без расчетов по формулам.

Для определения возможных значений случайной величины с этим распределением можно использовать алгоритм, схема которого приведена на рис. 1.5.

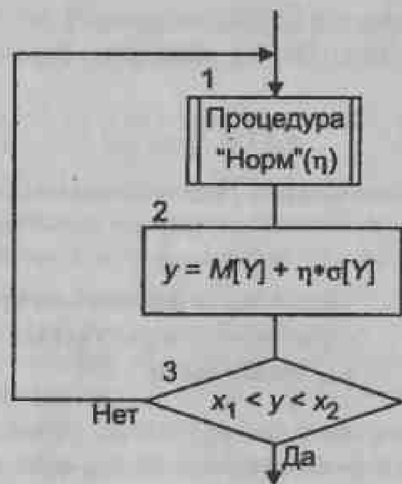


Рис. 1.5. Схема алгоритма моделирования случайной величины с усеченным нормальным распределением

Оператор 1 обращается к процедуре моделирования возможных значений нормированной и центрированной случайной величины η с нормальным распределением. Оператор 2 вычисляет значение случайной величины *y* с заданными параметрами *M*(*Y*) и σ(*Y*).

Условный оператор 3 проверяет условие попадания случайной величины *y* в неусеченную область. При выполнении этого условия значение случайной величины *y* с усеченным нормальным распределением считается найденным. В противном случае управление в алгоритме передается вновь на вход оператора 1 и генерируется другая случайная величина.

1.8.6. Моделирование случайных величин с произвольным распределением

Пусть случайная величина *x* задана в интервале (a_0, a_n) кусочно-постоянной функцией *f(x)*. Это значит, что интервал разбит на *n* частичных интервалов и плотность распределения *f(x)* на каждом из них постоянна (рис. 1.6).

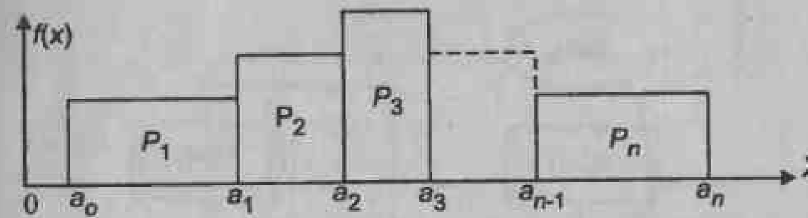


Рис. 1.6. Плотность распределения произвольной функции

Целесообразно выбрать величины a_K так, чтобы вероятности попадания в любой частичный интервал P_k были одинаковы, т. е.

$$\int_{a_{k-1}}^{a_k} f(x)dx = \frac{1}{n} \quad (k = 1, 2,\dots,n).$$

Из условия постоянства функции на каждом частичном интервале следует, что случайная величина *x* может быть определена по формуле

$$x = a_{k-1} + z(a_k - a_{k-1}) \quad (k = 1, 2,\dots,n). \qquad (1.4)$$

где *z* – возможное значение (реализация) случайной величины, равномерно распределенной в интервале (0,1);
a_{k-1} – левая граница частичного интервала;
a_k – правая граница частичного интервала.

Попадание в любой частичный интервал можно рассматривать как событие, входящее в полную группу несовместных событий. Поэтому процедура моделирования в общем случае состоит в следующем.

1. С помощью датчика случайных чисел с равномерным распределением, вырабатывающего величину *z*, моделируют дискретную случайную величину – номер интервала *k*.

2. Вторично разыгрывают случайную величину z и определяют возможное значение случайной величины *x* по формуле (1.4).

Схема алгоритма показана на рис. 1.7.

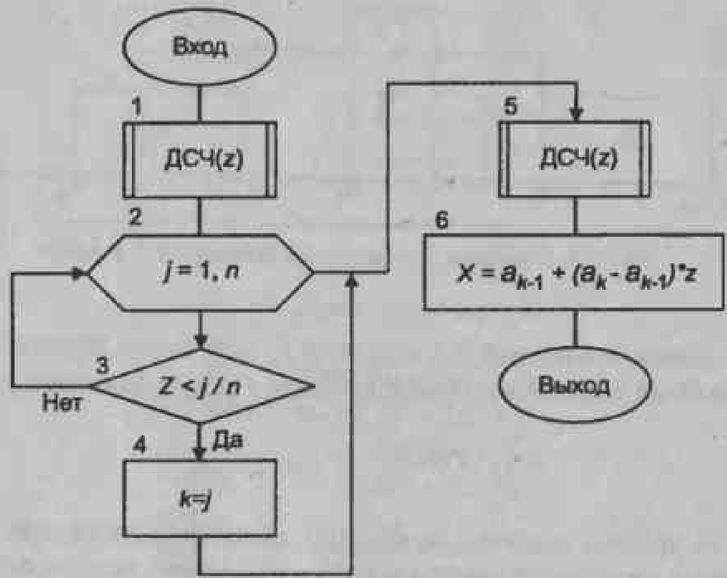


Рис. 1.7. Схема алгоритма моделирования случайной величины с произвольным распределением

1.9. СПОСОБЫ ПОСТРОЕНИЯ МОДЕЛИРУЮЩИХ АЛГОРИТМОВ

Существуют следующие способы (или принципы) построения моделирующих алгоритмов:

способ повременного моделирования с постоянным шагом;
способ повременного моделирования с переменным шагом;
способ последовательной проводки заявок;
способ поэтапной последовательной проводки заявок.

1.9.1. Повременное моделирование с постоянным шагом

Процесс функционирования системы можно рассматривать как последовательную смену ее состояний. Пусть, например, в одноканальной системе массового обслуживания происходит процесс обслуживания поступающих заявок (рис. 1.8).

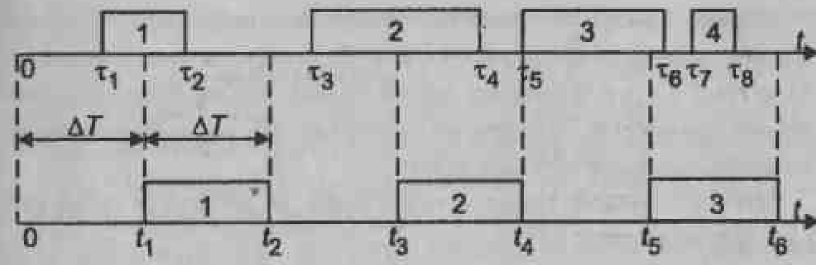


Рис. 1.8. Моделирование способом ΔT

Введем следующие обозначения:

τ_1 – момент начала обслуживания 1-й заявки;
τ_2 – момент конца обслуживания 1-й заявки;
τ_3 – момент начала обслуживания 2-й заявки;
τ_4 – момент конца обслуживания 2-й заявки;
τ_5 – момент начала обслуживания 3-й заявки;
τ_6 – момент конца обслуживания 3-й заявки;
τ_7 – момент начала обслуживания 4-й заявки;
τ_8 – момент конца обслуживания 4-й заявки.

Выберем шаг ΔT и будем анализировать состояние системы через промежутки времени t_1, t_2, t_3, ..., отстоящие друг от друга на ΔT. Этот способ иногда называют способом ΔT.

В момент t_1 будет обнаружено, что в системе началось обслуживание 1-й заявки. В момент $t_2 = t_1 + \Delta T$ будет установлено, что обслуживание 1-й заявки завершено. В момент $t_3 = t_2 + \Delta T$ будет обнаружено, что в системе началось обслуживание 2-й заявки. В момент $t_4 = t_3 + \Delta T$ будет установлено, что обслуживание 2-й заявки завершено. В момент $t_5 = t_4 + \Delta T$ будет обнаружено, что в системе началось обслуживание 3-й заявки. В момент $t_6 = t_5 + \Delta T$ будет установлено, что обслуживание 3-й заявки завершено. Факт поступления 4-й заявки и факт окончания ее обслуживания не будут обнаружены.

Для предотвращения потерь информации и повышения точности работы модели нужно уменьшить шаг ΔT. При малом ΔT можно достаточно точно описать процесс функционирования системы.

Однако способ ΔT является весьма неэкономичным с точки зрения расхода машинного времени. Достоинство способа состоит в том, что он позволяет моделировать любые процессы: детерминированные, непрерывные, случайные, с зависимыми или независимыми событиями и т. п.

Данный способ редко используется на практике из-за его неэкономичности.

1.9.2. Повременное моделирование с переменным шагом

В процессе функционирования системы массового обслуживания можно выделить два типа состояний:

обычные, в которых система находится почти все время;

особые, характеризующиеся сменой состояний.

К особым состояниям относятся: поступление заявки, начало и окончание ее обслуживания, появление отказа (сбоя) в работе системы, восстановление работоспособности системы и т. п.

Если моделирующий алгоритм построен по принципу особых состояний, то он просматривает процесс функционирования только в те моменты времени, когда состояние системы меняется.

В общем случае события, вызывающие особые состояния, могут быть зависимыми.

Моменты возникновения особых состояний и их характеристики (признаки) записываются в ***календарь событий***. В ходе моделирования может происходить корректировка календаря. Например, при поступлении заявки высшего приоритета может произойти прекращение обслуживания заявки низшего приоритета. Затем после освобождения канала возобновляется прерванный процесс обслуживания этой заявки.

Анализ календаря событий позволяет установить результаты моделирования для каждой случайной реализации. Данный метод применяется редко из-за сложности разработки алгоритма, основанного на использовании календаря событий.

1.9.3. Последовательная проводка заявок

При моделировании процессов обслуживания заявок в системах массового обслуживания иногда удобно строить моделирующий алгоритм по способу ***последовательной проводки заявок***. Этот способ может быть осуществлен, если события, происходящие в системе, не зависят друг от друга.

Моделирующий алгоритм последовательно воспроизводит истории отдельных заявок в порядке их поступления в систему. Алгоритм обращается к сведениям о других заявках лишь в том случае, если это необходимо для решения вопроса о дальнейшем порядке обслуживания данной заявки.

Этот способ не требует создания календаря событий.

Существуют две разновидности способа последовательной проводки:

проводка одиночных заявок;

проводка потоков заявок.

Рассмотрим вначале первый вариант. Пусть, например, в одноканальную систему массового обслуживания в одной случайной реализации процесса поступили четыре однородные заявки (с одинаковым приоритетом), как показано на рис. 1.9.

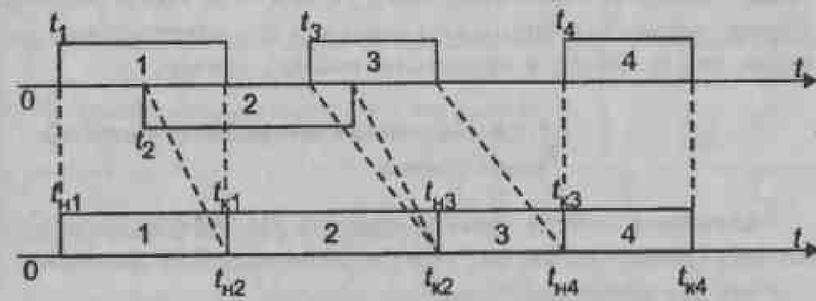


Рис. 1.9. Последовательная проводка заявок

В случайный момент времени t_1 в систему поступила 1-я заявка. Поскольку канал свободен, ее обслуживание начинается в момент времени $t_{н1} = t_1$. Если распределение времени обслуживания известно, то с помощью жребия можно определить случайную величину времени обслуживания τ_1. Тогда момент окончания обслуживания 1-й заявки будет равен: $t_{к1} = t_{н1} + \tau_1$. При этом к счетчику числа обслуженных заявок прибавляется единица.

Далее определяется время поступления 2-й заявки. Если распределение случайной величины времени между соседними заявками известно, то с помощью жребия можно определить интервал между 1-й и 2-й заявками и найти момент времени поступления 2-й заявки t_2. Поскольку канал занят, для 2-й заявки начинается период ожидания продолжительностью $\Delta T_{ож} = t_{к1} - t_2$. Обслуживание этой заявки начинается в момент $t_{н2} = t_{к1}$. Далее с помощью жребия определяется случайная величина времени обслуживания 2-й заявки τ_2. Тогда момент окончания обслуживания 2-й заявки будет равен: $t_{к2} = t_{н2} + \tau_2$. К счетчику числа обслуженных заявок прибавляется единица.

Аналогичным образом обслуживаются 3-я и 4-я заявки.

Вторым вариантом способа последовательной проводки заявок является способ, при котором вначале формируется поток заявок, а затем начинается процесс обслуживания заявок. При этом предполагается, что все заявки однородные. В этом случае для хранения данных о заявках в памяти ЭВМ необходимо создавать массивы чисел. Поскольку размерность такого массива заранее неизвестна, приходится выбирать ее с определенным запасом, что приводит к напрасному расходу памяти.

1.9.4. Поэтапная последовательная проводка заявок

Рассмотрим систему, предназначенную для обслуживания заявок двух различных приоритетов. Сделаем следующие допущения:

- все заявки независимы;
- при обслуживании используется абсолютный приоритет, т. е. поступающая заявка высшего приоритета немедленно вытесняет обслуживаемую заявку низшего приоритета;
- после освобождения канала может производиться «дообслуживание» той заявки второго приоритета, которая была вытеснена заявкой первого приоритета.

При этих допущениях для построения моделирующего алгоритма может быть применен усовершенствованный способ последовательной проводки, который можно назвать ***способом поэтапной последовательной проводки.***

Рассмотрим пример использования этого способа. Пусть в канал системы обслуживания поступают два потока заявок: первого (высшего) приоритета и второго (низшего) приоритета.

Первый этап моделирования

Распределение случайных величин интервалов между соседними заявками первого приоритета будем считать известным (например, показательным). Если задать интенсивность потока, то с помощью датчика случайных чисел можно определить случайные моменты поступления заявок $T_{1.1}$, $T_{1.2}$, $T_{1.3}$ и т. д. Распределение случайного времени обслуживания будем также считать известным (например, показательным). Если задать среднее время обслуживания, то с помощью датчика случайных чисел можно заранее определить случайные интервалы обслуживания (рис. 1.10).

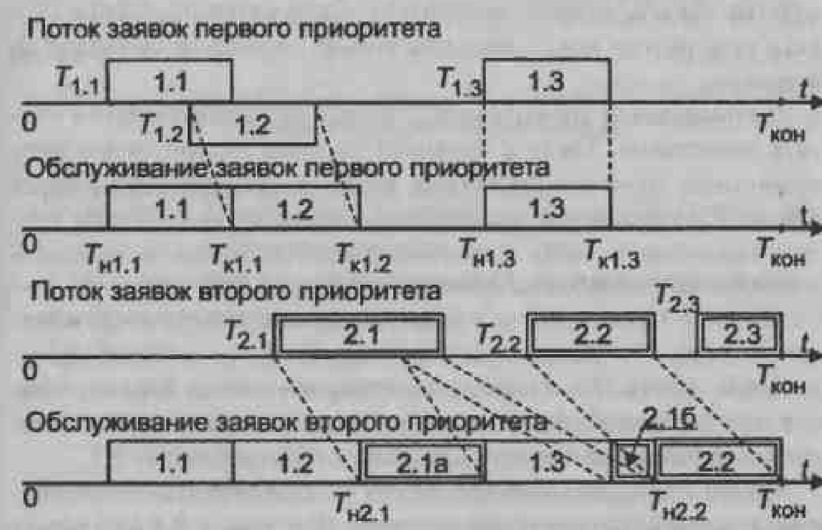


Рис. 1.10. Поэтапная последовательная проводка заявок

Установим модельное время на нуль. Будем рассматривать изолированный поток заявок первого приоритета так, как будто заявок второго приоритета не существует.

Применив способ последовательной проводки, можно установить моменты начала и окончания обслуживания заявок первого приоритета. Иначе говоря, в результате выполнения первого этапа моделирования можно определить значения элементов массива { $T_{н1.1}$, $T_{н1.2}$, $T_{н1.3}$, … } и элементов массива { $T_{к1.1}$, $T_{к1.2}$, $T_{к1.3}$, … }.

Можно также подсчитать число обслуженных заявок до конца периода обслуживания $T_{кон}$.

В данном случае заявки 1.1 и 1.3 обслуживаются сразу после их поступления, а заявка 1.2 обслуживается после некоторого ожидания, связанного с занятостью канала.

Второй этап моделирования

Вновь установим модельное время на нуль. Начинается этап моделирования процесса обслуживания заявок второго приоритета в условиях, что на временной оси располагаются уже обслуженные заявки первого приоритета. Следовательно, заявки второго приоритета могут занимать только свободные промежутки времени.

Распределение времени между соседними заявками будем считать известным. Тогда с помощью жребия можно определить возможные значения случайных величин времени поступления заявок. Распределение времени обслуживания также будем считать известным. Тогда с помощью жребия можно определить планируемые интервалы времени обслуживания.

Заявка 2.1 поступает в момент, когда канал занят обслуживанием заявки 1.2. Затем канал освобождается и начинается обслуживание заявки 2.1. Однако, поступившая заявка первого приоритета 1.3 вытесняет заявку 2.1. Только после освобождения канала происходит процесс «дообслуживания» заявки 2.1.

Заявка 2.2 также некоторое время ожидает начала обслуживания, а затем обслуживается до конца. Для заявки 2.3 не хватает времени, так как наступает конец периода обслуживания.

Рассмотрим теперь, как формализуется процесс обслуживания заявок второго приоритета в присутствии заявок первого приоритета. При взаимодействии заявок двух различных приоритетов могут возникнуть три возможные ситуации.

Ситуация 1. Ни одна из имеющихся N_{Z1}-заявок первого приоритета не препятствует обслуживанию заявки второго приоритета. Два возможных варианта этой ситуации иллюстрируются схемой, показанной на рис. 1.11.

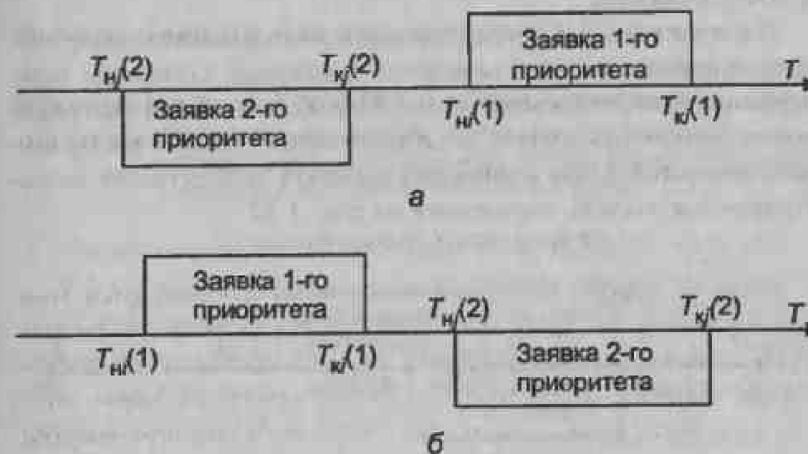


Рис. 1.11. Схемы вариантов 1-й ситуации:
а – первый вариант; *б* – второй вариант

Здесь $T_{нi}(1)$ – фактическое время начала обслуживания i-й заявки первого приоритета;

$T_{кi}(1)$ – фактическое время окончания обслуживания i-й заявки первого приоритета;

$T_{нj}(2)$ – первоначально намеченное время начала обслуживания j-й заявки второго приоритета (без учета возможности поступления заявки первого приоритета);

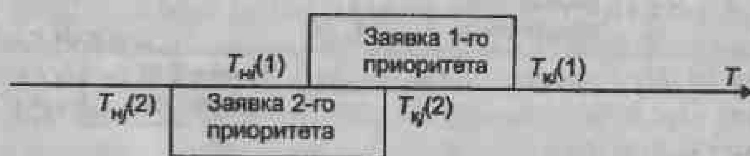
$T_{кj}(2)$ – первоначально намеченное время окончания процесса обслуживания j-й заявки второго приоритета (без учета возможности поступления заявки первого приоритета).

Логическое условие, при котором создается 1-й или 2-й вариант 1-й ситуации, записывается так:

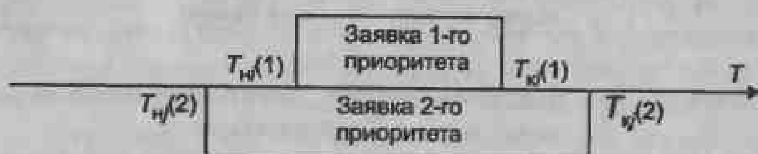
$$\{\ T_{кj}(2) <= T_{нi}(1)\ \}\ \mathrm{OR}\ \{\ T_{кi}(1) <= T_{нj}(2)\ \}. \qquad (1.5)$$

Если для j-й заявки второго приоритета условие (1.5) выполняется по отношению ко всем заявкам первого приоритета (i=1–N_{Z1}), то j-я заявка может быть обслужена. Ей может помешать только другая заявка 2-го приоритета, принятая ранее к обслуживанию.

Ситуация 2. Система приняла к обслуживанию заявку второго приоритета, и она начала обслуживаться. Однако до истечения расчетного времени окончания обслуживания поступила заявка первого приоритета, которая вытесняет данную заявку второго приоритета. Два возможных варианта этой ситуации иллюстрируются схемой, показанной на рис. 1.12.



а



б

Рис. 1.12. Схемы вариантов 2-й ситуации:
а – первый вариант; *б* – второй вариант

Логическое условие, при котором создается любой из вариантов 2-й ситуации, записывается так:

$$\{\ T_{нj}(2) < T_{нi}(1)\ \}\ \mathrm{AND}\ \{\ T_{нi}(1) < T_{кj}(2)\ \}. \qquad (1.6)$$

Если условие (1.6) выполняется хотя бы для какой-либо пары значений переменных $T_{нj}(2)$ и $T_{кi}(1)$ при изменении i от 1 до N_{Z1}, то продолжение процесса обслуживания заявки второго приоритета откладывается до момента освобождения канала.

После этого рассматривается возможность «дообслуживания» заявки. С этой целью производится корректировка времени начала и окончания обслуживания заявки по формулам:

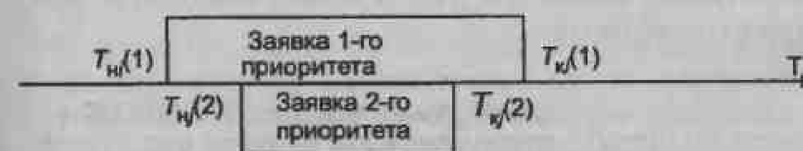
$$T_{нj}(2) = T_{к,fix};$$

$$T_{кj}(2) = T_{к,fix} - [\ T_{кj}(2) - T_{н,fix}\ ],$$

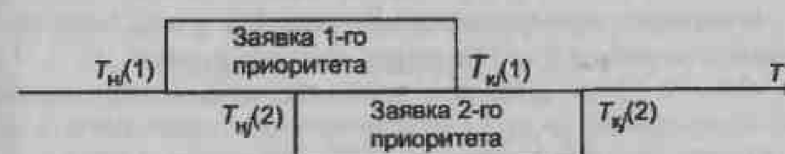
где $T_{н,fix}$ – фиксированное время начала обслуживания заявки первого приоритета, для которой выполняется условие (1.6);
$T_{к,fix}$ – фиксированное время окончания обслуживания заявки первого приоритета, для которой выполняется условие (1.6).

После этого вновь рассматривается возникшая ситуация.

Ситуация 3. Заявка второго приоритета поступила в период обслуживания заявки первого приоритета. Следовательно, заявка второго приоритета не может быть принята к обслуживанию. Два возможных варианта этой ситуации иллюстрируются схемой, показанной на рис. 1.13.



а



б

Рис. 1.13. Схемы вариантов 3-й ситуации:
а – первый вариант; *б* – второй вариант

Логическое условие, при котором создается любой из вариантов 3-й ситуации, записывается так:

$$\{ T_{нi}(1) < T_{нj}(2)\} \text{ AND } \{T_{нj}(2) < T_{кi}(1) \}. \qquad (1.7)$$

Если условие (1.7) выполняется для какой-либо пары значений переменных $T_{нi}(2)$ и $T_{кi}(2)$ при изменении i от 1 до N_{Z1}, то производится «сдвиг» времени начала и окончания обслуживания заявки по формулам:

$$T_{нj}(2) = T_{к,fix};$$

$$T_{кj}(2) = T_{к.fix} - [T_{кj}(2) - T_{нj}(2)].$$

После этого вновь рассматривается возникшая ситуация для «сдвинутой» заявки.

«Дообслуживаемая», или «сдвинутая», заявка, в свою очередь, может оказаться в одной из трех перечисленных выше возможных ситуаций.

В конечном счете процесс обслуживания может иметь два исхода:

1) заявка будет обслужена до конца;

2) истечет время функционирования системы, и заявка останется необслуженной (так же, как и все последующие заявки второго приоритета).

1.10. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ОБСЛУЖИВАНИЯ ЗАЯВОК В УСЛОВИЯХ ОТКАЗОВ

В системах, включающих технические подсистемы, возможно возникновение отказов. Различают два рода отказов.

Отказы первого рода (неисправности) приводят к временному прекращению процесса обслуживания очередной заявки с сохранением достигнутого состояния. После устранения отказа процесс обслуживания заявки может продолжаться. В качестве примера можно указать отказ оборудования бензоколонки. После устранения неисправности заправка автомашины продолжается.

Отказы второго рода (аварии) приводят к такому состоянию системы, что после устранения отказа процесс обслуживания заявки начинается сначала. Примером может служить временное отключение электропитания при работе персонального компьютера во время решения расчетной задачи. После устранения аварии процесс решения задачи начинается сначала.

Время возникновения отказов в системе следует считать случайным событием. Период устранения отказа также может рассматриваться как случайный отрезок времени. Принято считать, что период безотказной работы и период устранения отказа имеют показательные распределения с определенными параметрами.

Функция плотности для времени безотказной работы

$$f(\tau_о) = \lambda_о \exp(-\lambda_о \tau_о),$$

где $\tau_о$ – время безотказной работы;
$\lambda_о$ – параметр (интенсивность потока отказов).

Функция плотности для времени устранения отказа

$$f(\tau_у) = \lambda_у \exp(-\lambda_у \tau_у),$$

где $\tau_у$ – время устранения отказа;
$\lambda_у$ – параметр (среднее число устраненных отказов в единицу времени).

Особенностью взаимодействия периодов безотказной работы и периодов устранения отказов является то, что они не могут пересекаться или накладываться друг на друга. Эти периоды должны чередоваться. Поэтому интервал между двумя соседними отказами должен рассматриваться как сумма (композиция) двух распределений случайных величин $\tau_о$ и $\tau_у$.

Можно показать, что композиция этих распределений приводит к обобщенному ***потоку Эрланга 2-го порядка***, плотность которого имеет вид:

$$f(\tau) = \frac{\lambda_о \lambda_у [\exp(-\lambda_о \tau) - \exp(-\lambda_у \tau)]}{\lambda_у - \lambda_о}.$$

Схема алгоритма формирования одиночного отказа показана на рис. 1.14.

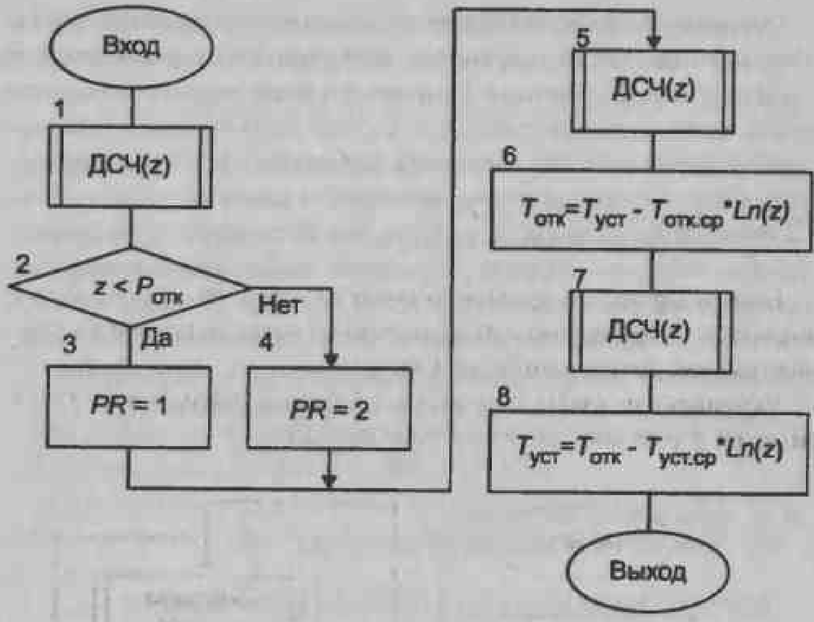


Рис. 1.14. Схема алгоритма формирования одиночного отказа

Оператор 1 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1). Условный оператор 2 служит для розыгрыша по жребию рода отказа. Если условие $z < P_{отк}$ выполняется (где $P_{отк}$ – вероятность появления отказа первого рода), то считается, что произошел отказ первого рода. В противном случае считается, что произошел отказ второго рода.

Оператор 5 вновь обращается к датчику случайных чисел с равномерным распределением в интервале (0,1), а оператор 6 формирует время появления отказа по формуле

$$T_{отк} = T_{уст} - T_{отк.ср} * \mathrm{Ln}(z),$$

где $T_{уст}$ – время устранения предыдущего отказа (в начале работы системы $T_{уст} = 0$);
$T_{отк.ср}$ – среднее время безотказной работы системы.

Оператор 7 еще раз обращается к датчику случайных чисел с равномерным распределением в интервале (0,1), а оператор 8 формирует время устранения отказа по формуле

$$T_{уст} = T_{отк} - T_{уст.ср} * \mathrm{Ln}(z),$$

где $T_{уст}$ – время появления отказа;
$T_{уст.ср}$ – среднее время устранения отказа.

Если в процессе функционирования системы отказы могут возникать неоднократно, то обращение к процедуре формирования отказов производится несколько раз.

Укрупненная схема алгоритма процесса функционирования системы с отказами показана на рис. 1.15.

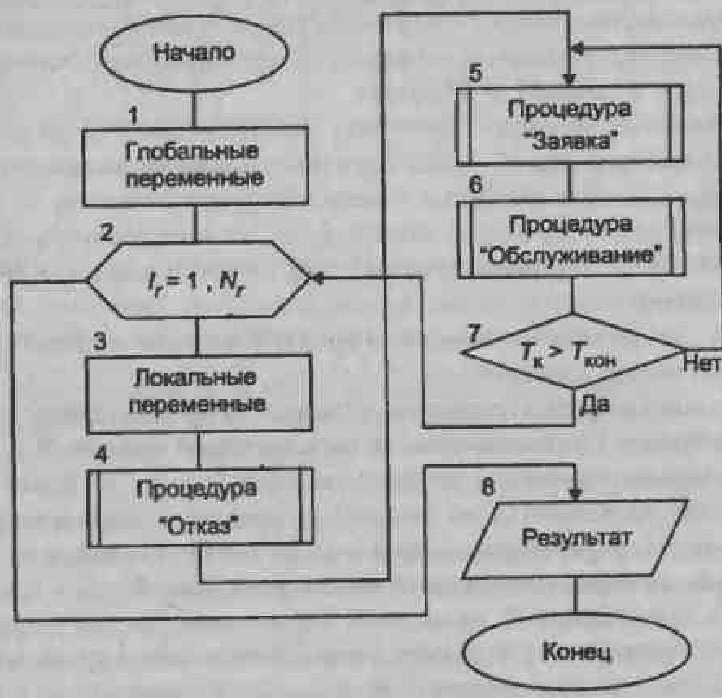


Рис. 1.15. Укрупненная схема алгоритма процесса функционирования системы с отказами

Рассматривается одноканальная система массового обслуживания с однородными заявками. Оператор 1 используется для обнуления глобальных переменных. Оператор 2 представляет собой заголовок циклического перебора случайных реализаций. Оператор 3 служит для обнуления локальных переменных. Оператор 4 обращается к процедуре формирования одиночного отказа, схема которой приведена на рис. 1.14.

Оператор 5 на рис. 1.15 обращается к процедуре формирования одиночной заявки. В этой процедуре определяется случайное время поступления заявки с учетом возможного времени ожидания начала обслуживания. Здесь же определяется возможное время окончания обслуживания без учета возможности появления отказа.

Оператор 6 обращается к процедуре обслуживания заявок при условии возникновения отказов. Внутри этой процедуры имеются операторы обращения к процедуре формирования одиночных отказов и к процедуре «Анализ».

Условный оператор 7 проверяет условие окончания процесса функционирования системы. Если это условие не выполняется, то управление в алгоритме передается вновь оператору 5 для формирования очередной заявки. Если условие выполнено, то управление в алгоритме передается на начало цикла случайных реализаций.

После окончания расчетов оператор 8 выводит на экран результаты моделирования.

Схема алгоритма процедуры «Анализ» приведена на рис. 1.16.

Оператор 1 устанавливает на нуль числовой признак F.

Условный оператор 2 проверяет условие $T_к < T_{отк}$ (условие 1). Если оно выполняется, то числовому признаку F присваивается значение 1, а управление передается на конец процедуры.

Если в условном операторе 4 выполняется условие $((T_н < T_{отк})$ и $(T_{отк} < T_к))$ (условие 2), то оператор 5 присваивает числовому признаку F значение 2 и управление передается на конец процедуры.

Если в условном операторе 6 выполняется условие $((T_{отк} < T_н)$ и $(T_н < T_{уст}))$ (условие 3), то оператор 7 присваивает числовому признаку F значение 3 и управление передается на конец процедуры.

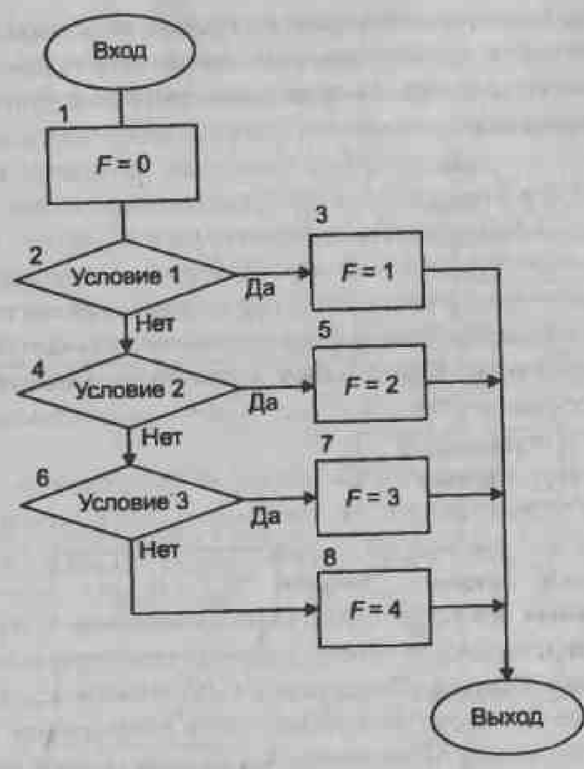


Рис. 1.16. Схема алгоритма процедуры «Анализ»

Наконец, если ни в одном из условных операторов не выполняются проверяемые условия, то оператор 8 присваивает числовому признаку F значение 4.

Схема алгоритма процедуры обслуживания заявок при наличии отказов приведена на рис. 1.17.

В состав исходных данных должны быть включены: время поступления заявки $T_н$, время возможного завершения обслуживания $T_к$ и числовой признак рода отказа PR, принимающий значение 1 для отказа 1-го рода и значение 2 для отказа 2-го рода.

Условный оператор 1 проверяет условие окончания процесса обслуживания. Если оно выполняется, то управление в алгоритме передается на выход из процедуры. Если же условие не выполняется, то оператор 2 обращается к процедуре анализа ситуации, возникающей в случае появления отказа. Процедура «Анализ» (описание которой приведено выше) вырабатывает значения числового признака F.

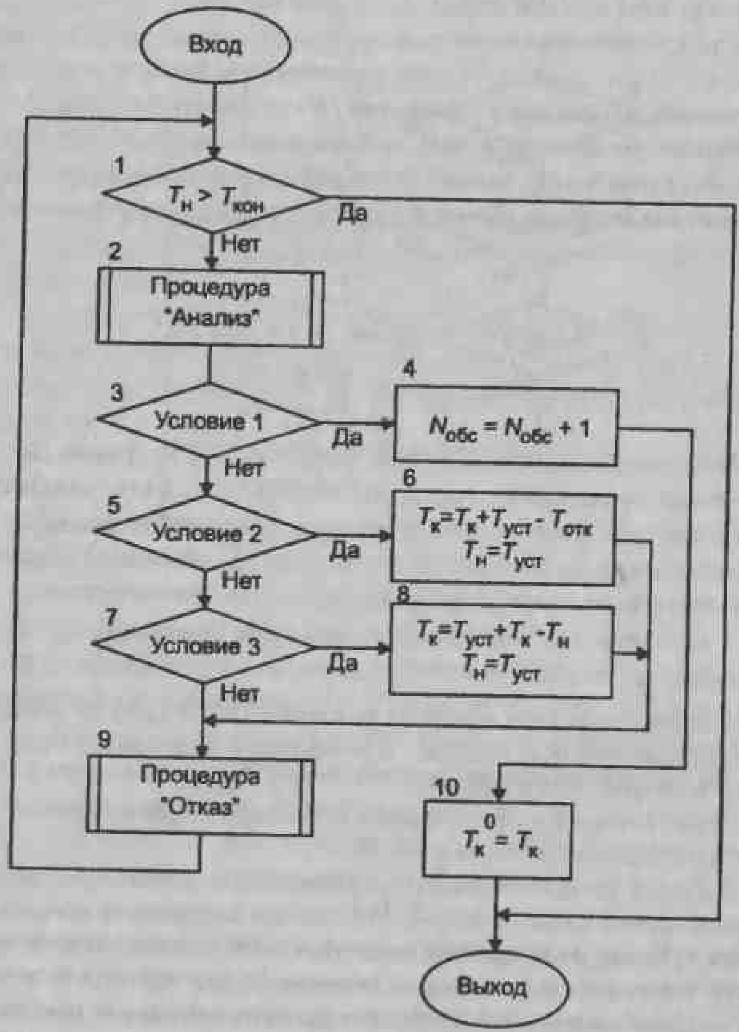


Рис. 1.17. Схема алгоритма процедуры обслуживания заявок при наличии отказов

Если выполняется условие 1 (F=1), то это означает, что отказ появился после того, как процесс обслуживания заявки был полностью завершен. В этом случае оператор 4 увеличивает на единицу показание счетчика числа обслуженных заявок, а затем управление в процедуре передается оператору 10.

Если выполняется условие 2 (F=2 и PR=1), то это означает, что отказ прервал обслуживание рассматриваемой заявки. После устранения отказа может происходить «дообслуживание» заявки. Однако это возможно только в том случае, если не произойдет новый отказ. Поэтому оператор 5 производит корректировку времени начала и окончания «дообслуживания» по формулам:

$$T_н = T_{уст};$$

$$T_к = T_к + T_{уст} - T_{отк},$$

где $T_н$ – время начала «дообслуживания» заявки;
$T_к$ – время окончания «дообслуживания» заявки.

Последняя формула является рекурсивной. В правой части помещено предыдущее значение времени окончания «дообслуживания», а в левой части – его скорректированное значение.

Далее управление в алгоритме передается оператору 9, который обращается к процедуре формирования нового отказа.

Если выполняется условие 3 ((F=2 и PR=2) или F=3), то это означает, что обслуживание заявки прервал отказ второго рода или заявка поступила в момент, когда происходит устранение отказа.

Оператор 8 производит корректировку времени по формулам:

$$T_н = T_{уст};$$

$$T_к = T_{уст} + T_к - T_н.$$

Выражение для $T_к$ является рекурсивным, т.е. в правой его части помещено предыдущее значение времени окончания обслуживания заявки, а в левой части – последующее значение. После этого управление в алгоритме передается оператору 9 для формирования очередного отказа.

После того как будет сформирован очередной отказ, работа алгоритма начинается сначала, т. е. с оператора 1.

Выход из процедуры может произойти только в двух случаях:

1) если закончится период функционирования системы, т. е. выполнится условие

$$T_н > T_{кон};$$

2) если при анализе положения заявки относительно потока отказов выяснится, что числовой признак F=1, т. е. обслуживание заявки закончилось до появления очередного отказа.

В последнем случае оператор 10 фиксирует время окончания обслуживания: $T_к^о = T_к$. Величина $T_к^о$ используется в дальнейшем в процедуре формирования очередной заявки для регулирования очередности в обслуживании заявок.

2. КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ЭКОНОМИЧЕСКИХ СИСТЕМ

В настоящей книге рассматриваются не сами экономические системы, а лишь методы разработки алгоритмических статистических моделей этих систем. Разработкой классификации экономических систем должны заниматься экономисты. Насколько можно судить по литературным источникам общепринятой классификации, таких систем пока не существует.

Первая классификация математических моделей экономических систем была приведена в монографии Т. Нейлора «Машинные имитационные эксперименты с моделями экономических систем» в 1971 г. Он предлагал разделить их все на две группы (рис. 2.1):

1) общие экономические модели;

2) модели управления предприятиями.

В основу классификации общих экономических моделей заложен масштаб изучаемой экономической системы. С этой точки зрения модели можно разделить на три большие группы: модели фирм, отраслевые модели и макроэкономические модели.

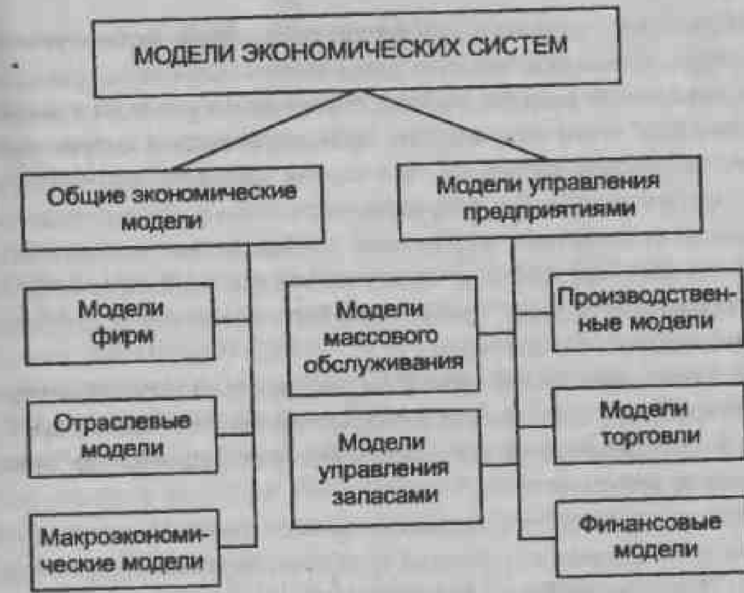


Рис. 2.1. Классификация алгоритмических экономических моделей по Т. Нейлору

Представителями моделей фирм являются:

- модели отдельных фирм;
- модели конкурентных отраслей;
- модели дуополий (объединений двух фирм);
- модели олигополий (объединений нескольких фирм);
- модели монополий.

Опыт создания моделей фирм в США, обобщенный Т. Нейлором, показывает, что разработка математических моделей даже для систем такого масштаба, как фирма, представляет сложную научно-исследовательскую проблему.

Во-первых, это проблема получения достоверной информации. Модель должна строиться на прочной эмпирической основе. Однако эта информация, как правило, недоступна для разработчиков экономических моделей. Руководство компаний просто не желает давать данные о деятельности своих предприятий посторонним лицам. Это особенно характерно для фирм, работающих в условиях сильной конкуренции.

Во-вторых, трудности построения адекватной численной модели фирмы связаны с тем, что такая модель должна опираться на глубокое знание реальных процессов принятия решений в организациях. Для этого надо хорошо ориентироваться в современном состоянии таких дисциплин, как теория принятия решений, теория организации, а также разбираться в вопросах психологии, социологии, политики, управления производством и экономики.

В-третьих, организация численных испытаний модели функционирования фирмы требует особого внимания к проблеме планирования эксперимента.

В итоге применение мощного аппарата алгоритмического моделирования оказывается неэффективным, так как в этих условиях традиционные аналитические методы дают не менее надежные результаты.

В качестве примера в разделе 3 1-й части рассматривается одна из моделей фирмы, получившая название «паутинообразной» модели. Это простейшая динамическая модель взаимодействия фирмы и рынка. В качестве типовой математической схемы вначале была выбрана непрерывно-детерминированная модель, в которую добавлены случайные составляющие входных переменных, в результате чего модель становится стохастической. Особенностью рассматриваемого варианта модели также является то, что это модель с обучением, т. е. с учетом тенденции развития моделируемого процесса.

К отраслевым моделям относятся комплексные, или агрегированные, модели, описывающие отдельные отрасли народного хозяйства как единое целое. В монографии Т. Нейлора дается краткая характеристика модели текстильной промышленности США, модели кожевенной и обувной промышленности, а также модели лесообрабатывающей промышленности. В большинстве случаев отраслевая модель представляет собой систему рекуррентных уравнений. Для нахождения коэффициентов этих уравнений используется метод наименьших квадратов.

Макроэкономические модели предназначены для имитации экономических систем крупного масштаба, таких, как область (штат) или страна в целом. С чисто технической точки зрения механизм численного моделирования экономики описывается теми же правилами, что и при имитации процесса функционирования фирмы, отрасли или их подразделений. По-прежнему надо определить структуру изучаемой системы, входные и выходные переменные, сформулировать задачу моделирования, построить схему алгоритма и написать программу. Однако по существу алгоритмические модели глобальных экономических систем сильно отличаются от микроэкономических моделей, и эти отличия связаны с проблемой вывода адекватных уравнений функционирования экономики в целом:

- входные переменные макроэкономической системы, такие, как национальный доход, национальный продукт и общая численность работающих, по-видимому, зависят от большого числа существенных факторов. Их количество обычно намного превышает число переменных, рассматриваемых в численных микроэкономических моделях;
- возникает проблема «агрегирования» микроэкономических переменных в обобщенные показатели макроэкономической системы;
- между входными переменными существуют очень сложные взаимодействия и обратные связи;
- для формулировки реалистических гипотез относительно функционирования экономики требуются глубокие знания закономерностей ее развития. Исследователь, пытающийся восполнить недостаток этих знаний с помощью имитационных экспериментов, скорее построит модель собственного невежества, а не реального мира;
- получить данные для построения математической модели макроэкономической системы намного сложнее, чем для микроэкономических систем.

По приведенным причинам в данной книге отраслевые и макроэкономические модели не рассматриваются.

Перейдем к рассмотрению классификации экономических систем, которые можно условно назвать ***моделями управления предприятием***. Это микроэкономические модели, отличающиеся друг от друга не столько областью применения, сколько тем, какая типовая математическая схема заложена в основу модели и каковы особенности используемого математического аппарата.

К моделям управления предприятиями относятся (см. рис. 2.1):

• модели массового обслуживания;
• модели управления запасами;
• производственные модели;
• модели торговли;
• финансовые модели.

Для многих промышленных систем характерен поток входных требований (заявок), поступающих в один или несколько каналов обслуживания и иногда образующих очередь. Заявками могут быть производственные и торговые заказы, заявки на ремонт станков, посадку самолетов в аэропорту и заправку автомобилей на автозаправочной станции. Канал обслуживания может представлять собой совокупность устройств, этап производственного процесса, аэропорт или театральную кассу. Интервалы между последовательными заявками и продолжительность их обслуживания являются случайными величинами.

В разделе 4 рассмотрена алгоритмическая модель, условно названная моделью бензоколонки. С таким же успехом ее можно было бы назвать моделью нотариальной конторы, моделью парикмахерской, моделью столовой самообслуживания, моделью станции автомобильного обслуживания и т. п. Общей для этих моделей является заложенная типовая математическая схема – схема системы массового обслуживания с переменным числом каналов, однородным потоком заявок, без отказов и с ограниченным ожиданием. В задачу моделирования входит установление оптимального числа каналов, которое при определенном соотношении входных параметров (среднего времени между соседними заявками и среднего времени обслуживания) обеспечивает максимальное значение показателя эффективности процесса функционирования системы. Для конкретной экономической системы в качестве критерия эффективности используется условие максимума прибыли.

В разделе 6 рассматривается алгоритмическая модель производственной фирмы, включающей несколько цехов, которые последовательно участвуют в процессе производства некоторого изделия. Заказы на изготовление изделия поступают нерегулярно (в случайные моменты времени). При оптимальной структуре предприятия (количестве цехов) и оптимальном распределении производственных ресурсов обеспечивается максимум прибыли. Имитация работы предприятия производится с помощью модели одноканальной многофазной системы массового обслуживания, без отказов с неограниченным ожиданием.

В разделе 9 описана модель, условно названная моделью управленческого звена учреждения, фирмы или предприятия, состоящего из начальника (заведующего) и двух его заместителей. Все они принимают участие в процессе приема посетителей или обработки документации. Часть посетителей, побывавших на приеме у одного из заместителей, затем отправляется на прием к начальнику. При определенном соотношении параметров системы можно обеспечить практически одинаковую занятость каждого из трех должностных лиц. С точки зрения используемого математического аппарата это модель двухканальной двухфазной системы массового обслуживания с двумя приоритетами заявок, без отказов с неограниченным ожиданием.

Обширную группу промышленных систем, при изучении которых эффективна численная имитация, образуют так называемые системы хранения запасов. Большинство задач управления запасами сводится к поиску оптимального распределения поставок в моделируемую систему. Модель должна дать ответ на вопрос: сколько следует фирме заказывать (или производить) и как часто она должна повторять заказы, чтобы минимизировать сумму издержек хранения запаса, издержек, связанных с организацией поставок, и потерь вследствие недостатка продукта на складе?

Одна из моделей управления запасами рассматривается в разделе 5. Это система управления запасом однородного товара на складе. Предполагается, что когда уровень запаса падает ниже некоторой критической отметки, оформляется заказ на поставку новой партии товара. При отсутствии товара на складе применяются штрафные санкции. При определенном соотношении параметров системы суммарные расходы на содержание склада могут быть минимизированы.

С точки зрения используемого математического аппарата это алгоритмическая модель, в которой две входные переменные (дневной спрос и время выполнения заказа) являются случайными величинами, что определяет случайный характер выходной характеристики – суммарных издержек, характеризующих работу склада за определенный период времени. Время между соседними заявками на приобретение товара и время на выдачу товара в модели не фигурируют. Поэтому эту модель нельзя отнести к классу моделей СМО, однако это тоже непрерывно-стохастическая модель.

Из группы производственных моделей в настоящем издании рассматривается модель фирмы. Как было отмечено выше, эта модель описана в разделе 6 как представитель класса моделей СМО.

Одна из моделей торговли рассматривается в разделе 7. Это модель выездной торговой точки, которая может вести торговлю в различных пунктах с различными условиями при действии случайных факторов. Задача состоит в установлении закономерностей моделируемого процесса и установлении условий, при которых пункты торговли могут считаться эквивалентными по получаемой прибыли.

В качестве типовой математической схемы здесь использована общая непрерывно-стохастическая модель, в которой имитируется влияние дискретных и непрерывных случайных факторов. Такая модель при изменении комплекта исходных данных может использоваться как вариант транспортной модели, в которой осуществляется имитация процесса перевозки грузов по нескольким маршрутам в условиях влияния случайных помех и непостоянства скорости движения на различных участках дороги.

В разделе 8 рассматривается финансовая модель, с помощью которой определяется объем капиталовложений в условиях неопределенности. Используемая типовая математическая схема

представляет собой непрерывно-стохастическую модель. Для раскрытия неопределенности необходимо выбрать в качестве одной из входных переменных случайную величину, имеющую произвольное дискретно-непрерывное распределение.

3. «ПАУТИНООБРАЗНАЯ» МОДЕЛЬ ФИРМЫ

3.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель: – Я собираюсь вложить средства в создание фирмы, которая будет выпускать товар и реализовывать его на рынке. Меня интересует, как будет вести себя цена на товар при изменении объема производства. Опыт подсказывает мне, что при увеличении производства происходит падение спроса и приходится снижать цену. Мне хотелось бы знать, при каких условиях цена будет стабильной. Можно ли дать ответ на этот вопрос с помощью математической модели?

Консультант: – Конечно, можно. В литературе описано несколько вариантов такой модели. Все они обладают определенными одинаковыми свойствами. Обычно в них предполагается, что спрос на некоторый продукт (чаще всего рассматривается сельскохозяйственная продукция) на заданном отрезке времени зависит от цены (и других факторов) на этом отрезке. Что же касается предложения, то оно определяется ценами предыдущего периода времени (недели, месяца, квартала и т. д.). Кроме того, предполагается, что рынок всегда находится в условиях локального равновесия. Исторически такая модель получила название «паутинообразной», вероятно, потому, что такого же принципа «учета предыдущего шага» придерживается паук, когда он ткет паутину.

Существуют четыре варианта этой модели: детерминированная, вероятностная, модель с обучением и модель с запасами.

Предприниматель: – А чем они отличаются друг от друга?

Консультант: – В детерминированной модели отсутствует учет влияния случайных факторов. В вероятностной модели учитываются влияние на спрос непредвиденных колебаний предпочтений и доходов потребителей, а также другие случайные факторы, влияющие на величину спроса. Предложение на предыдущем отрезке времени также считается подверженным влиянию случайных факторов. Они отражают влияние колебаний технологии и эффективности производственного процесса и т. д. Наконец, условие локального равновесия означает совпадение спроса и предложения с точностью до некоторой случайной величины.

В модели с обучением предполагается, что поставщики учитывают сложившуюся тенденцию изменения цен и с учетом этого планируют выпуск продукции на очередной отрезок времени.

В последних двух моделях цены устанавливаются на таком уровне, чтобы обеспечить локальное равновесие рынка только за счет текущего производства, и никаких запасов продукции не создается (например, потому, что продукты быстро портятся).

В модель с запасами вводится дополнительная группа участников рыночного механизма, которых можно назвать «коммерсантами». Они держат запасы и организуют торговлю.

Предприниматель: – Для моего случая, наверное, больше подойдет вероятностная модель с обучением. При каких допущениях она составлена? Как выглядит зависимость для определения текущего спроса?

Консультант: – Предполагается, что спрос на T-м отрезке времени линейно зависит от текущей цены и, кроме того, спрос подвержен случайному разбросу. Таким образом, для описания спроса нужно задать коэффициенты линейного уравнения (например, A и B) и случайную величину (например, U_T), имеющую заданное распределение.

В результате получается расчетная формула следующего вида:

$$D_T = A - B\rho P_T + U_T, \qquad (3.1)$$

где D_T – цена на T-м отрезке времени;
A, B – коэффициенты линейного уравнения;
P_T – подлежащая определению цена на T-м отрезке времени;
U_T – случайная величина с заданным законом распределения.

Предприниматель: – А что означает знак «минус»?

Консультант: – То, что с повышением цены спрос на продукцию снижается.

Предприниматель: – Какое именно распределение следует выбрать в этом случае?

Консультант: – Логично предположить, что спрос симметрично колеблется относительно среднего значения, которое определяется постоянными коэффициентами линейного уравнения. Поэтому можно выбрать нормальное распределение с нулевым математическим ожиданием и заданным средним квадратическим отклонением (СКО) – σ_u.

Предприниматель: – А как определяется предложение?

Консультант: – Предполагается, что предложение на текущем отрезке также линейно зависит от цены, но не текущей, а представляющей собой некоторую комбинацию цен на двух предыдущих отрезках времени. В простейшем случае это может быть средняя цена. Поэтому для расчета предложения используется следующая зависимость:

$$S_T = C + E \cdot P(\rho) + V_T, \qquad (3.2)$$

где S_T – предложение на Т-м отрезке времени;
C, E – коэффициенты линейного уравнения;
$P(\rho)$ – среднее (точнее, средневзвешенное) значение цены на двух предыдущих отрезках времени;
V_T – случайная величина с заданным законом распределения.

Предприниматель: – И опять нормальное распределение?

Консультант: – Да, конечно, с теми же основаниями. Только с другим СКО – σ_v.

Предприниматель: – А как определяется средняя цена за предшествующий период?

Консультант: – Средневзвешенная цена определяется по формуле

$$P(\rho) = P_{T-1} - \rho(P_{T-1} - P_{T-2}), \qquad (3.3)$$

где P_{T-1} – цена на (T–1)-м отрезке времени;
P_{T-2} – цена на (T–2)-м отрезке времени;
ρ – весовой коэффициент, значение которого задается в модели в диапазоне ($0 \le \rho \le 1$).

Нетрудно убедиться в том, что при $\rho = 0$ средневзвешенная цена $P(\rho) = P_{T-1}$. Это означает, что обучение в модель не заложено. Для другого крайнего случая (при $\rho = 1$) средневзвешенная цена $P(\rho) = P_{T-2}$. Это также означает, что обучение в модели отсутствует, но для определения предложения используется более удаленная цена. Наконец, при $\rho = 0{,}5$ средневзвешенная цена $P(\rho)$ равна среднему арифметическому значению из цен P_{T-1} и P_{T-2}.

Предприниматель: – Какие еще зависимости входят в модель?

Консультант: – Нужно еще добавить уравнение локального равновесия рынка, которое можно записать так:

$$S_T = D_T + W_T\,, \tag{3.4}$$

где S_T – предложение на T-м отрезке времени;
D_T – спрос на T-м отрезке времени;
W_T – случайная величина с заданным распределением.

Предприниматель: – С нормальным распределением?

Консультант: – Конечно, так как для выбора других распределений нет особых оснований. Можно было бы взять усеченное нормальное распределение, но не ясно, какова должна быть величина усечений. Случайная величина W_T характеризуется нулевым математическим ожиданием и средним квадратическим отклонением σ_w.

Предприниматель: – А как же все-таки в модели определяется закон изменения цены на продукт во времени?

Консультант: – Система уравнений (3.1), (3.2), (3.3) и (3.4) после преобразований сводится к выражению вида

$$P_T = F(P_{T-1}, P_{T-2}), \tag{3.5}$$

где $F(P_{T-1}, P_{T-2})$ – функциональная связь между переменными.

Вначале необходимо каким-либо приближенным способом определить цену для первых двух отрезков времени. После этого можно производить вычисления по зависимости (3.5) неограниченное число раз. Результаты расчетов удобнее всего представить в виде графика.

3.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Пусть имеется торговая фирма, реализующая некоторый товар на рынке.

Спрос на товар на T-м отрезке времени линейно зависит от текущей цены P_T и случайной переменной U_T, учитывающей влияние случайных факторов на величину спроса. Переменная U_T имеет нормальное распределение с нулевым математическим ожиданием M_u и заданным СКО σ_u. Таким образом, зависимость для спроса на товар имеет следующий вид:

$$D_T = A - B \cdot P_T + U_T.$$

Предложение на Т-м отрезке времени рассчитывается с учетом обучения системы. Поэтому оно зависит от цены на предыдущих $(T-1)$-м и $(T-2)$-м отрезках времени и случайной переменной V_T, которая учитывает влияние случайных факторов на величину предложения. Переменная V_T имеет нормальное распределение с нулевым математическим ожиданием M_v и заданным СКО σ_v. Таким образом, зависимости для предложения имеют следующий вид:

$$S_T = C + E \cdot P(\rho) + V_T,$$

$$P(\rho) = P_{T-1} - \rho \cdot (P_{T-1} - P_{T-2}),$$

где ρ – весовой коэффициент, задаваемый в диапазоне $(0 \le \rho \le 1)$.

Условие локального равновесия рынка означает совпадение спроса и предложения с точностью до случайной величины W_T. Предполагается, что переменная W_T имеет нормальное распределение с нулевым математическим ожиданием M_w и заданным СКО σ_w. Зависимость, учитывающая равновесие рынка, имеет вид:

$$S_T = D_T + W_T. \tag{3.6}$$

Подставляя выражения для D_T, $P(\rho)$ и S_T в (3.6) и разрешая уравнение относительно P_T, получаем:

$$P_T = \{A - C - E \cdot [P_{T-1} - \rho \cdot (P_{T-1} - P_{T-2})] + U_T - V_T + W_T\}/B. \tag{3.7}$$

Поскольку для определения величины P_T необходимо знать значения P_{T-1} и P_{T-2} для двух предыдущих отрезков времени, то проводить расчеты по формуле (3.7) можно только, начиная с 3-го отрезка, при условии, что P_1 и P_2 известны.

Для их нахождения сделаем дополнительное допущение о том, что на первых двух отрезках обучение отсутствует, т. е. весовой коэффициент ρ=0. Без учета случайностей цена на 2-м отрезке определится по формуле

$$P_2 = (A - C - E \cdot P_1) / B. \tag{3.8}$$

Если предположить, что перед началом работы фирмы исходная цена совпадает с ценой на 1-м отрезке, то величина P_1 определится по формуле

$$P_1 = (A - C) / (B + E). \tag{3.9}$$

Задача моделирования заключается в исследовании влияния параметров системы на характер зависимости цены от времени.

3.3. СХЕМЫ АЛГОРИТМОВ МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0. Один из возможных вариантов общего вида (макета) стартовой формы показан на рис. 3.1.

В верхней части окна помещено 8 текстовых полей для ввода и корректировки исходных данных (параметров A, B, C, E, Ro, SigU, SigV и SigW). В нижней части окна располагаются три командные кнопки. Кнопка «Расчет» предназначена для проведения расчетов. Кнопка «График» служит для построения в центральной части окна графика зависимости цены от времени. Кнопка «Выход» используется при окончании работы с программой.

Схема алгоритма процедур обработки прерываний показана на рис. 3.2. После подачи команды Start на экране появляется активная стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

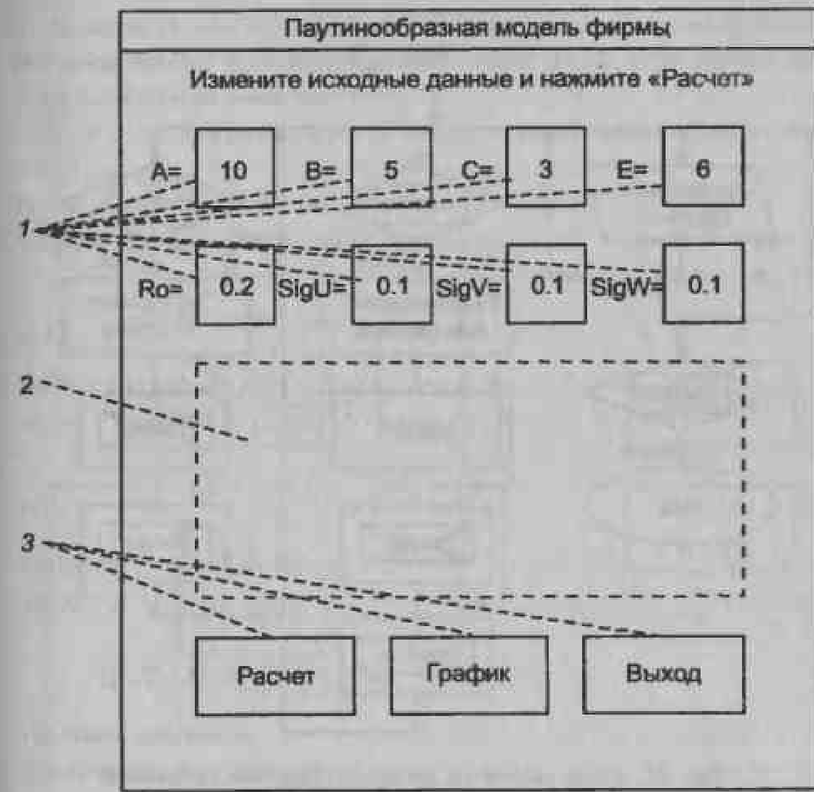


Рис. 3.1. Макет стартовой формы:
1 – текстовые поля; *2* – область для графика; *3* – командные кнопки

Цифрой 1 на схеме обозначено действие пользователя, которое заключается в корректировке исходных данных. Измененные данные вводятся в соответствующие текстовые поля. При этом они фиксируются в памяти не как числа, а как значения символьных переменных. Цифрой 2 на схеме обозначено действие пользователя, которое заключается в нажатии командной кнопки «Расчет». В результате этого вызывается одна из так называемых процедур прерываний. Внутри этой процедуры оператор 3 производит преобразование символьных данных в числовые.

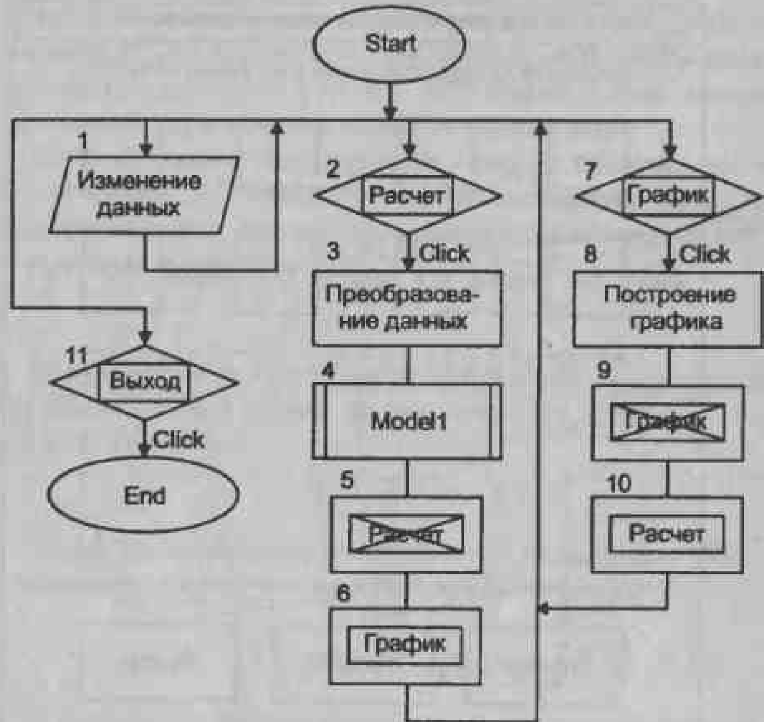


Рис. 3.2. Схема алгоритма процедур обработки прерываний

Затем оператор 4 обращается к программному модулю общего назначения «Model1», который производит расчет массива значений цен как функций времени. После окончания работы программного модуля оператор 5 делает кнопку «Расчет» неактивной, а оператор 6 активизирует кнопку «График». Одновременно производится очистка части стартовой формы, которая отведена для построения графика.

Цифрой 7 на схеме обозначено действие пользователя, которое заключается в нажатии кнопки «График» (если она активна). В результате группа операторов 8 обеспечивает построение в центре стартовой формы графика зависимости текущей цены на продукт от времени. Затем оператор 9 делает кнопку «График» неактивной, а оператор 10 вновь активизирует кнопку «Расчет».

Числом 11 на схеме обозначено действие пользователя, которое заключается в нажатии кнопки «Выход». В этом случае работа программы заканчивается.

Схема алгоритма модуля «Model1» показана на рис. 3.3.

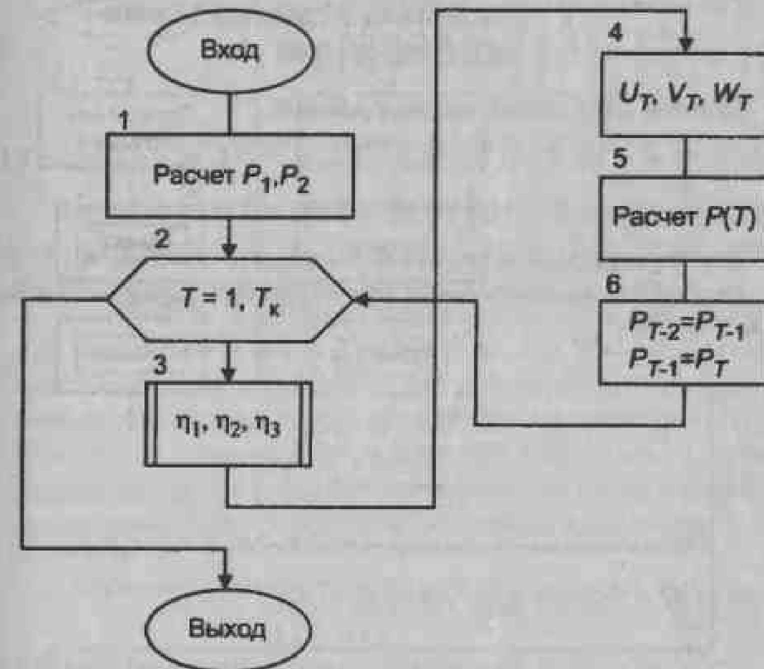


Рис. 3.3. Схема алгоритма модуля *«Model1»*

Внутри этого модуля группа операторов 1 определяет цены для 1-го и 2-го отрезков времени по формулам (3.8) и (3.9). Оператор 2 является началом циклического перебора временных отрезков, начиная с 3-го и кончая последним $T_{K\text{–}M}$.

Группа операторов 3 вырабатывает три возможных значения эталонной (нормированной и центрированной) случайной величины η с нормальным распределением, которые используются группой операторов 4 для расчета возможных значений случайных переменных U_T, V_T, W_T с заданными СКО. Оператор 5 осуществляет расчет выходной переменной P_T по формуле (3.7). Оператор 6 подготавливает новые значения переменных P_{T-1} и P_{T-2} для расчета P_T на следующем временном отрезке (следующем витке цикла).

3.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Примем следующие входные данные:

$$T_K = 30;\ A = 10;\ B = 5;\ C = 3;\ \rho = 0.2;\ \sigma_u = 0.1;\ \sigma_v = 0.1;\ \sigma_w = 0.1;$$

$$M_u = 0;\ M_v = 0;\ M_w = 0.$$

Для варьируемой переменной E выберем значения: 4; 5; 5,5.

Результаты моделирования представлены графиками на рис. 3.4, 3.5 и 3.6.

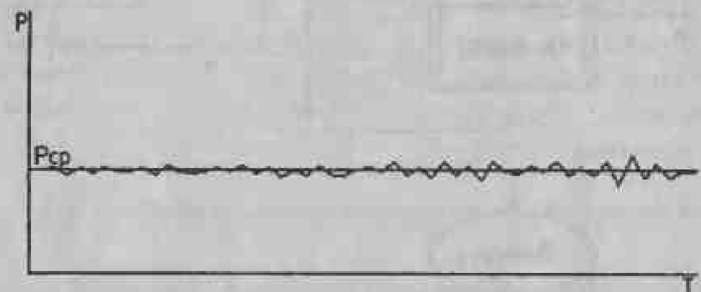


Рис. 3.4. Зависимость цены от времени при E=4 и B=5

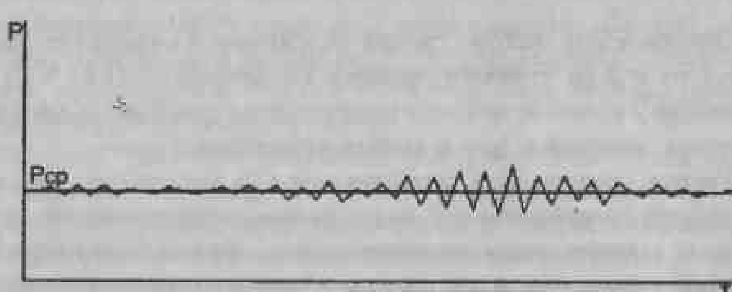


Рис. 3.5. Зависимость цены от времени при E=5 и B=5

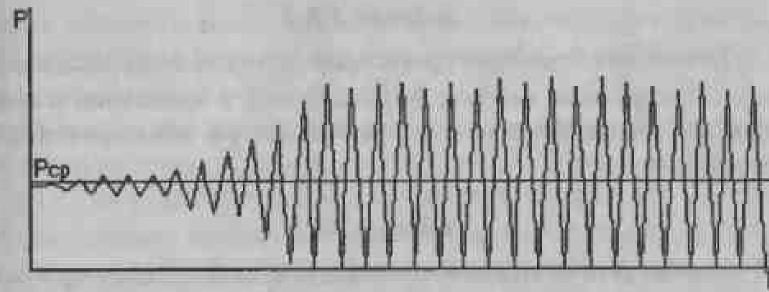


Рис. 3.6. Зависимость цены от времени при E=6 и B=5

Анализ результатов моделирования показывает, что зависимость цены товара P от времени имеет колебательный характер и зависит от соотношения параметров E и B.

Если $E<B$, то колебания незначительны, если $E=B$, то колебания имеют постоянную амплитуду, а если $E>B$, то амплитуда колебаний имеет тенденцию к безграничному возрастанию. Однако по физическим соображениям цена не может быть отрицательной. С учетом этого ограничения возрастание амплитуды происходит до тех пор, пока не начнут появляться нулевые значения цены. После этого колебания стабилизируются.

3.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Для рассматриваемой модели целесообразно совместить табличное представление исходных данных с графическим представлением результатов расчетов. Поэтому наиболее подходящим языком программирования является Quick Basic 4.5 или Visual Basic 5.0.

Задание 3.5.1

В интегральной среде Visual Basic 5.0 создайте исполняемый модуль программы «Паутинообразная модель фирмы», текст которой приведен в приложении 1. Произведите отладку программы с целью ликвидации формальных ошибок.

5—3041

Задание 3.5.2

Произведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 3.5.3

Произведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Самостоятельно выберите исходные данные, проведите расчеты и проанализируйте результаты моделирования.

4. МОДЕЛЬ БЕНЗОКОЛОНКИ

4.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель: – Я собираюсь вложить деньги в строительство новой бензоколонки. Однако точного представления о том, сколько автомашин будет ежедневно заправляться на этой колонке, у меня, конечно, нет. Их число, вероятно, может колебаться в некотором диапазоне. Но я хотел бы ориентировочно знать, какова должна быть оптимальная структура бензоколонки и на получение какой прибыли я могу рассчитывать. Можно ли дать ответы на эти вопросы с помощью математической модели?

Консультант: – В принципе можно. Поскольку входные данные имеют неопределенный характер, это должна быть алгоритмическая статистическая модель. Правда, такой модели у меня под руками нет, но ее можно разработать.

Предприниматель: – А что для этого нужно сделать?

Консультант: – Все начинается с разработки концептуальной модели. Прежде всего нужно выбрать математическую схему, которая ближе всего подходит к такой экономической системе, как бензоколонка. Нужно также установить, что является входными параметрами модели, а что выходными характеристиками. Далее нужно выбрать показатель и критерий эффективности будущей экономической системы.

Потом нужно разработать алгоритм и составить программу на алгоритмическом языке, отладить ее и убедиться в том, что она обеспечивает получение достоверных результатов. Наконец, нужно выбрать конкретные исходные данные и провести серию расчетов при разных значениях входных параметров. Анализ результатов моделирования позволит дать ответ на все Ваши вопросы.

Предприниматель: – А какая математическая схема больше подходит для описания работы бензоколонки?

Консультант: – Схема системы массового обслуживания, или сокращенно СМО. Для таких систем характерны три отличительные особенности:

1) имеется поток клиентов, желающих быть обслуженными (в данном случае это поток автомашин, желающих заправиться бензином);

2) имеются устройства или агрегаты, которые обеспечивают удовлетворение заявок клиентов (в данном случае одна или несколько раздаточных колонок);

3) имеется определенный набор правил обслуживания клиентов (в данном случае можно, например, считать, что все клиенты равноправны, т. е. никто не имеет права на заправку вне очереди).

СМО различаются прежде всего по числу мест или каналов обслуживания (одноканальная, двухканальная и т. д.).

Предприниматель: – Но я пока не знаю, сколько раздаточных колонок выгоднее иметь. Если они будут простаивать, я буду терпеть убытки.

Консультант: – Значит, в модели нужно сделать число каналов обслуживания переменным, т. е. включить его в состав входных параметров. В задачу моделирования будет входить определение оптимального числа каналов.

Предприниматель: – А от чего оно будет зависеть?

Консультант: – От соотношения между средним временем между поступлением заявок (приезд автомашин) и средним временем обслуживания (время заправки), которое нужно задать как входные характеристики модели.

Предприниматель: – А откуда взять эти характеристики?

Консультант: – Среднее время обслуживания можно определить, понаблюдав за работой какой-нибудь действующей бензоколонки. А среднее время между соседними заявками зависит от интенсивности потока автомашин на том участке дороги, где Вы собираетесь строить бензоколонку.

Предприниматель: – Но ведь не все проезжающие автомашины нуждаются в заправке.

Консультант: – Вот и нужно хотя бы приближенно оценить среднее количество автомашин, которые будут заправляться. Это Ваша задача, а не разработчика модели.

Предприниматель: – Но поток автомашин не однороден. Интервалы между соседними автомашинами не постоянны. Как же это будет учтено в модели?

Консультант: – Нужно выбрать подходящий для типичного потока автомашин на данном участке дороги закон распределения случайных величин времени между соседними автомашинами, заезжающими на заправку. Опыт показывает, что лучше всего такой поток описывается показательным распределением с заданным средним значением случайной величины. А возможные значения случайного времени между соседними заявками будут определяться в модели с помощью датчика случайных чисел.

Предприниматель: – Но время обслуживания тоже не постоянно для всех автомашин.

Консультант: – Конечно. Это случайная величина, и нужно определить закон ее распределения. Понаблюдав за работой действующей бензоколонки, можно установить эмпирическое распределение реального времени обслуживания. Но при построении первого варианта модели чаще всего обычно выбирают одно из стандартных распределений, которое ближе всего подходит к полученному эмпирическому распределению. В дальнейшем модель может быть уточнена.

Предприниматель: – Так какое распределение мы выберем для начала?

Консультант: – Я предлагаю остановиться на показательном распределении.

Предприниматель: – Но мы уже выбрали его для времени между соседними заявками.

Консультант: – Ну и что же. Там было одно среднее значение для времени между заявками, а здесь другое – для времени обслуживания.

Предприниматель: – А что еще нужно знать для построения модели?

Консультант: – Нужно высказать предположение о том, как будут себя вести клиенты, если им придется стоять в очереди. В СМО обычно описывается один из трех вариантов режима ожидания: с неограниченным ожиданием; с ограниченным ожиданием и без ожидания.

Предприниматель: – Вероятно, в нашем случае больше подойдет вариант с ограниченным ожиданием. Но я сейчас не могу сказать, сколько времени мои будущие клиенты согласятся ждать в очереди на заправку.

Консультант: – А этого и не требуется. Достаточно ввести в модель в качестве входной переменной максимальное время ожидания. Тогда в процессе моделирования заявка с временем ожидания, превышающим максимально допустимое, будет покидать систему необслуженной.

Предприниматель: – И это все входные переменные?

Консультант: – Осталось только ограничить период функционирования системы. Нужно ввести время начала и время конца работы, чтобы расчеты каждой случайной реализации проводились в одинаковых условиях.

Предприниматель: – Осталось ли что-либо неучтенным?

Консультант: – Конечно. Ведь всякая модель отражает только существенные с точки зрения разработчика свойства объекта-оригинала. Модель проще системы. Мы не учли, например, возможности возникновения отказов при работе раздаточных колонок, не учли перерывов в работе. Но это можно будет учесть при доработке модели.

Предприниматель: – А какие выходные характеристики должна иметь наша модель?

Консультант: – Это зависит от того, что мы примем в качестве показателя эффективности процесса функционирования системы. Это, между прочим, самый важный момент в процессе создания концептуальной модели исследуемой системы.

Предприниматель: – А что такое показатель эффективности?

Консультант: – Эффективность – это часто употребляемое слово, смысл которого не всегда правильно трактуется. В теории эффективности ***показателем эффективности*** называют ***меру степени достижения поставленной цели***. Вы ведь вкладываете средства в строительство бензоколонки с целью получения прибыли?

Предприниматель: – Да, конечно. Но строительство сопряжено и с расходами, которые зависят от структуры бензоколонки, т. е. от количества раздаточных колонок. Как в этом случае может выглядеть показатель эффективности системы?

Консультант: – Нужно выбрать такой показатель эффективности, который отражал бы влияние на прибыль не только доходов, но и расходов. В первом варианте модели можно предположить, что все Ваши будущие клиенты будут заправлять примерно одно и то же количество бензина. Тогда доход будет определяться по формуле:

$$\text{Дох} = C_1 \cdot N_{\text{обс.ср}},$$

где Дох – средний доход за период функционирования системы;
C_1 – средняя стоимость заправки одной автомашины;
$N_{\text{обс.ср}}$ – среднее число заправленных автомашин.

Расходы можно оценить по данным о стоимости строительства действующих бензоколонок. Предположим, что расходы связаны с числом каналов $N_к$ некоторой функциональной зависимостью:

$$\text{Расх} = F(N_к).$$

Итак, показатель эффективности для нашей модели имеет вид:

$$\text{Эфф} = \text{Дох} - \text{Расх}.$$

Предприниматель: – А нельзя ли сформировать безразмерный показатель эффективности? Меня сейчас не интересует абсолютная величина прибыли, мне важно установить с помощью модели структуру системы, которая обеспечивает получение максимально возможной прибыли.

Консультант: – Это очень просто сделать. Поделим доход и расход на коэффициент C_1. Получим следующее выражение для показателя эффективности:

$$C_{\text{отн.ср}} = N_{\text{обс.ср}} - F1(N_к),$$

где $C_{\text{отн.ср}}$ – средняя относительная прибыль;
$F1(N_к)$ – безразмерная функциональная зависимость расходов от числа каналов.

Предприниматель: – Но, кроме показателя эффективности, существует еще критерий эффективности. Что он представляет собой для нашей модели?

Консультант: – В теории эффективности ***критерием эффективности*** называют правило, с помощью которого выбирается наивыгоднейший вариант структуры моделируемой системы. Если имеется несколько показателей эффективности, то критерий объединяет их в единое выражение.

В данном случае показатель один, а поэтому в качестве критерия естественно принять условие достижения максимума этого показателя. На практике это означает, что нужно перебрать несколько вариантов структуры модели при разных значениях входных параметров и установить, при каких условиях выбранный нами показатель эффективности будет достигать максимума. Для наглядности можно привести выражение для определения величины критерия эффективности. Оно будет иметь вид:

$$\text{Критерий эффект.} : MAX\ \{\ C_{\text{отн.ср}}(u)\ \},$$

$$u \in U,$$

где u – порядковый номер варианта расчета, принадлежащий множеству U.

Однако в самой алгоритмической модели эта формула использоваться не будет. Просто ею должен руководствоваться исследователь, производящий расчеты различных вариантов при переменных значениях входных параметров модели.

Предприниматель: – Надеюсь, на этом разработка концептуальной модели заканчивается? Хотелось бы теперь увидеть описание модели в концентрированном виде.

Консультант: – Для этого достаточно прочитать следующий пункт данного раздела.

4.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Пусть имеется система массового обслуживания с переменным числом каналов $N_к$, которое может принимать любое значение в диапазоне от одного до трех. Входной поток заявок – ***простейший***, следовательно, время между соседними заявками имеет показательное распределение с известным математическим ожиданием (средним значением) $T_{з.ср}$.

Время обслуживания заявки в любом канале – величина случайная, имеющая показательное распределение с известным средним временем обслуживания $T_{обс.ср}$.

Все заявки однородны и независимы.

Правило (дисциплина) ***обслуживания*** состоит в том, что очередная заявка поступает в тот канал, который раньше других освободился. Если время ожидания начала обслуживания превышает заданную величину $T_{ож.max}$, то заявка покидает систему необслуженной. Период функционирования СМО характеризуется величиной $T_{кон}$.

Таким образом, входными характеристиками модели являются: число каналов $N_к$, среднее время между соседними заявками $T_{з.ср}$, среднее время обслуживания заявки $T_{обс.ср}$, максимально допустимое время ожидания $T_{ож.max}$, период работы системы $T_{кон}$, число случайных реализаций моделируемого процесса $N_р$.

Выходной характеристикой модели является среднее число обслуженных заявок $N_{обс.ср}$.

4.2.1. Выбор показателя и критерия эффективности

В качестве показателя эффективности работы системы целесообразно выбрать среднюю прибыль, определяемую по формуле

$$C_{ср} = C_1 \cdot N_{обс.ср} - C_2(N_к), \qquad (4.1)$$

где C_1 – чистая прибыль, полученная в результате обслуживания одной заявки;

$C_2(N_к)$ – издержки обслуживания всех заявок, зависящие от числа каналов.

Разделим обе части равенства (4.1) на величину C_1. Получим следующее выражение для расчета показателя эффективности:

$$C_{ср.отн} = N_{обс.ср} - \frac{C_2}{C_1}(N_k), \qquad (4.2)$$

где $C_{ср.отн}$ – средняя относительная прибыль.

Величину C_2/C_1 (отношение издержек обслуживания к чистой прибыли, полученной в результате обслуживания одной заявки) будем рассматривать как функцию числа каналов.

Предположим, что возможными вариантами этой функциональной зависимости являются: *а* – линейная зависимость; *б* – возрастающая зависимость с положительной 2-й производной и *в* – возрастающая зависимость с отрицательной 2-й производной (рис. 4.1).

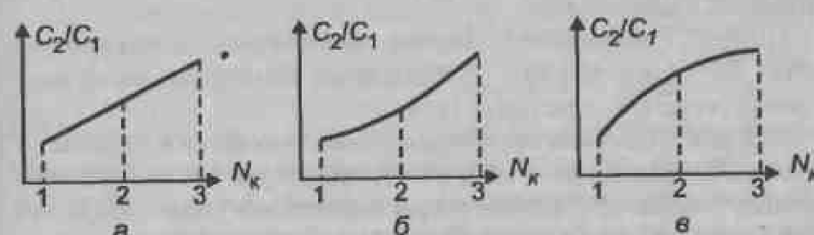


Рис. 4.1. Зависимости отношения C_2/C_1 от N_k :
а – первый вариант; *б* – второй вариант; *в* – третий вариант

Выберем второй вариант. Примем в качестве конкретной зависимости издержек обслуживания от числа каналов следующую функцию:

$$\frac{C_2}{C_1}(N_k)=1-0{,}5\cdot N_k+0{,}5\cdot N_k^2. \qquad (4.3)$$

Итак, для расчета показателя эффективности будем использовать зависимости (4.2) и (4.3).

В качестве критерия выбора наивыгоднейшей структуры СМО примем оптимальное число каналов, обеспечивающее максимум средней относительной прибыли:

$$K(N_к^*) : MAX\;[\;C_{ср.отн}(N_к)\;]\,. \qquad (4.4)$$

$$N_к = 1,\ 2,\ 3,$$

где $N_к^*$ – наивыгоднейшее число каналов.

4.3. СХЕМЫ АЛГОРИТМОВ МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0.

Общий вид (макет) стартовой формы показан на рис. 4.2. Видно, что она включает ряд объектов управления, среди которых имеются три командные кнопки: «Расчет», «Очистка» и «Выход». Это обстоятельство определяет структуру алгоритма процедур обработки прерываний, показанную на рис. 4.3.

После нажатия кнопки «Start» активизируется стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

Цифрой 1 обозначено действие, заключающееся в корректировке исходных данных. Необходимые изменения вносятся в соответствующие текстовые поля.

Цифрой 2 обозначено действие, заключающееся в нажатии (с помощью мыши) кнопки «Расчет». В процедуре, связанной с этой кнопкой, оператор 3 осуществляет перевод исходных данных из символьной формы в числовую. Затем оператор 4 обращается к модулю общего назначения «Model2». Схема алгоритма этого модуля приведена на рис. 4.4.

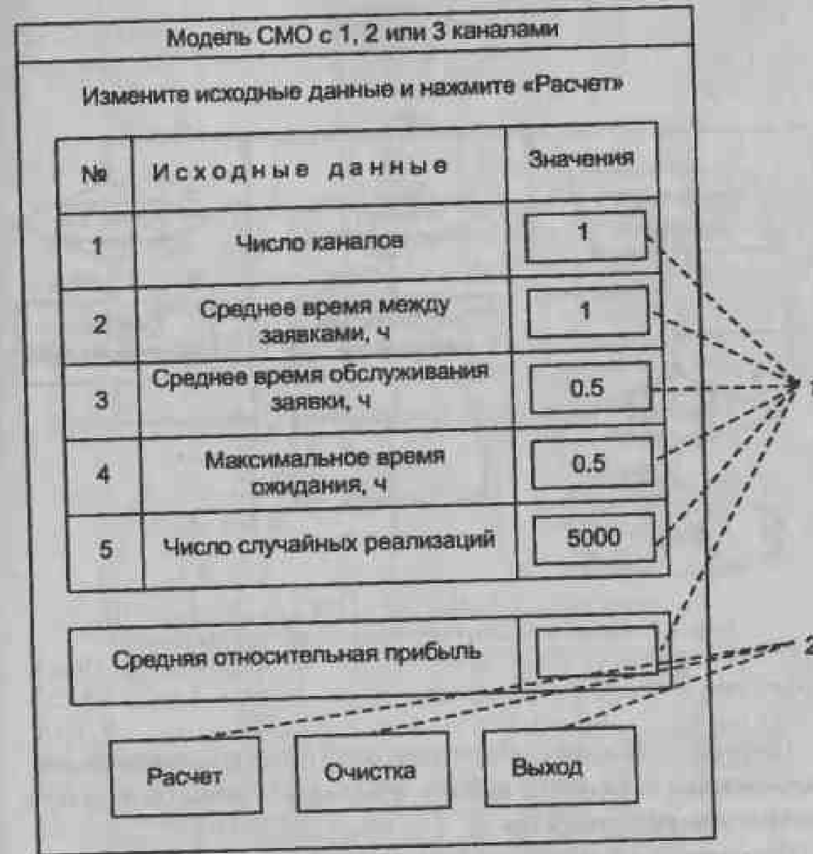


Рис. 4.2. Макет стартовой формы:
1 – текстовые поля; *2* – командные кнопки

После окончания работы модуля и выдачи на экран результатов моделирования работа процедуры, связанной с кнопкой «Расчет», заканчивается. Программа вновь переходит в режим ожидания действий пользователя.

Цифрой 5 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Очистка». В процедуре, связанной с ней, производится очищение текстового поля для вывода результата моделирования. Затем обычно производятся изменение исходных данных и проведение новых расчетов с использованием кнопки «Расчет».

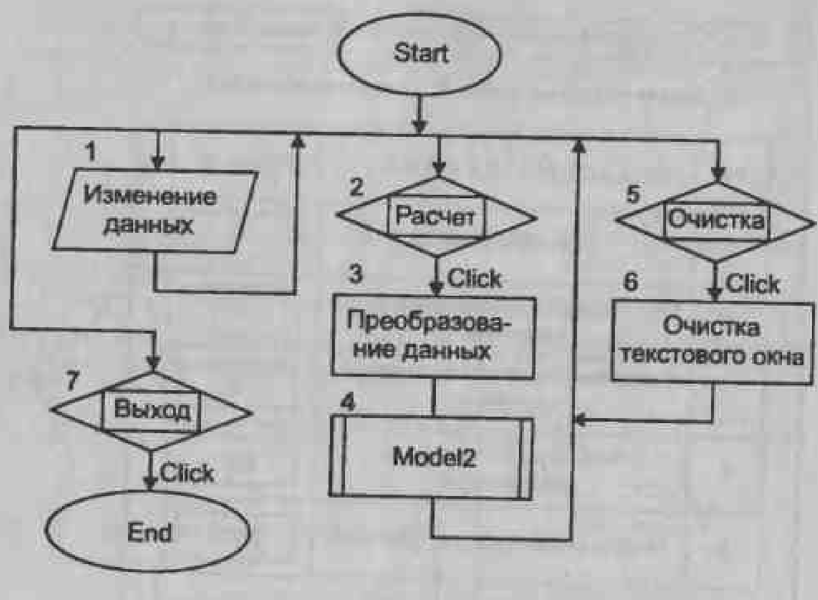


Рис. 4.3. Схема алгоритма процедур обработки прерываний

Цифрой 7 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Выход». В результате работа программы прекращается.

Оператор 1 обнуляет глобальную переменную $SN_{обс}$ – суммарное число обслуженных заявок. Оператор 2 активизирует окно формы № 2 и делает неактивным окно формы № 1. Оператор 3 начинает циклический перебор случайных реализаций. Оператор 4 выводит на экран (в окно формы № 2) счетчик числа рассчитанных реализаций.

Оператор 5 в начале каждой случайной реализации обнуляет локальные переменные, к которым относятся: число заявок, поступающих в одной реализации $N_з$, число обслуженных заявок в каждом из трех каналов $N_{обс.1}$, $N_{обс.2}$, $N_{обс.3}$, начальные значения времени освобождения 1-го, 2-го и 3-го каналов $T_{ок1}$, $T_{ок2}$ и $T_{ок3}$.

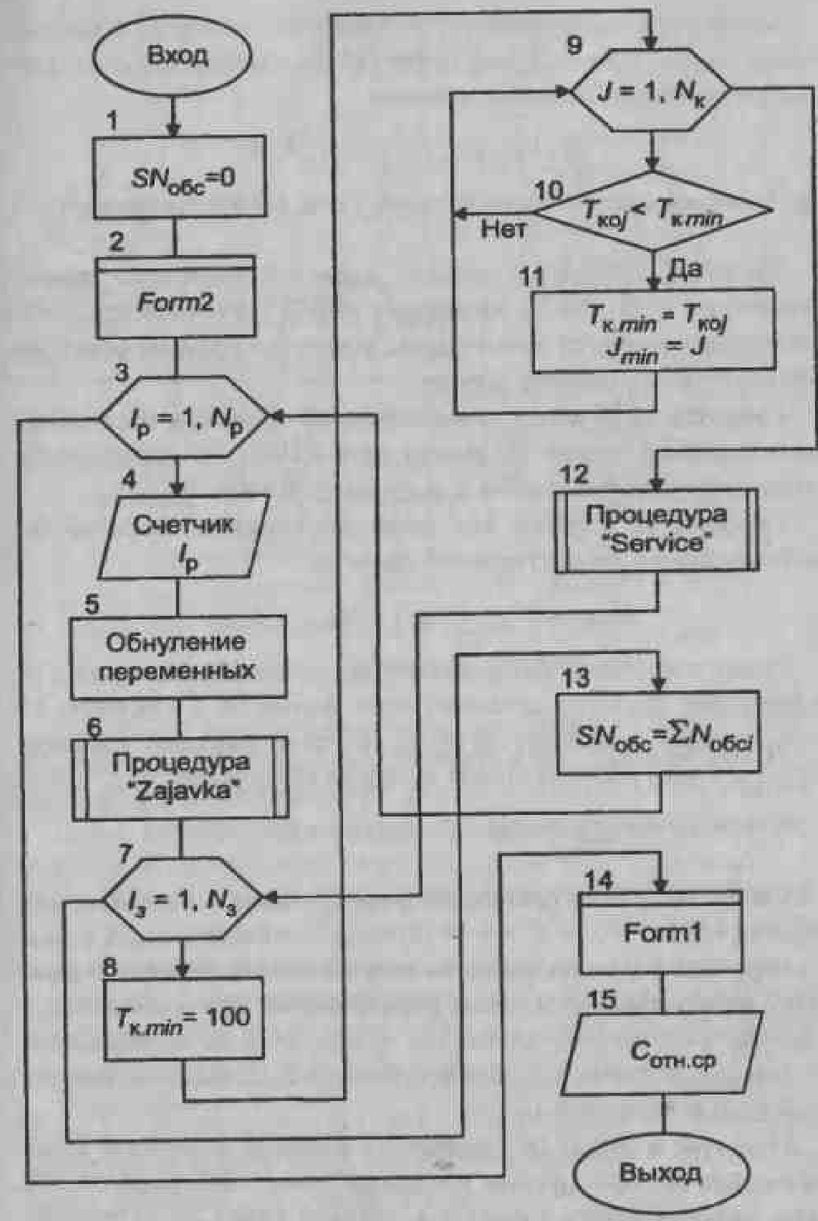


Рис. 4.4. Схема алгоритма модуля «*Model2*»

Оператор 6 обращается к автономной процедуре формирования потока заявок. В результате работы этой процедуры формируется массив значений времени

$$[T_з(1),\ T_з(2),\ T_з(3),\ ...,\ T_з(N_{зi})],$$

где $N_{зi}$ –общее число поступивших заявок для i-й случайной реализации.

Оператор 7 является началом цикла обслуживания заявок. Операторы 8, 9, 10 и 11 производят выбор номера канала, который характеризуется наименьшим значением времени освобождения от обслуживания заявки.

Оператор 12 обращается к автономной процедуре обслуживания очередной заявки. На выходе этой процедуры определяется число обслуженных заявок в выбранном канале $N_{обс}(J_{min})$.

Оператор 13 служит для расчета суммарного числа обслуженных заявок по рекурсивной формуле

$$SN_{обс} = SN_{обс} + N_{обс.1} + N_{обс.2} + N_{обс.2}.$$

После окончания цикла случайных реализаций оператор 14 возвращает свойство активного окна форме № 1. Оператор 15 рассчитывает и выводит на экран значение выходной переменной – средней относительной прибыли по формуле

$$C_{отн.ср} = N_{обс} - 1 + 0{,}5 \cdot N_к - 0{,}5 \cdot N_к^2.$$

Схема алгоритма процедуры формирования заявок показана на рис. 4.5.

Оператор 1 устанавливает на нуль модельное время T. Оператор 2 является началом цикла формирования заявок. Оператор 3 обращается к датчику случайных чисел, который вырабатывает возможное значение случайной величины z, равномерно распределенной в интервале (0,1).

Оператор 4 определяет возможное значение случайной величины времени поступления очередной заявки при условии, что среднее время между соседними заявками равно $T_{з.ср}$. Оператор 5 проверяет условие окончания процесса формирования заявок.

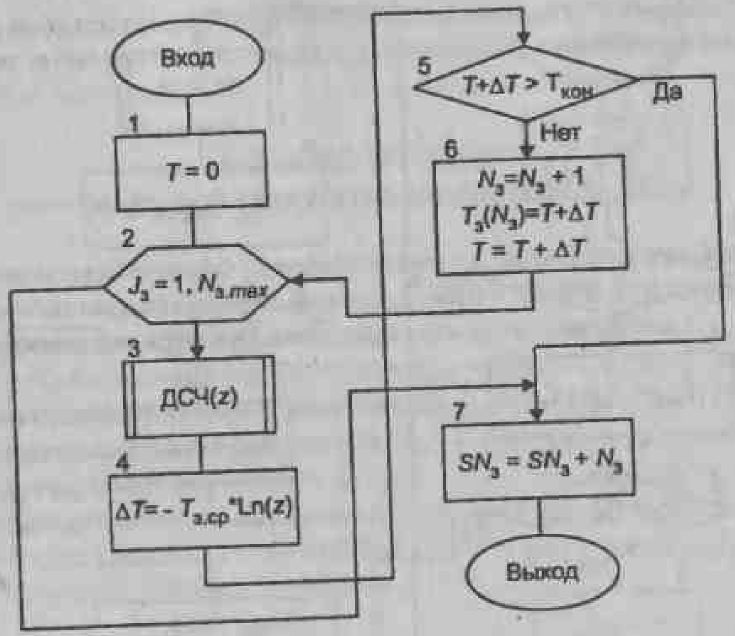


Рис. 4.5. Схема алгоритма формирования заявок

Оператор 6 подсчитывает число поступивших заявок, помещает время поступления каждой заявки в специальный массив и изменяет модельное время T.

Схема алгоритма процедуры обслуживания заявок показана на рис. 4.6.

Оператор 1 обнуляет время ожидания начала обслуживания заявки $T_{ож}$ и присваивает времени начала обслуживания $T_н$ время поступления очередной заявки $T_з(J_з)$.

Оператор 2 производит проверку занятости канала. Начальное значение времени освобождения канала $T_{кoj}$ приравнивается нулю в главном модуле в блоке обнуления локальных переменных.

Если канал занят, то оператор 3 определяет время ожидания $T_{ож}$ как разность времени освобождения канала $T_{кoj}$ и времени поступления заявки $T_з(J_з)$. Оператор 4 проверяет условие, что время ожидания $T_{ож}$ превышает допустимое $T_{ож.max}$. Если это условие выполняется, то управление передается на конец процедуры и заявка остается необслуженной.

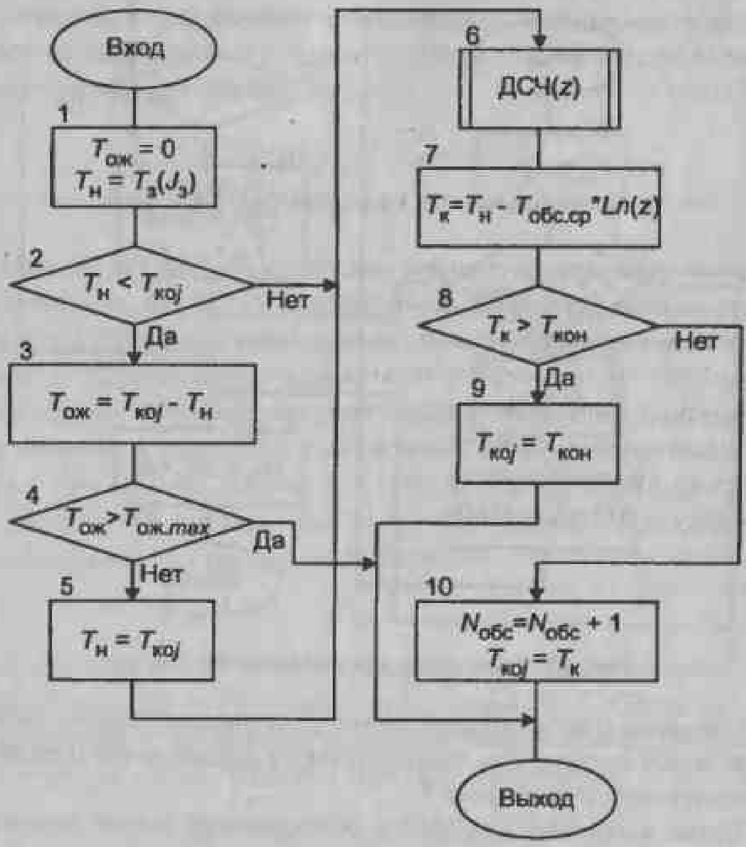


Рис. 4.6. Схема алгоритма процедуры обслуживания заявок

Оператор 5 служит для коррекции времени начала обслуживания заявки. Оно теперь должно равняться времени освобождения канала T_{koj}. Оператор 6 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1), который вырабатывает возможное значение случайной величины z. Оператор 7 определяет возможное значение времени окончания обслуживания заявки $T_к$.

Оператор 8 проверяет условия окончания периода обслуживания, а оператор 9 фиксирует тот факт, что данный канал будет

занят до конца рабочего дня. Оператор 10 увеличивает на единицу число обслуженных заявок в j-м канале и фиксирует время освобождения канала.

4.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Рассмотрим пример решения задачи исследования системы массового обслуживания с помощью разработанной алгоритмической модели. Выберем следующие входные параметры:

- среднее время между заявками $T_{з.ср} = 1$ ч;
- максимальное время ожидания $T_{ож.max} = 0,25$ ч;
- число случайных реализаций $N_р = 5000$.

Варьируемые переменные:

- среднее время обслуживания заявок $T_{обс.ср} = 0,5; 1; 2; 4$ ч;
- число каналов обслуживания $N_к = 1; 2; 3$.

Результаты расчетов приведены в табл. 4.1.

Таблица 4.1

$N_{вар}$	$N_к$	$T_{обс.ср}$	$C_{ооо.ср}$
1	1	0,5	6,27
2	2	0,5	7,12
3	3	0,5	5,53
4	1	1	4,24
5	2	1	5,83
6	3	1	4,74
7	1	2	2,33
8	2	2	3,61
9	3	2	2,94
10	1	4	0,95
11	2	4	1,40
12	3	4	0,65

Следовательно, при выбранных исходных данных, в частности при среднем времени между соседними заявками, равном 1 ч, наибольшая прибыль достигается при числе каналов $N_к = 2$.

Проведенные исследования показали, что оптимальное число каналов зависит от соотношения между величинами среднего времени между соседними заявками и среднего времени обслуживания (рис. 4.7).

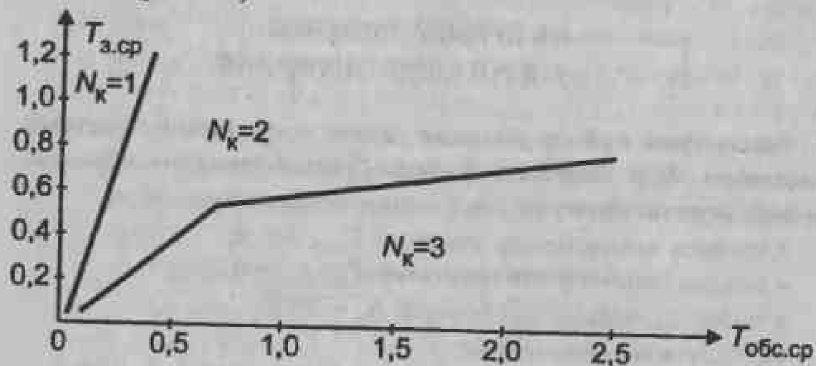


Рис. 4.7. Зависимость оптимального числа каналов $N_к$ от параметров $T_{з.ср}$ и $T_{обс.ср}$

4.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 4.5.1

В интегральной среде Visual Basic 5.0 создайте программу «Модель бензоколонки», текст которой приведен в приложении 2. Произведите отладку программы с целью ликвидации формальных ошибок.

Задание 4.5.2

Произведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 4.5.3

Произведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Выберите исходные данные, проведите расчеты и проанализируйте результаты моделирования.

Задание 4.5.4 (контрольное)

Разработайте алгоритмическую модель процесса функционирования одноканальной системы массового обслуживания с отказами. Поток заявок – однородный, простейший. Время обслуживания заявок имеет заданное распределение. Поток отказов – поток Эрланга 2-го рода. Род отказа задан. Если отказы могут быть 1-го и 2-го рода, то задается вероятность появления отказа 1-го рода. Время ожидания заявки в очереди ограничено. Время работы системы также ограничено. Характеристики вариантов модели приведены в табл. 4.2.

Таблица 4.2

Характеристики вариантов модели

№ варианта	Вид распределения времени обслуживания заявок	Род отказа	№ варианта	Вид распределения времени обслуживания заявок	Род отказа
1	Нормальное	1-й	7	Показательное	2-й
2	Равномерное	1-й	8	Усеченное нормальное	2-й
3	Показательное	1-й	9	Нормальное	1-й и 2-й
4	Усеченное нормальное	1-й	10	Равномерное	1-й и 2-й
5	Нормальное	2-й	11	Показательное	1-й и 2-й
6	Равномерное	2-й	12	Усеченное нормальное	1-й и 2-й

5. МОДЕЛЬ УПРАВЛЕНИЯ ЗАПАСАМИ

5.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель: – Я хотел бы вложить средства в строительство склада для продажи автомашин. У меня уже есть контора, где будут оформляться документы на покупку, а затем покупатель будет отправляться на склад и получать автомашину. Я бы хотел разработать модель процесса работы склада.

Консультант: – На какие вопросы Вы хотите получить ответы с помощью модели?

Предприниматель: – Нужно установить, каким должно быть оптимальное количество автомашин, хранящихся на складе. Количество покупателей, приобретающих товар каждый день, заранее неизвестно. Его можно только ориентировочно оценить. Поэтому неопределенным является и минимальный запас, который нужно поддерживать на складе ежедневно. С одной стороны, с увеличением запаса товара растут эксплуатационные расходы. С другой стороны, при уменьшении запаса товара может возникнуть ситуация, при которой некоторые покупатели, оформившие покупку, не смогут получить товар к концу рабочего дня. По правилам, действующим на нашей фирме, в этом случае нужно выплачивать покупателям неустойку в довольно крупных размерах. В то же время организация подвоза новых партий товара требует нескольких дней и сопряжена с дополнительными затратами.

Консультант: – Зависят ли затраты на поставку дополнительной партии товара от ее объема?

Предприниматель: – Да, конечно. Стоимость затрат на хранение товара пропорциональна количеству автомашин, находящихся на складе в данный день. Стоимость затрат на поставку товара пропорциональна объему партии. Стоимость затрат, вызванных дефицитом товара, пропорциональна количеству покупателей, спрос которых остался неудовлетворенным.

Консультант: – Все ясно. С точки зрения разработчика моделей задача заключается в создании алгоритмической модели процесса функционирования сложной системы с целью оптимизации ее параметров. В данном случае условием оптимальности будет условие минимизации затрат на содержание склада в течение определенного периода времени.

Предприниматель: – А к какому классу будет относиться такая модель? Будет ли она моделью системы массового обслуживания?

Консультант: – Процесс, который Вы описали, не содержит характерных признаков системы массового обслуживания. Здесь нет потока заявок, нет явного описания процесса обслуживания,

отсутствуют и правила обслуживания. Однако это, несомненно, должна быть алгоритмическая статистическая модель, так как среди входных переменных имеются случайные переменные. Одна из них – случайное число продаваемых ежедневно автомашин (ежедневный расход товара). Вторая – случайное число дней, затраченных на организацию поставки дополнительной партии товара. Для этих величин нужно подобрать подходящие по физическому смыслу стандартные распределения. Все остальные входные переменные могут задаваться как постоянные величины или варьироваться для различных вариантов расчета.

Предприниматель: – Для величины ежедневного спроса я могу указать только возможный диапазон ее изменения. Речь может идти о сотнях покупателей в день с некоторым отклонением в ту и другую сторону.

Консультант: – Это уже кое-что. Поскольку количество покупателей – это дискретная случайная величина, здесь больше всего подошло бы так называемое биномиальное распределение, которое при большом числе возможных значений (что мы и имеем в данном случае) может быть успешно заменено нормальным распределением.

Его характеризуют двумя параметрами: математическим ожиданием и средним квадратическим отклонением числа покупателей. Единственное дополнительное условие заключается в том, что полученное возможное значение случайной величины нужно округлить до целой величины.

Предприниматель: – Очевидно, так же можно подойти и к выбору вида распределения для случайной величины времени поставки партии товара?

Консультант: – Да, конечно. Здесь также можно использовать вместо биномиального распределения дискретной случайной величины нормальное распределение непрерывной случайной величины. Однако в данном случае диапазон возможных значений числа дней на поставку товара невелик, и поэтому такая аппроксимация является менее точной, чем для количества покупателей. Характеристиками распределения являются математическое ожидание и среднее квадратическое отклонение времени поставки партии товара. Естественно, что и в этом случае возможное значение случайной величины должно быть округлено до целого числа.

Предприниматель: – Так какие же характеристики нужно знать, чтобы можно было разработать модель?

Консультант: – Давайте перечислим. Ко входным переменным следует отнести следующие величины:

средний ежедневный спрос и среднее квадратическое отклонение ежедневного спроса;

среднее время поставки дополнительной партии товара и его среднее квадратическое отклонение;

стоимость хранения единицы товара на складе в течение одних суток;

стоимость затрат на доставку единицы товара;

затраты, связанные с дефицитом каждой единицы товара;

начальный уровень запаса товара;

период работы склада.

Итого получилось семь входных переменных. К ним нужно добавить две варьируемых переменных:

объем дополнительной партии товара;

критический уровень запаса товара, при достижении которого администратор склада должен заказывать новую партию.

Предприниматель: – А что будет определяться на выходе модели?

Консультант: – В качестве выходной величины можно использовать средние затраты на содержание склада за заданный период времени.

Предприниматель: – Средняя величина затрат для меня не является полной характеристикой работы склада. Мне важно знать, каков может быть верхний предел затрат. Это необходимо для того, чтобы избежать банкротства фирмы.

Консультант: – Действительно, среднее значение случайной величины не является ее полной характеристикой. Нужно еще учесть разброс этой величины. Поэтому в качестве показателя эффективности моделируемого процесса целесообразно выбрать максимальные гарантированные затраты с заданным уровнем гарантии.

Предприниматель: – А что характеризует такой показатель?

Консультант: – Он представляет собой максимальный уровень затрат, которые с заданной надежностью не могут быть превышены при длительном использовании склада. В какие-то периоды склад может работать с относительно малыми затратами, а в какие-то с большими затратами. Но практически никогда затраты не превзойдут величины выбранного показателя эффективности.

Предприниматель: – Ну что же, такой показатель меня устраивает. Только мне пока не ясно, каков должен быть уровень надежности?

Консультант: – Это зависит от того, какова Ваша «склонность к риску». Если Вы не хотите рисковать, то должны ориентироваться на высокий уровень надежности, порядка 0,99. Это означает, что только в одном из ста периодов функционирования склада затраты превысят величину показателя эффективности. Если Вы согласны на определенный риск, то можно выбрать уровень надежности 0,9. Тогда уже в десяти случаях из ста затраты превысят величину показателя эффективности.

Предприниматель: – Но кроме показателя существует еще критерий эффективности. Что он будет представлять собой в нашей модели?

Консультант: – Вопрос о выборе критерия эффективности возникает только тогда, когда имеется несколько вариантов или стратегий использования системы. В данном случае под стратегией мы можем подразумевать вариант сочетаний значений варьируемых переменных: объема партии товара и критического уровня товара, при достижении которого необходимо заказывать дополнительную партию. Критерием может быть условие минимума показателя эффективности при изменении варьируемых переменных.

5.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Имеется склад, где хранится запас однотипного товара. Начальный уровень запаса равен $N_{ур}$. В результате удовлетворения ежедневного спроса происходит уменьшение запаса на случайную величину, имеющую нормальное распределение с известными характеристиками: математическим ожиданием $M_д$ и средним квадратическим отклонением $\sigma_д$. Затраты на хранение единицы товара в течение суток составляют $\$C_1$.

Если на складе не останется товара, то администрации склада придется уплатить прибывшим покупателям компенсацию в размере $\$C_3$ за каждую недостающую единицу товара по отношению к заказанному количеству.

Чтобы избежать затрат, вызванных дефицитом товара, администрация склада периодически подает заявки на поставку дополнительных партий товара выбранного объема *Part*. Подача заявки производится в момент, когда уровень запаса товара снизится до выбранного уровня UR_{min}. Время выполнения заявки является случайной величиной с нормальным распределением и известными характеристиками M_T и σ_T. Затраты на поставку единицы товара составляют $\$C_2$. Цикл работы склада составляет T_D дней.

Требуется разработать математическую модель, которая бы описывала возможный реальный процесс, позволяла изучать его закономерности и выбирать наивыгоднейший режим работы склада, обеспечивающий минимальные затраты на его содержание.

В частности, требуется установить:

оптимальный уровень запаса UR_{min}, при достижении которого необходимо подавать заявку на поставку партии товара;

оптимальный объем партии товара *Part*.

В качестве ***показателя эффективности*** моделируемого процесса целесообразно выбрать максимальные гарантированные затраты на содержание склада (с заданным уровнем гарантии), определяемые по формуле

$$C_{gar} = M_C + K_\alpha \cdot \sigma_C,$$

где M_C – средние затраты на содержание склада в течение периода T_D дней;
σ_C – среднее квадратическое отклонение затрат на содержание склада;
K_α – квантиль, зависящий от уровня гарантии α (K_α=1,28 при α=0,9).

В качестве ***критерия выбора оптимального режима работы склада*** может быть принят ***минимум максимальных гарантированных затрат:***

$$K(UR^*_{min}, Part^*) : MIN\{ C_{gar}(UR_{min}, Part) \},$$

где UR^*_{min} – оптимальное значение уровня запаса товара, при достижении которого необходимо подавать заявку на поставку новой партии;
$Part^*$ – оптимальный объем партии товара.

5.3. АЛГОРИТМ МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0.

Общий вид (макет) стартовой формы показан на рис. 5.1. Она включает следующие объекты управления:

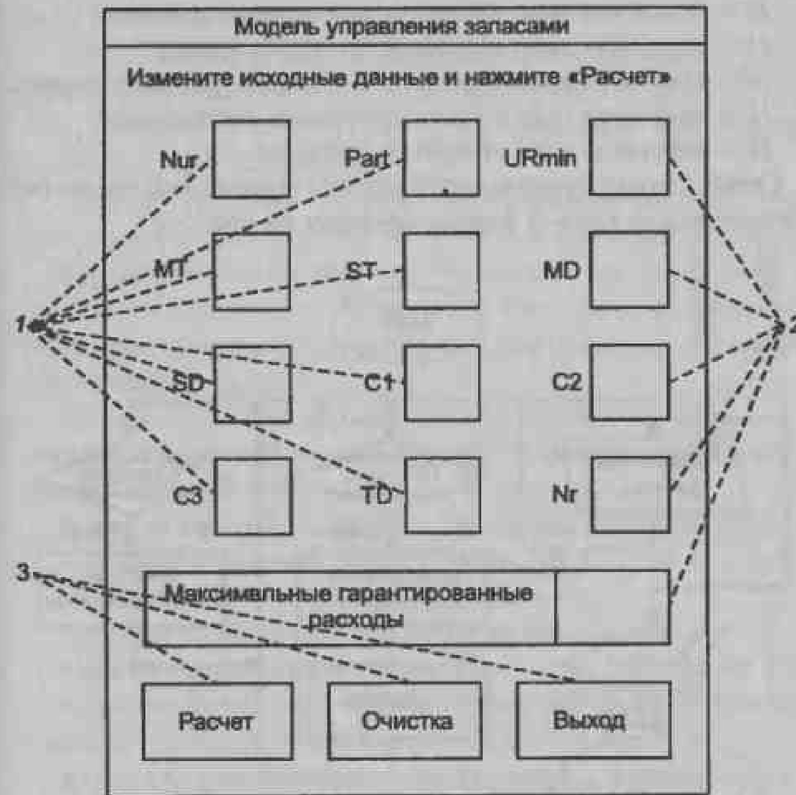


Рис. 5.1. Макет стартовой формы:
1, 2 – текстовые поля; *3* – командные кнопки

1 – текстовое поле «Начальный уровень товара»;
2 – текстовое поле «Объем партии товара»;
3 – текстовое поле «Среднее время выполнения заказа»;
4 – текстовое поле «СКО времени выполнения заказа»;
5 – текстовое поле «СКО дневного расхода»;
6 – текстовое поле «Стоимость хранения единицы товара»;
7 – текстовое поле «Штраф за отсутствие единицы товара»;
8 – текстовое поле «Период работы склада»;
9 – командную кнопку «Расчет»;
10 – командную кнопку «Очистка»;
11 – командную кнопку «Выход»;
12 – текстовое поле «Критический уровень товара»;
13 – текстовое поле «Средний дневной расход»;
14 – текстовое поле «Стоимость доставки единицы товара»;
15 – текстовое поле «Число случайных реализаций»;
16 – текстовое поле «Результат расчета».

Схема алгоритма процедур обработки прерываний, связанных с объектами стартовой формы, показана на рис. 5.2.

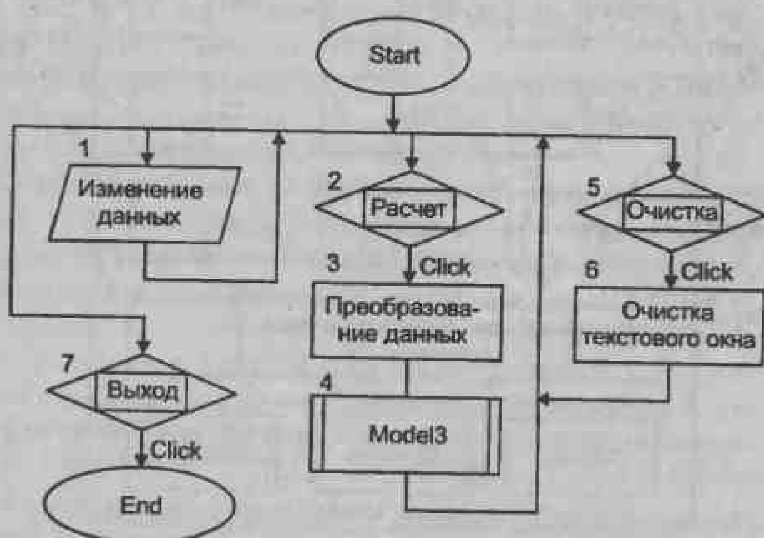


Рис. 5.2. Схема алгоритма процедур обработки прерываний

После нажатия кнопки «Start» активизируется стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

Цифрой 1 обозначено действие, заключающееся в корректировке исходных данных. Необходимые изменения вносятся в соответствующие текстовые поля.

Цифрой 2 обозначено действие, заключающееся в нажатии (с помощью мыши) кнопки «Расчет». В процедуре, связанной с этой кнопкой, оператор 3 осуществляет перевод исходных данных из символьной формы в числовую. Затем оператор 4 обращается к модулю общего назначения «Model3». Схема алгоритма этого модуля приведена на рис. 5.3.

После окончания работы модуля и выдачи на экран результатов моделирования работа процедуры, связанной с кнопкой «Расчет», заканчивается. Программа вновь переходит в режим ожидания действий пользователя.

Цифрой *5* на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Очистка». В процедуре, связанной с ней, производится очищение текстового поля для вывода результата моделирования. Затем может быть произведено изменение исходных данных и проведены новые расчеты с использованием кнопки «Расчет».

Цифрой *7* на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Выход». В результате работа программы прекращается.

В состав базового комплекта исходных данных входят:

- начальный уровень запаса товара – $N_{ур}$;
- средний ежедневный спрос товара – $M_д$;
- среднее квадратическое отклонение ежедневного спроса – $\sigma_д$;
- объем партии товара – *Part*;
- уровень запаса, при котором должна оформляться заявка на поставку дополнительной партии товара, – UR_{min};
- среднее время поставки партии товара – M_T дней;
- среднее квадратическое отклонение времени поставки товара – σ_T дней;

• стоимость хранения единицы товара в течение суток – $\$C_1$;

• стоимость поставки единицы товара – $\$C_2$;

• стоимость затрат на компенсацию отсутствия товара на складе – $\$C_3$ за каждую недостающую единицу;

• период функционирования склада – T_D дней;

• число случайных реализаций – N_p.

Блок 1 обеспечивает обнуление глобальных переменных, к которым относятся:

MS_C – начальное значение суммы случайных величин общих затрат на содержание склада для различных реализаций моделируемого процесса;

SS_C – начальное значение суммы квадратов случайных величин общих затрат.

Кроме того, в этом же блоке устанавливается начальное значение времени поставки дополнительной партии товара, которое выходит за пределы периода работы склада и равняется

$$T_{\text{пост.0}} = T_D + 1.$$

Оператор 2 представляет собой заголовок цикла случайных реализаций.

Блок 3 производит обнуление локальных переменных, к которым относятся:

S_{C1} – начальное значение суммарных затрат на хранение товара в течение всего периода работы склада для одной реализации;

S_{C2} – начальное значение суммарных затрат, связанных с нехваткой товара в течение всего периода работы склада для одной реализации;

S_{C3} – начальное значение суммарных затрат на поставку дополнительных партий товара в течение всего периода работы склада для одной реализации;

Т – исходное значение счетчика дней (модельное время);

Zajav – числовой признак отсутствия заявки на поставку дополнительной партии товара (если заявка подана, то *Zajav* = 1).

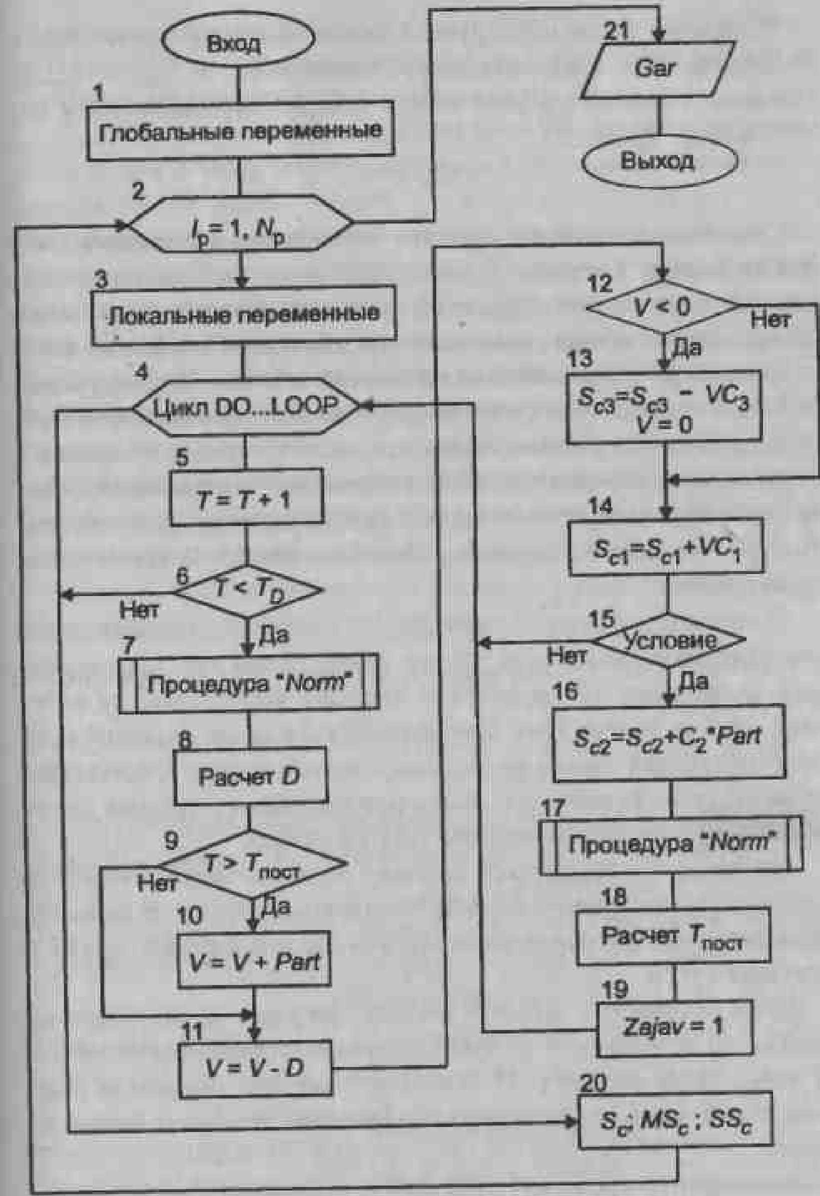


Рис. 5.3. Схема алгоритма модуля «*Model3*»

В этом же блоке планируемое время поставки партии товара приравнивается к его начальному значению, т. е. $T_{пост} = T_{пост0}$. Наконец, текущий уровень запаса товара приравнивается к его начальному уровню:

$$V = N_{ур}.$$

Оператор 4 является началом циклического перебора дней работы склада. Оператор 5 увеличивает модельное время на один шаг (на одни сутки). Условный оператор 6 проверяет условие продолжения работы склада для каждой случайной реализации.

Оператор 7 обращается к процедуре «*Norm*», формирующей возможное значение нормированной, центрированной случайной величины с нормальным распределением. Оператор 8 определяет возможное значение величины случайного ежедневного спроса при заданном его математическом ожидании $M_д$ и среднем квадратическом отклонении $\sigma_д$. Результат расчета округляется до целого числа.

Условный оператор 9 проверяет условие наступления срока поставки дополнительной партии товара. Если этот срок наступил, то оператор 10 увеличивает текущий запас товара на величину объема партии *Part*. Одновременно здесь же числовой признак отсутствия заявки на поставку партии товара устанавливается на нуль. Кроме того, планируемое время поставки вновь выводится за пределы периода работы склада.

Оператор 11 определяет текущий уровень запаса товара на складе с учетом удовлетворения ежедневного спроса. В дальнейшем оператор 14 определяет затраты на содержание склада в текущие сутки.

Если количество заявок превысит текущий запас товара на складе, то в операторе 11 будет получено отрицательное число. В этом случае оператор 13 определяет затраты, связанные с дефицитом товара, и устанавливает значение текущего запаса на нуль.

Оператор 15 проверяет выполнение одновременно двух условий: превышает ли текущий запас минимально допустимый (критический) уровень и была ли уже оформлена заявка на поставку

дополнительной партии товара. Если хотя бы одно из этих условий не выполняется, то в алгоритме происходит передача управления на начало цикла, т. е. оператору 4.

Если же одновременно выполняются оба условия, то группа операторов с 16-го по 19-й определяет затраты на поставку дополнительной партии товара.

Оператор 16 увеличивает текущие затраты на величину стоимости партии товара. Оператор 17 обращается к процедуре формирования возможного значения нормированной, центрированной случайной величины с нормальным распределением. Оператор 18 определяет возможное значение случайной величины времени выполнения заявки на поставку партии товара с учетом заданных параметров: среднего времени поставки и среднего квадратического отклонения времени поставки. При этом результат расчета округляется до целого числа. Оператор 19 устанавливает на единицу числовой признак подачи заявки.

Блок 20 служит для расчета суммарных характеристик затрат на содержание склада по формулам:

$$S_C = S_{C1} + S_{C2} + S_{C3};$$

$$MS_C = MS_C + S_C;$$

$$SS_C = SS_C + S_C \cdot S_C.$$

Блок 21 служит для определения показателя эффективности по формулам:

$$M_C = MS_C / N_p;$$

$$\sigma_C = \sqrt{\frac{1}{N_p - 1}(SS_C - N_p \cdot M_C^2)};$$

$$Gar = M_C + 1{,}28 \cdot \sigma_C.$$

5.4. ГРАФИК ИЗМЕНЕНИЯ УРОВНЯ ЗАПАСА ТОВАРА

Типовой график изменения уровня запаса товара на складе с учетом дополнительных поставок показан на рис. 5.4.

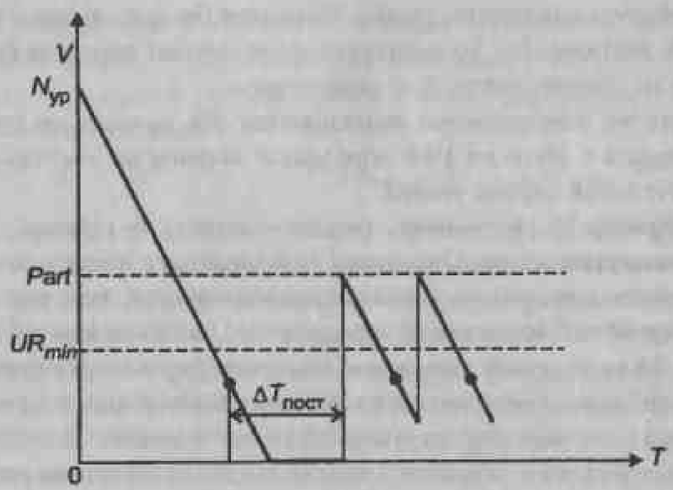


Рис. 5.4. График изменения запаса товара на складе

В начальный момент на складе имеется $N_{ур}$ единиц товара. В этот момент числовой признак подачи заявки на поставку дополнительного товара $Zajav$=0, а время поставки превышает время окончания работы склада.

В результате удовлетворения случайного ежедневного спроса уровень запаса товара снижается. Если уровень запаса товара снизится ниже величины UR_{min}, то подается заявка на поставку дополнительной партии товара объемом ***Part*** единиц. В этот момент числовой признак $Zajav$ становится равным единице, а время выполнения поставки определяется по формуле

$$T_{пост} = T + \Delta T_{пост} = T + M_T + n \cdot \sigma_T ,$$

где T – текущее модельное время (в днях);
$\Delta T_{пост}$ – период выполнения заказа;
M_T – среднее время выполнения заказа;
σ_T – среднее квадратическое отклонение времени выполнения заказа;
n – возможное значение нормированной, центрированной случайной величины с нормальным распределением.

Когда модельное время становится равным $T_{пост}$, поступает партия товара, и текущий уровень запаса товара скачком увеличивается на величину $Part$. В этот момент числовой признак $Zajav$ становится равным нулю, а время поступления партии товара вновь приравнивается величине $T_{пост.0}$.

5.5. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Примем следующие исходные данные:

- начальный уровень запаса товара $N_{ур}$ = 1000;
- средний ежедневный спрос товара $M_д$ = 100;
- среднее квадратическое отклонение ежедневного спроса $\sigma_д$ = 10;
- среднее время поставки партии товара M_T = 3 дня;
- среднее квадратическое отклонение времени поставки товара σ_T = 1день;
- стоимость хранения единицы товара в течение суток C_1 = 2$;
- стоимость поставки единицы товара C_2 = 1$;
- затраты на компенсацию отсутствия единицы товара на складе C_3 = 10$;
- период функционирования склада T_D = 30 дней;
- число случайных реализаций N_P = 2000.

Варьируемые переменные:

- объем партии товара $Part$ = 100; 200; 300; 400; 500;
- уровень запаса, при котором должна оформляться заявка на поставку дополнительной партии товара UR_{min} = 50; 150; 250; 350; 450; 550.

Результаты моделирования представлены в табл. 5.1.

Таблица 5.1

Зависимость максимальных гарантированных затрат от варьируемых переменных, тыс. $

Part	UR_{min}					
	50	150	250	350	450	550
100	234306	23438	22959	23020	23111	23111
200	21973	19751	19617	20459	22262	25382
300	21781	20200	21132	24282	29537	35024
400	25512	21414	13395	27222	32458	37665
500	23276	23120	25315	30165	35188	40188

Как видно из табл. 5.1, оптимальными параметрами исследуемой системы, при которых максимальные гарантированные затраты на содержание склада минимальны, являются:

объем партии товара $Part = 200$;

критический уровень запаса товара $UR_{min} = 250$.

5.6. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 5.6.1

В среде Visual Basic 5.0 создайте исполняемый модуль программы «Модель управления запасами», текст которой приведен в приложении 3. Произведите отладку программы с целью ликвидации формальных ошибок.

Задание 5.6.2

Произведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 5.6.3

Произведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Выберите исходные данные, проведите расчеты и проанализируйте результаты моделирования.

6. МОДЕЛЬ ПРОИЗВОДСТВЕННОЙ ФИРМЫ

6.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель:– Мне вновь нужна Ваша консультация. На этот раз я хочу вложить средства в создание производственной фирмы, которая будет производить изделия по заказам потребителей. Количество заказов, конечно, заранее неизвестно. Можно только предположительно указать среднее число заказов в единицу времени (например, в день).

Консультант: – В таком случае следует принять допущение о том, что поток отказов является простейшим, а это означает, что время между соседними заказами представляет собой случайную величину с показательным распределением.

Предприниматель: – Я предполагаю, что в технологическом процессе производства будет участвовать несколько цехов. Каждый из них будет выполнять определенную часть операций по изготовлению изделия, а из последнего цеха будет выпускаться готовое изделие. Сколько будет цехов и какую часть времени изделие будут находиться в каждом из них, я пока не знаю. Желательно, чтобы структура предприятия была оптимальной.

Консультант: – В каком смысле «оптимальной»?

Предприниматель: – В смысле обеспечения наибольшей прибыли. Пока я могу лишь указать среднее время на изготовление одного изделия. Оно складывается из среднего времени работы с изделием в различных цехах. Можно также ориентировочно указать величину максимального разброса времени изготовления изделия в процентах от его среднего значения.

Консультант: – В этих условиях целесообразно принять допущение о том, что время изготовления изделия в каждом цехе является случайной величиной с нормальным распределением. Среднее время работы с изделием для каждого цеха должно быть задано. Разброс времени изготовления изделий можно характеризовать относительным значением среднего квадратического отклонения времени работы с изделием, постоянным для всех цехов.

Предприниматель: – В качестве показателя эффективности я предполагаю использовать величину средней прибыли. Ее можно определить как произведение среднего числа изготовленных изделий на цену одного изделия за вычетом общих затрат на организацию производства.

Консультант: – Можно ориентироваться на среднюю прибыль, но предпочтительнее выбрать минимальную гарантированную прибыль с заданным уровнем гарантии, которая учитывает разброс возможных величин прибыли относительно ее среднего значения.

Предприниматель: – Все же наиболее неопределенной мне представляется структура будущего предприятия. Существует множество вариантов распределения ресурсов по цехам. Можно ли сформировать такой числовой параметр, который отражал бы изменения в структуре предприятия?

Консультант: – Нужно выбрать такую переменную, которая была бы чувствительна к изменению структуры производства и представляла бы собой непрерывную величину, меняющуюся в некотором диапазоне (например, в интервале (0,1)). Для этого придется сделать дополнительные допущения.

Прежде всего будем считать суммарное среднее время работы всех цехов по изготовлению одного изделия величиной постоянной и заданной. Для каждого варианта структуры предприятия будем считать, что средние времена работы в цехах образуют убывающий ряд чисел. Тогда среднее время работы в первом цехе будет максимальным, а в последнем цехе – минимальным. Если разделить разность между максимальным и минимальным временем на общее среднее время работы с изделием, то получим безразмерный параметр, принимающий значения в интервале (0,1). Этот параметр можно назвать числовым фактором.

Предприниматель: – А что физически означает равенство числового фактора нулю?

Консультант: – Это означает, что время работы с изделием во всех цехах одинаково. Это реализация конвейерного способа производства.

Предприниматель: – А что означает равенство числового фактора единице?

Консультант: – Это означает, что среднее время работы в первом цехе (максимальное среднее время работы) численно равно общему среднему времени работы предприятия по изготовлению одного изделия. Это случай, когда весь производственный процесс сосредоточен в одном цехе.

Все остальные возможные варианты являются промежуточными, им соответствуют значения числового фактора *Fact*, удовлетворяющие неравенству: $0 < Fact < 1$.

Предприниматель: – Так сколько же входных переменных будет иметь модель производственной фирмы?

Консультант: – Это будет зависеть от числа цехов. Если ограничить число цехов четырьмя, то можно насчитать 7 входных переменных:

- интенсивность потока заказов;
- среднее время работы с изделием в каждом цехе (для четырех цехов);
- относительная величина среднего квадратического отклонения времени работы с изделием в любом цехе;
- цена одного изделия;
- общая стоимость затрат на организацию производства;
- продолжительность периода функционирования производственной фирмы;
- число случайных реализаций моделируемого процесса.

Предприниматель: – А где же числовой фактор?

Консультант: – Он не является независимой входной переменной. Числовой фактор определяется после выбора значений среднего времени работы с изделием в каждом цехе. Он является в модели выходной переменной наряду со средним числом изготовленных изделий, средней прибылью и средним квадратическим отклонением прибыли.

Предприниматель: – Ну, а к какому классу можно отнести создаваемую модель?

Консультант: – Это алгоритмическая модель одноканальной, многофазной системы массового обслуживания с однородными заявками и неограниченным ожиданием.

6.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Имеется производственная фирма, которая осуществляет выпуск сложных изделий по заказам потребителей. Технологический процесс предусматривает изготовление отдельных узлов и агрегатов с помощью различного оборудования, которое может размещаться в нескольких цехах. Количество цехов $N_ц$ – величина переменная, изменяющаяся от 1 до 4.

Поток заказов – простейший (пуассоновский) с известным средним числом заказов в день – L_z.

Случайная величина времени изготовления одного изделия складывается из случайных величин времени изготовления агрегатов в каждом цехе, имеющих нормальное распределение. Среднее значение времени работы (времени «обслуживания» изделия) в j-м цехе $T_{об\,j}$ для каждого варианта структуры предприятия считается заданным. Суммарное среднее время изготовления изделия T_S является постоянным. Разброс времени работы в каждом цехе характеризуется относительным средним квадратическим отклонением $\sigma_{об}^{отн}$, постоянным для всех цехов.

На выпуск продукции оказывает влияние структура предприятия, т. е. количество цехов и распределение ресурсов по цехам. Число возможных вариантов структуры бесконечно. Поэтому для исследования влияния структуры фирмы на эффективность производственного процесса необходимо выбрать параметр или числовой фактор, который свел бы количество расчетных вариантов к обозримому числу.

В качестве такого числового фактора предлагается ввести следующее выражение:

$$Fact = \{MAX(T_{об\,j}) - MIN(T_{об\,j})\} \;/\; T_S \quad (j\text{=}1,\ 2,\ 3,\ 4),$$

где $MAX(T_{об\,j})$ – максимальное среднее время работы для различных цехов;
$MIN(T_{об\,j})$ – минимальное среднее время работы для различных цехов;
T_S – суммарное среднее время изготовления изделия.

Чистая прибыль, получаемая в результате изготовления одного изделия, равна \$$P$. Дополнительные затраты фирмы на обеспечение производства за период T_D дней равны \$$C$. Поэтому случайная прибыль определяется по формуле

$$Prof = N_{изд} \cdot P - C,$$

где $N_{изд}$ – количество выпущенных изделий для одной случайной реализации моделируемого процесса.

В качестве показателя эффективности исследуемой системы принята минимальная гарантированная прибыль, определяемая по формуле

$$G_{prof} = M_{prof} - K_\alpha \cdot \sigma_{prof},$$

где M_{prof} – математическое ожидание прибыли;
σ_{prof} – среднее квадратическое отклонение прибыли;
K_α – квантиль нормального распределения, соответствующий заданному уровню надежности α (K_α=1,28 при α=0,9).

Управляющими (варьируемыми) параметрами модели являются:

1) массив среднего времени «обслуживания» изделий в цехах, используемый для расчета значения числового фактора;

2) относительная величина разброса времени работы в каждом цехе $\sigma_{об}^{отн}$.

В качестве критерия эффективности принят ***максимум минимальной гарантированной прибыли***

$$K(Fact^*) : MAX\ \{G_{prof}(Fact)\},$$

где $Fact^*$ – оптимальное значение числового фактора.

6.3. СХЕМЫ АЛГОРИТМА МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0.

Общий вид (макет) стартовой формы показан на рис. 6.1. Она включает следующие объекты управления: текстовые поля для изменения исходных данных, вывода значения числового фактора и показателя эффективности, командные кнопки «Расчет», «Очистка» и «Выход».

Укрупненная схема алгоритма модели показана на рис. 6.2.

После нажатия на панели инструментов кнопки «Start» активизируется стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

Цифрой 1 обозначено действие, заключающееся в корректировке исходных данных. Необходимые изменения вносятся в соответствующие текстовые поля.

Цифрой 2 обозначено действие, заключающееся в нажатии (с помощью мыши) кнопки «Расчет». В процедуре, связанной с этой кнопкой, оператор 3 осуществляет перевод исходных данных из

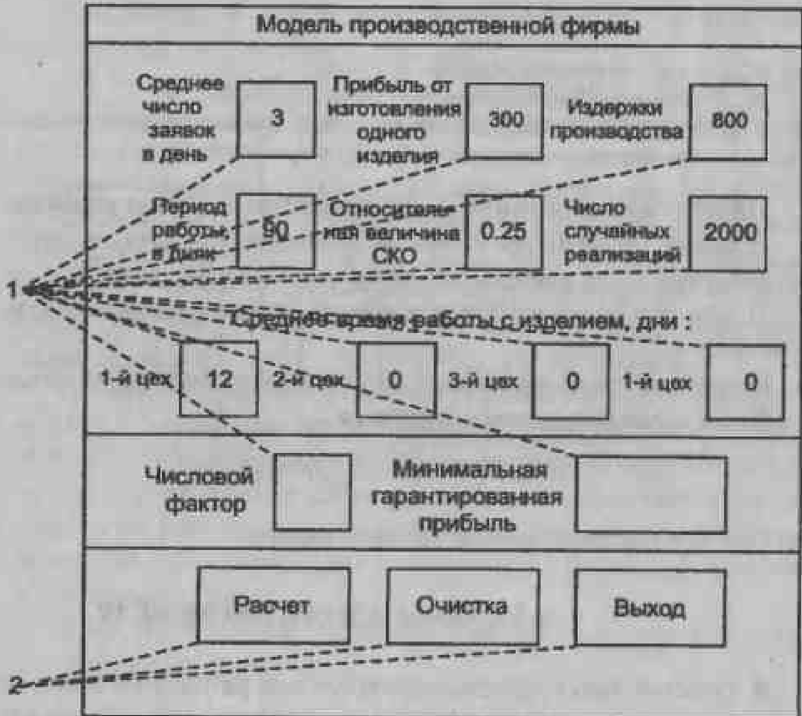


Рис. 6.1. Макет стартовой формы:
1 –текстовые поля; *2* – командные кнопки

символьной формы в числовую. Затем оператор 4 обращается к модулю общего назначения «Model4». Схема алгоритма этого модуля приведена на рис. 6.3. После окончания работы модуля и выдачи на экран результатов моделирования работа процедуры, связанной с кнопкой «Расчет», заканчивается. Программа вновь переходит в режим ожидания действий пользователя.

Цифрой 5 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Очистка». В процедуре, связанной с ней, производится очищение текстовых полей для вывода показателя эффективности и значения числового фактора. Затем может быть произведено изменение исходных данных и проведены новые расчеты с использованием кнопки «Расчет».

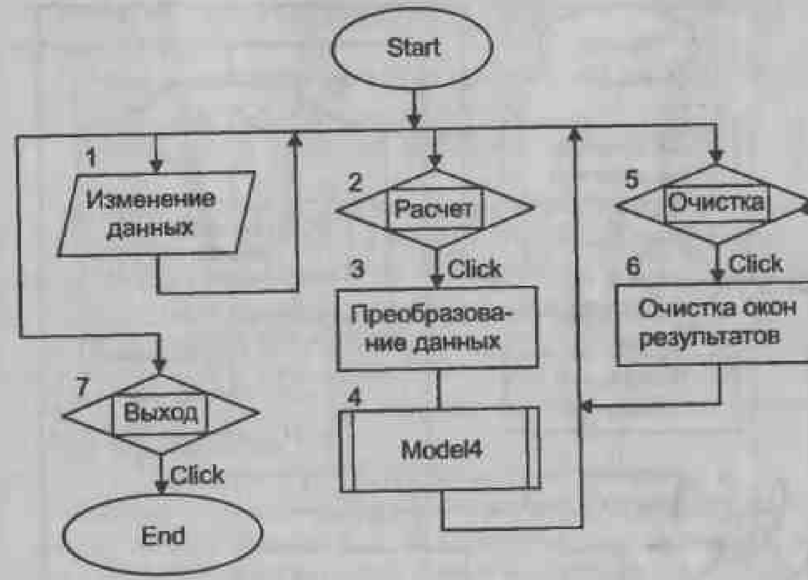


Рис. 6.2. Схема алгоритма процедур обработки прерываний

Цифрой 7 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Выход». В результате работа программы прекращается.

В состав базового комплекта исходных данных входят:

L_z – среднее число заказов в день;

$T_{обс\,j}$ (j=1, 2, 3, 4) – среднее время работы с изделием в j-м цехе для выбранного варианта структуры предприятия (дней);

P – прибыль от изготовления одного изделия ($);

C – издержки производства за период функционирования фирмы (в $);

T_D – период функционирования фирмы (дней);

$\sigma_{об}^{отн}$ – относительная величина разброса времени работы в каждом цехе;

N_p – число случайных реализаций моделируемого процесса.

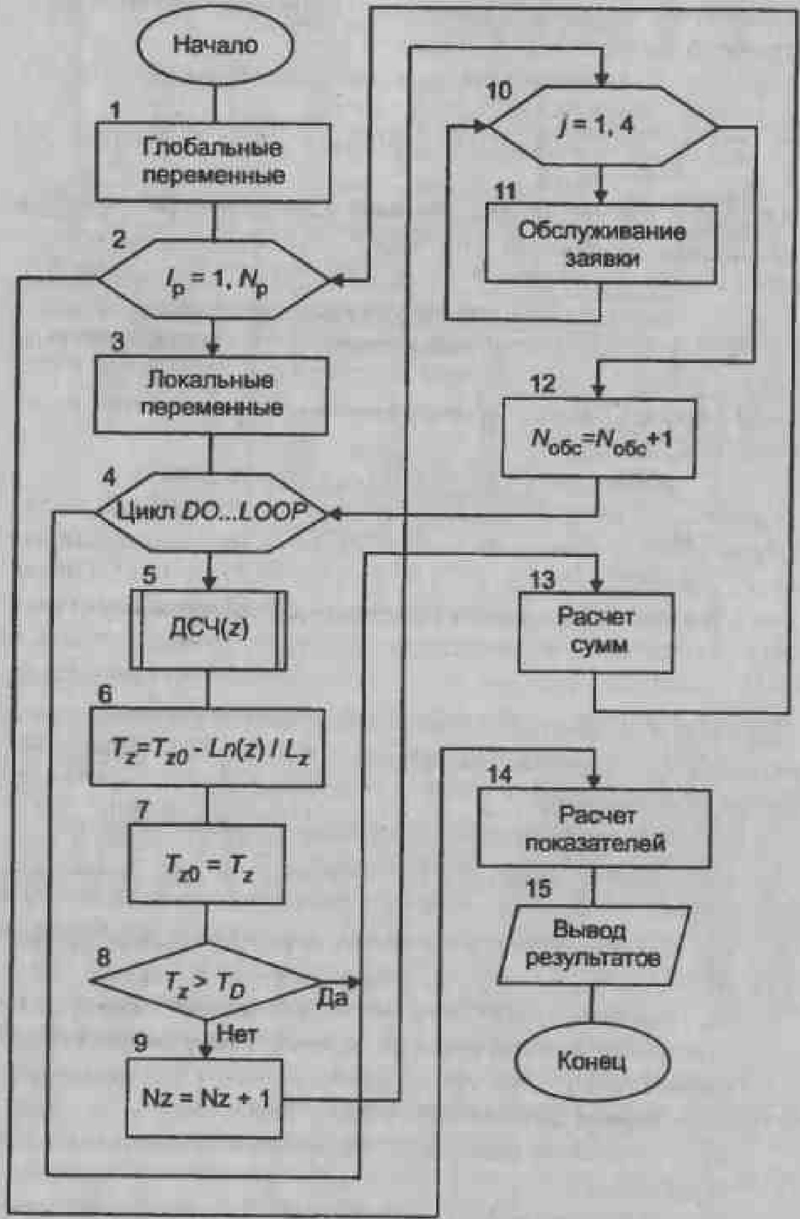


Рис. 6.3. Схема алгоритма модуля «*Model 4*»

Оператор 1 служит для обнуления глобальных переменных, к которым относятся:

M_{prof} – суммарная величина прибыли для всех случайных реализаций;

Sum2 – сумма квадратов прибыли для всех случайных реализаций.

Оператор 2 является началом цикла случайных реализаций.

Оператор 3 служит для обнуления локальных переменных, к числу которых относятся:

N_z – число заказов для текущей реализации;

$N_{обс}$ – число выполненных заказов для текущей реализации;

T_{kj} (j=1, 2, 3, 4) – время окончания работы с очередным изделием в j-м цехе;

T_{z0} – время поступления предыдущего заказа.

Оператор 4 является началом циклического перебора поступающих заказов. Оператор 5 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1). Оператор 6 служит для определения возможного значения случайной величины времени поступления заказа при условии, что время между соседними заказами имеет показательное распределение. Оператор 7 фиксирует новое значение переменной T_{z0}.

Условный оператор 8 проверяет условие прекращения функционирования производственной фирмы. Если это условие не выполняется, оператор 9 увеличивает на единицу счетчик числа поступивших заказов. В противном случае расчет данной случайной реализации заканчивается и управление передается оператору 13.

Операторы 10 и 11 описывают циклический процесс работы цехов предприятия по обслуживанию очередной заявки.

После того как изделие покинет последний цех, оператор 12 подсчитывает число изготовленных изделий. После окончания периода функционирования предприятия оператор 13 производит расчеты по формулам:

$$Prof = N_{обс} \cdot P - C;$$

$$M_{prof} = M_{prof} + Prof;$$

$$Sum2 = Sum2 + Prof \times Prof.$$

После окончания цикла случайных реализаций оператор 14 производит расчеты показателя эффективности по формулам:

$$C_{prof} = M_{prof} \,/\, N_P;$$

$$\sigma_{prof} = \sqrt{\frac{1}{N_p - 1}(Sum2 - N_p \cdot C_{prof}^2)};$$

$$G_{prof} = C_{prof} - 1{,}28 \cdot \sigma_{prof}.$$

Процесс, происходящий в каждом из цехов (блок 11 на рис. 6.3), подробно рассматривается на схеме блока обслуживания заявки (рис. 6.4).

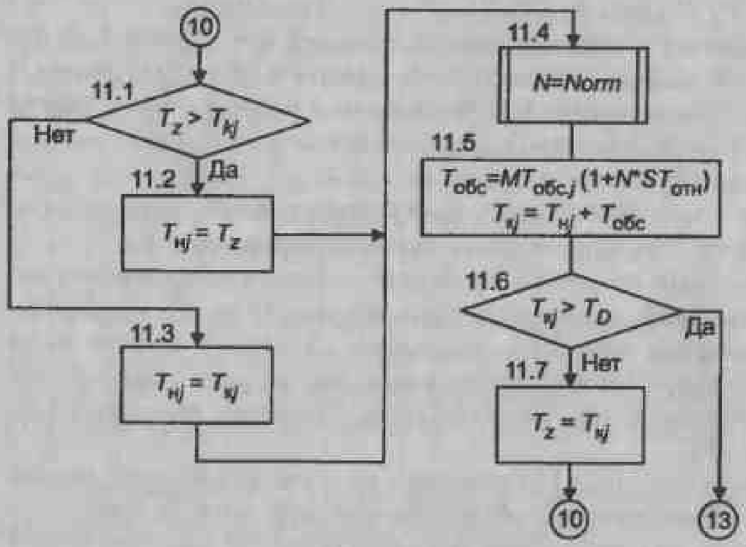


Рис. 6.4. Схема алгоритма блока обслуживания заявки

Оператор 11.1 проверяет, свободно ли оборудование j-го цеха для начала работы над очередным заказом. Если время получения заказа цехом превышает время окончания работ над преды-

дущим заказом, то в операторе 11.2 время начала работы $T_{нj}$ приравнивается времени поступления заказа T_z. В противном случае в операторе 11.3 оно приравнивается времени освобождения оборудования цеха от работы над предыдущим заказом T_{kj}.

Оператор 11.4 обращается к процедуре «Norm» – датчику случайных чисел с нормальным распределением, а оператор 11.5 определяет возможное значение случайной величины времени работы с изделием в j-м цехе.

Оператор 11.6 проверяет условие окончания процесса функционирования предприятия. Оператор 11.7 присваивает времени поступления заказа в следующий цех значение времени окончания работ в предыдущем цехе. С этого момента начинается имитация процесса работы с изделием в следующем цехе.

Отметим, что в отличие от алгоритма модели бензоколонки (см. раздел 3) в алгоритме модели производственной фирмы формируется не поток заявок, а очередная одиночная заявка (заказ), после чего сразу рассматривается процесс ее обслуживания способом последовательной проводки.

6.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Примем следующие исходные данные:

- среднее число заказов в день $L_z = 3$;
- прибыль от изготовления одного изделия $P = 300$ $;
- издержки производства за период функционирования фирмы $C = 800$ $;
- период функционирования фирмы $T_D = 90$ дней;
- число случайных реализаций моделируемого процесса $N_p = 2000$;
- относительная величина разброса времени работы в каждом цехе

$$\sigma_{об}^{отн} = 0{,}25;\ 0{,}50;\ 1{,}00;$$

- среднее время работы с изделием в j-м цехе для выбранного варианта структуры предприятия (дней) $T_{об\,j}$ (j=1, 2, 3, 4).

Рассматриваемые варианты структуры предприятия приведены в табл. 6.1.

Таблица 6.1

Варианты структуры производственной фирмы

№ варианта	Количество цехов	Среднее время работы по цехам, дни				Суммарное среднее время, дни	Числовой фактор *Fact*
		1ц	2ц	3ц	4ц		
1	1	12	0	0	0	12	1,000
2	3	8	3	1	0	12	0,667
3	4	6	4	1,5	0,5	12	0,458
4	4	4,8	3,6	2,4	1,2	12	0,300
5	4	4	3,5	2,5	2	12	0,167
6	4	3	3	3	3	12	0,000

Вариант № 1 предполагает, что работа по изготовлению изделий производится только в одном цехе. При этом значение числового фактора равно единице.

В варианте № 2 используются три цеха. Среднее время выполнения операций по изготовлению изделия в цехах распределено неравномерно, и при этом числовой фактор имеет значение 0,667.

Для вариантов № 4, 5 и 6 используются четыре цеха. Распределение времени работы указано в табл. 6.1. Для последнего варианта время распределено равномерно, и при этом числовой фактор принимает нулевое значение. Результаты моделирования представлены в табл. 6.2.

Таблица 6.2

Результаты решения задачи моделирования

№ варианта	$\sigma_{об}^{отн}$	Среднее время работы, дни				Числовой фактор	G_{prof}, $
		1-й цех	2-й цех	3-й цех	4-й цех		
1	0,25	12	0	0	0	1,000	1021
2	0,25	8	3	1	0	0,667	1918
3	0,25	6	4	1,5	0,5	0,458	2871
4	0,25	4,8	3,6	2,4	1,2	0,300	3784
5	0,25	4	3,5	2,5	2	0,167	4589
6	0,25	3	3	3	3	0,000	6001

Продолжение

№ варианта	$\sigma_{об}^{отн}$	Среднее время работы, дни				Числовой фактор	G_{prof}, $
		1-й цех	2-й цех	3-й цех	4-й цех		
7	0,50	12	0	0	0	1,000	745
8	0,50	8	3	1	0	0,667	1676
9	0,50	6	4	1,5	0,5	0,458	2441
10	0,50	4,8	3,6	2,4	1,2	0,300	3399
11	0,50	4	3,5	2,5	2	0,167	4131
12	0,50	3	3	3	3	0,000	4954
13	1,00	12	0	0	0	1,000	308
14	1,00	8	3	1	0	0,667	1035
15	1,00	6	4	1,5	0,5	0,458	1768
16	1,00	4,8	3,6	2,4	1,2	0,300	2436
17	1,00	4	3,5	2,5	2	0,167	2980
18	1,00	3	3	3	3	0,000	3487

Анализ данных, приведенных в табл. 6.2, показывает, что наибольшая прибыль достигается в том случае, когда производственная фирма имеет четыре цеха, а среднее время работы с изделием одинаково для всех цехов. В этом случае производственная линия переходит к выполнению очередного заказа сразу после того, как закончатся работы с изделием в первом цехе, т. е. в среднем через каждые 3 дня.

Наименьшая прибыль будет получена тогда, когда все производственные мощности цеха сосредоточены в одном цехе. В этом случае очередной заказ может быть принят только после того, как будет полностью изготовлено предыдущее изделие, т. е. в среднем через 12 дней.

6.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 6.5.1

В среде Visual Basic 5.0 создайте исполняемый модуль программы «Модель производственной фирмы», текст которой приведен в приложении 4. Отладьте программу с целью ликвидации формальных ошибок.

Задание 6.5.2

Проведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 6.5.3

Проведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Выберите исходные данные, произведите расчеты и проанализируйте результаты моделирования.

7. МОДЕЛЬ ТОРГОВОЙ ТОЧКИ

7.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель: – Мне снова нужна Ваша помощь. Я собираюсь вложить средства в создание подвижной торговой точки. Она будет торговать несколькими видами товара (например, тремя). Торговля может производиться в одной из трех точек (пунктов), в каждой из которых назначаются свои цены на товар. Кроме того, каждый из трех пунктов торговли имеет свои дополнительные особенности, которые заключаются в том, что в некоторых заранее не предсказуемых случаях фирме придется нести внеплановые убытки, связанные с действиями рэкетеров. Это создает неопределенность при планировании ожидаемой прибыли. Разумеется, объем продаж тоже не может быть предсказан заранее.

Мне нужна математическая модель, с помощью которой можно было бы описать данную ситуацию и выработать оптимальный способ организации торговли.

Консультант: – Очевидно, и на этот раз речь идет о разработке алгоритмической статистической модели. Имеющиеся неопределенности могут быть разрешены путем задания распределений

случайных величин, являющихся входными переменными модели. Прежде всего целесообразно предположить, что выручка от продажи какого-либо вида товара на одном из пунктов торговли имеет усеченное нормальное распределение с заданными параметрами. Поскольку имеются три вида товара и три пункта торговли, то необходимо задать девять значений математического ожидания, девять значений среднего квадратического отклонения (СКО), девять минимальных значений и девять максимальных значений случайных величин выручки. Всего 36 входных переменных.

Предприниматель: – А как описать неопределенную ситуацию, связанную с возможностью понести внеплановые убытки?

Консультант: – Нужно ввести в число входных переменных вероятность события, которое заключается в возникновении такой ситуации на каждом пункте торговли. Сам факт причинения убытка будет разыгрываться в модели по жребию как событие с заданной вероятностью. Если событие произойдет, то нужно из общей выручки вычесть величину внепланового убытка. Если же при очередном розыгрыше событие не произойдет, то убытка не будет.

Предприниматель: – А как определить величину убытка?

Консультант: – Здесь целесообразно также принять усеченное нормальное распределение, но с другими параметрами. Для трех торговых точек это составит вместе с вероятностями 15 входных переменных. К ним еще нужно добавить число случайных реализаций моделируемого процесса. Получится всего 52 входных переменных.

Предприниматель: – Многовато. Нельзя ли сократить число переменных?

Консультант: – Конечно, можно. Но для этого придется принять ряд дополнительных допущений. Прежде всего можно сократить число переменных, характеризующих разброс случайных величин выручки за проданный товар. Допустим, что среднее квадратическое отклонение дневной выручки для всех пунктов и всех видов товара составляет одну и ту же долю от соответствующей средней величины дневной выручки.

Допустим также, что минимальная дневная выручка для всех пунктов торговли и всех видов товаров равна отклонению от среднего значения выручки на одинаковое количество средних квадратических отклонений в сторону уменьшения величины выручки.

Допустим еще, что максимальная дневная выручка для всех пунктов торговли и всех видов товаров равна отклонению от среднего значения выручки на одинаковое количество средних квадратических отклонений в сторону увеличения величины выручки.

Далее допустим, что среднее квадратическое отклонение внепланового убытка для всех пунктов торговли составляет одну и ту же долю от соответствующей средней величины внепланового убытка.

Допустим еще, что минимальный внеплановый убыток для всех пунктов торговли равен отклонению от среднего значения убытка на одинаковое количество средних квадратических отклонений в сторону уменьшения величины убытка.

Наконец, допустим, что максимальный внеплановый убыток для всех пунктов торговли равен отклонению от среднего значения убытка на определенное количество средних квадратических отклонений в сторону увеличения величины убытка.

Осталось принять еще одно общее допущение о том, что относительные характеристики распределений всех случайных величин одинаковы для всех торговых точек и всех видов товара. В итоге число входных переменных сократится до 19.

Предприниматель: – Осталось выбрать показатель эффективности. Наверное, это должна быть минимальная гарантированная прибыль для каждой торговой точки. А что принять в качестве критерия эффективности?

Консультант: – Можно выбрать торговую точку, для которой величина показателя эффективности будет максимальной. Можно также использовать разработанную алгоритмическую модель для решения задачи в такой постановке: при различных величинах вероятностей внеплановых убытков определить максимально допустимые величины средних убытков, при которых условия торговли на всех пунктах становятся практически одинаковыми.

7.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Имеется подвижная торговая точка, которая производит торговлю в трех различных пунктах. Торговая точка реализует три вида товаров. Выручка, полученная от продажи каждого вида товара, является случайной величиной с усеченным нормальным распределением. Предполагается также, что на каждом пункте с некоторой заданной вероятностью торговой точке будет причинен внеплановый убыток (например, продавцу придется уплатить «дань» рэкетерам). Размер этого убытка можно считать случайной величиной, имеющей усеченное нормальное распределение с заданными параметрами.

Для каждого пункта торговли задан комплект входных параметров:

1) $T_{ср.ij}$ – среднее значение дневной выручки от продажи j-го товара на i-м пункте (i = 1, 2, 3 и j = 1, 2, 3);

2) P_i – вероятность того, что на данном пункте торговой точке будет причинен внеплановый убыток;

3) U_i – средняя величина внепланового убытка.

Кроме того, в число входных параметров входят:

4) $\sigma_T^{отн}$ – относительная величина среднего квадратического отклонения дневной выручки, одинаковая для всех пунктов торговли и всех видов товара и используемая для определения возможного значения случайной выручки T_{ij} по формуле

$$T_{ij} = T_{ср.ij}\,(1 + \sigma_T^{отн}\eta),\ (i=1, 2, 3; j=1, 2, 3),$$

где η – возможное значение случайной величины с усеченным нормальным распределением;

5) $\sigma_U^{отн}$ – относительная величина среднего квадратического отклонения внепланового убытка, одинаковая для всех пунктов торговли и используемая для определения возможного значения случайного убытка U_i по формуле

$$U_i = U_{ср.i}\,(1 + \sigma_U^{отн}),\ (i=1, 2, 3;\ j=1, 2, 3);$$

6) T_{min} – заданное минимальное значение случайной величины для эталонного усеченного нормального распределения (используемое при генерировании величины η);

7) T_{max} – заданное максимальное значение случайной величины для эталонного усеченного нормального распределения (используемое при генерировании величины η).

Величины T_{min} и T_{max} используются в процедуре, вырабатывающей возможные значения случайной величины, имеющей усеченное нормальное распределение.

Примем дополнительное допущение:

$$\sigma_T^{отн} = \sigma_U^{отн} = \sigma^{отн}.$$

Суммарная выручка от продажи всех видов товара с учетом убытка составляет случайную прибыль. В качестве показателя эффективности работы *i*-й торговой точки примем минимальную гарантированную прибыль, определяемую по следующей зависимости:

$$Gar_{пр.i} = M_{пр.i} - K_\alpha \cdot \sigma_{пр.i},$$

где $M_{пр.i}$ – математическое ожидание (среднее значение) прибыли для *i*-го пункта торговли;

$\sigma_{пр.i}$ – среднее квадратическое отклонение прибыли для *i*-го пункта торговли;

K_α – квантиль нормального распределения, соответствующий заданной надежности α (K_α=1,28 при α=0,9).

Примечание. Предполагается, что случайная величина прибыли имеет нормальное распределение.

Задача исследования может заключаться в определении условий, при которых все пункты торговли будут равноценными по величине приносимой прибыли. В этих условиях необходимо установить, какова функциональная связь между увеличением среднего убытка и компенсирующим его уменьшением вероятности причинения убытка.

7.3. АЛГОРИТМ МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0. Общий вид (макет) стартовой формы показан на рис.7.1.

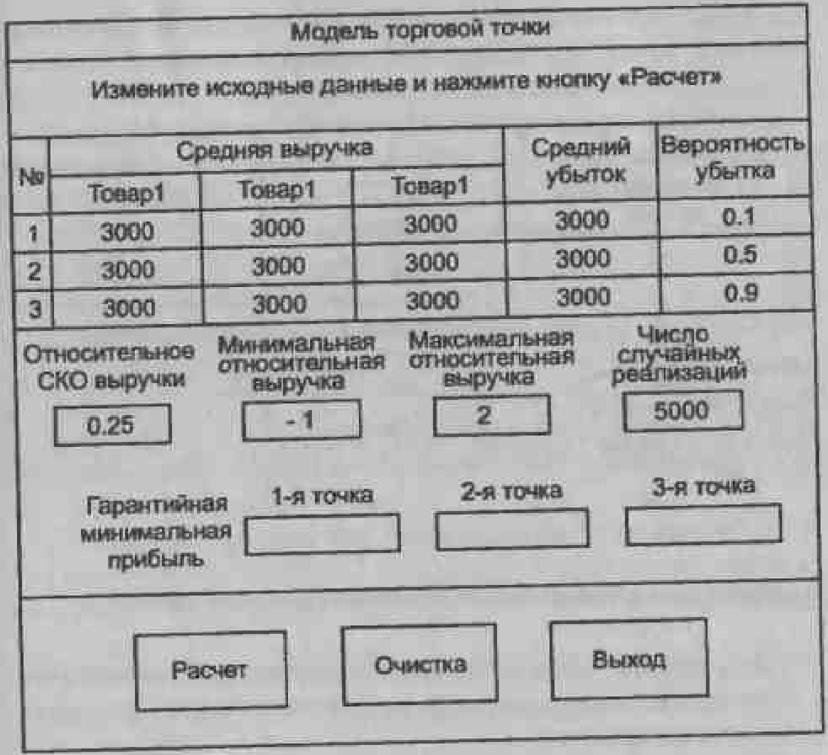


Рис. 7.1. Макет стартовой формы

В нее включены следующие объекты управления:

1) девятнадцать текстовых полей для изменения исходных данных;

2) три текстовых поля для вывода значений минимальной гарантированной прибыли;

3) командная кнопка «Расчет» для запуска расчетного модуля;

4) командная кнопка «Очистка» для очистки полей результатов расчета;

5) командная кнопка «Выход» для окончания работы программы.

Схема алгоритма процедур обработки прерываний показана на рис. 7.2.

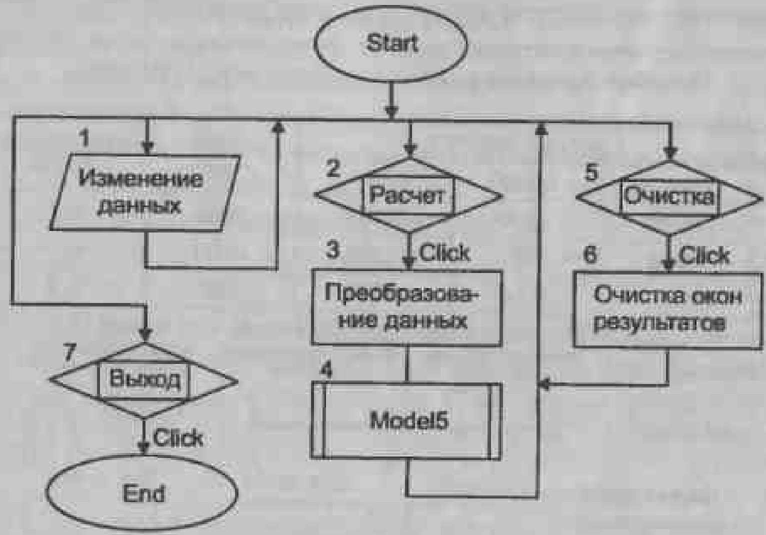


Рис. 7.2. Схема алгоритма процедур обработки прерываний

После нажатия кнопки «Start» активизируется стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

Цифрой 1 обозначено действие, заключающееся в корректировке исходных данных. Необходимые изменения вносятся в соответствующие текстовые поля.

Цифрой 2 обозначено действие, заключающееся в нажатии (с помощью мыши) кнопки «Расчет». В процедуре, связанной с этой кнопкой, оператор 3 осуществляет перевод исходных данных из символьной формы в числовую. Затем оператор 4 обращается к модулю общего назначения «Model 5». Схема алгоритма этого модуля приведена на рис. 7.3. После окончания работы модуля и выдачи на экран результатов моделирования работа процедуры, связанной с кнопкой «Расчет», заканчивается. Программа вновь переходит в режим ожидания действий пользователя.

Цифрой 5 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Очистка». В процедуре, связанной с ней, производится очищение текстовых полей для вывода показателя эффективности и значения числового фактора. Затем может быть произведено изменение исходных данных и проведены новые расчеты с использованием кнопки «Расчет».

Цифрой 7 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Выход». В результате работа программы прекращается.

Оператор 1 обнуляет глобальные переменные:

суммарную прибыль для каждого пункта торговли для всех случайных реализаций моделируемого процесса $T_{sum.i}$ (i=1, 2, 3);

сумму квадратов прибыли для каждого пункта торговли для всех случайных реализаций моделируемого процесса $T_{sum2.i}$ (i=1, 2, 3).

Оператор 2 является началом циклического перебора случайных реализаций, а оператор 3 осуществляет циклический перебор пунктов торговли.

Оператор 4 обращается к процедуре, вырабатывающей возможные значения случайной величины η с усеченным нормальным распределением при заданных нормированных параметрах $T_{\min}$ и $T_{\max}$.

Оператор 5 определяет возможное значение выручки для 1-го пункта торговли от продажи 1-го вида товара и помещает результат в сумматор:

$$T_S = T_{i1} = T_{ср.i1}\,(1 + \eta \cdot \sigma^{отн}),$$

где η – возможное значение случайной величины с усеченным нормальным распределением.

Операторы 6 и 7 определяют случайное приращение выручки для i-го пункта торговли от продажи 2-го вида товара и помещают результат в сумматор:

$$T_S = T_S + T_{i2} = T_S + T_{ср\text{-}i2}\,(1 + \eta \cdot \sigma^{отн}).$$

Аналогичным образом операторы 8 и 9 определяют случайное приращение выручки для i-го пункта торговли от продажи 3-го вида товара и помещают результат в сумматор:

$$T_S = T_S + T_{i3} = T_S + T_{ср.i3}\,(1 + \eta \cdot \sigma^{отн}).$$

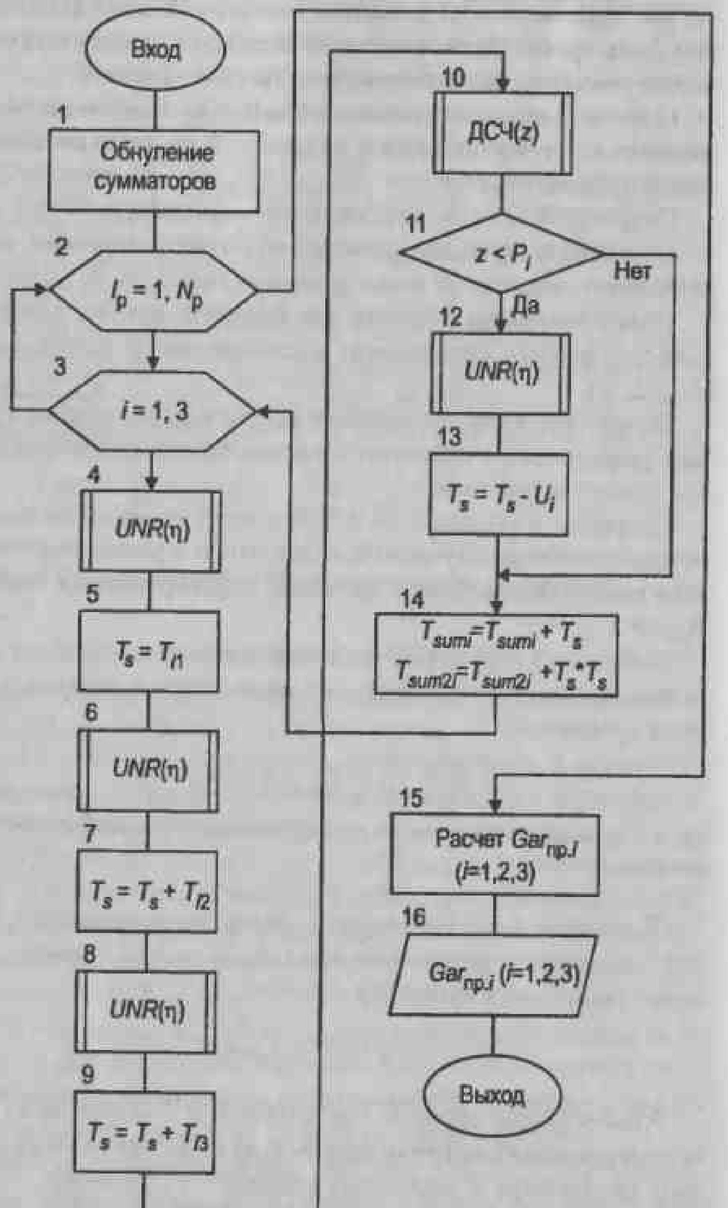


Рис. 7.3. Схема алгоритма модуля «*Model 5*»

Операторы 10 и 11 моделируют появление случайного события с заданной вероятностью. Оператор 10 с помощью датчика случайных чисел с равномерным распределением в интервале (0,1) вырабатывает возможное значение этой величины z. Если в операторе 11 выполняется условие $z < P_i$, то считается, что в данном пункте торговли имели место внеплановые убытки.

Операторы 12 и 13 определяют возможное значение внепланового убытка, который вычитается из накопленной суммарной выручки:

$$T_S = T_S - U_i = T_S - U_{ср.i}(1 + \eta \cdot \sigma^{отн}).$$

В операторе 14 определяются:

• суммарная величина прибыли для всех случайных реализаций моделируемого процесса, определяемая по рекурсивной формуле

$$T_{sum.i} = T_{sum.i} + T_S;$$

• сумма квадратов величин прибыли для всех случайных реализаций моделируемого процесса, определяемая по рекурсивной формуле

$$T_{sum2.i} = T_{sum2.i} + T_S \cdot T_S.$$

После этого управление в алгоритме передается оператору 3 для моделирования процесса продажи товаров на 2-м, а затем и на 3-м пункте торговли.

Оператор 15 служит для расчета показателя эффективности моделируемого процесса по следующим зависимостям:

$$M_{пр.i} = \frac{1}{N_p} T_{sum.i};$$

$$\sigma_{пр.i} = \sqrt{\frac{1}{N_p - 1}(T_{sum2.i} - N_p \cdot M^2_{пр.i})};$$

$$Gar_{\,пр.i} = M_{пр.i} - K_\alpha \cdot \sigma_{пр.i},$$

где $M_{пр.i}$ – математическое ожидание (среднее значение) прибыли для i-го пункта торговли;

$\sigma_{пр.i}$ – среднее квадратическое отклонение прибыли для i-го пункта торговли;

$Gar_{\,пр.i}$ – минимальная гарантированная прибыль с уровнем гарантии α (при α=0,95 квантиль нормального распределения K_α=1,645).

Оператор 16 выводит на экран значения показателей эффективности для всех трех пунктов торговли.

7.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Примем следующие исходные данные:

- средняя выручка от продажи товара 1-го вида на 1-м пункте торговли $T_{ср \cdot 11}$=3000 руб.;
- средняя выручка от продажи товара 2-го вида на 1-м пункте торговли $T_{ср \cdot 12}$=3000 руб.;
- средняя выручка от продажи товара 3-го вида на 1-м пункте торговли $T_{ср \cdot 13}$=3000 руб.;
- средняя выручка от продажи товара 1-го вида на 2-м пункте торговли $T_{ср \cdot 21}$=3000 руб.;
- средняя выручка от продажи товара 2-го вида на 2-м пункте торговли $T_{ср \cdot 22}$=3000 руб.;
- средняя выручка от продажи товара 3-го вида на 2-м пункте торговли $T_{ср \cdot 23}$=3000 руб.;
- средняя выручка от продажи товара 1-го вида на 3-м пункте торговли $T_{ср 31}$=3000 руб.;
- средняя выручка от продажи товара 2-го вида на 3-м пункте торговли $T_{ср \cdot 32}$=3000 руб.;
- средняя выручка от продажи товара 3-го вида на 3-м пункте торговли $T_{ср \cdot 33}$=3000 руб.;
- средние внеплановые убытки на 1-м пункте торговли $U_{ср \cdot 1}$=3000 руб.;
- средние внеплановые убытки на 2-м пункте торговли $U_{ср \cdot 2}$=3000 руб.;
- средние внеплановые убытки на 3-м пункте торговли $U_{ср \cdot 3}$=3000 руб.;
- вероятность того, что на 1-м пункте торговли будут внеплановые убытки, P_1=0,1;
- вероятность того, что на 2-м пункте торговли будут внеплановые убытки, P_2=0,5;
- вероятность того, что на 3-м пункте торговли будут внеплановые убытки, P_3=0,9;

- нормированное среднее квадратическое отклонение случайных величин дневной выручки и внепланового убытка $\sigma^{отн}$ = 0,25;
- нормированное минимальное значение случайных величин дневной выручки и внепланового убытка $T_{min} = -1$;
- нормированное максимальное значение случайных величин дневной выручки и внепланового убытка $T_{max} = +2$;
- квантиль нормального распределения K_α = 1,645 (надежность α=0,95);
- число случайных реализаций $N_р$ = 20 000.

Требуется установить, насколько можно допустить увеличение среднего внепланового убытка на 1-м и 2-м пунктах торговли, чтобы при фиксированных вероятностях нанесения убытков прибыль на них была бы практически такой же, как на 3-м пункте.

Варьируя $U_{ср \cdot 1}$ и $U_{ср.2}$, можно подобрать значения, при которых величины Gar1, Gar2 и Gar3 будут практически одинаковы. Расчетные данные сведены в табл. 7.1.

Таблица 7.1

Расчетные данные

Исходные данные							Показатели		
№ варианта	U_1	P_1	U_2	P_2	U_3	P_3	*Gar1*	*Gar2*	*Gar3*
1	7750	0,1	3400	0,5	3000	0,9	4349	4226	4302
2	7750	0,1	3400	0,5	3000	0,9	4309	4234	4285
3	7750	0,1	3400	0,5	3000	0.9	4243	4236	4299
4	7750	0,1	3400	0,5	3000	0.9	4305	4228	4321
5	7750	0,1	3400	0.5	3000	0.9	4354	4210	4332
Средние значения:							4312	4307	4300

Как видно из табл. 7.1, при выбранном числе реализаций показатели эффективности являются случайными величинами и имеют заметный разброс.

Для повышения точности результатов было проведено 5 серий расчетов при одинаковых исходных данных. Анализируя полученные осредненные данные, приведенные в таблице, можно сделать вывод о том, что с достаточной для практики точностью одинаковые условия торговли в каждом из трех пунктов обеспечиваются при следующих значениях входных параметров (табл. 7.2).

Таблица 7.2

Результаты моделирования

№ пункта	Вероятность внепланового убытка	Допустимый средний убыток, $	Средняя дневная прибыль, $
1	0,1	7750	4312
2	0,5	3400	4307
3	0,9	3000	4300

7.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 7.5.1

В среде Visual Basic 5.0 создайте исполняемый модуль программы «Модель торговой точки», текст которой приведен в приложении 5. Произведите отладку программы с целью ликвидации формальных ошибок.

Задание 7.5.2

Проведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 7.5.3

Проведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Выберите исходные данные, произведите расчеты и проанализируйте результаты моделирования.

8. ФИНАНСОВАЯ МОДЕЛЬ

8.1. РАЗГОВОР ПРЕДПРИНИМАТЕЛЯ С КОНСУЛЬТАНТОМ

Предприниматель: – Мне опять нужен Ваш совет. Я собираюсь вложить средства в строительство нового предприятия, которое будет выпускать определенную продукцию, пользующуюся спросом на рынке. Аналогичную продукцию выпускают и некоторые другие фирмы, поэтому придется действовать в условиях конкуренции.

Консультант: – Какие данные можно считать известными?

Предприниматель: – Можно приближенно оценить предполагаемые эксплуатационные расходы по выпуску продукции, т. е. можно считать известными математическое ожидание (среднее значение) расходов и среднее квадратическое отклонение этих расходов.

Консультант: – Значит, можно принять допущение о том, что расходы имеют нормальное распределение с заданными параметрами. А что известно относительно возможностей сбыта продукции, каковы характеристики рынка?

Предприниматель: – Можно предположить, что емкость рынка также имеет нормальное распределение с некоторыми известными параметрами: математическим ожиданием и средним квадратическим отклонением. Хуже обстоит дело с определением характеристик той доли в рынке, которую может завоевать наше предприятие после его вступления в строй. Единственно, что можно предсказать, – это средняя величина этой доли. Вид распределения неизвестен, и нет основания для того, чтобы считать его нормальным.

Консультант: – В этом случае при создании модели исследуемого процесса можно использовать распределение произвольного типа, например кусочно-равномерное. Можно выбрать несколько вариантов такого распределения и проанализировать реакцию модели на изменение его параметров.

Предприниматель: – Каким же показателем будет оцениваться эффективность предприятия?

Консультант: – Логично будет выбрать в качестве показателя эффективности минимальную гарантированную прибыль от продажи продукции. При этом предполагается, что случайная величина прибыли имеет нормальное распределение.

8.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Пусть намечается строительство нового предприятия при следующих условиях

1. Выпуск продукции связан с эксплуатационными расходами, которые имеют нормальное распределение с заданными параметрами: математическим ожиданием расхода M_{rash} и средним квадратическим отклонением расхода σ_{rash}.
2. Емкость рынка, где должна реализовываться продукция предприятия, имеет нормальное распределение с заданными параметрами: математическим ожиданием емкости рынка M_{ryn} и средним квадратическим отклонением емкости рынка σ_{ryn}.
3. Доля предприятия в рынке является неопределенной и может быть задана некоторой произвольной функцией распределения (например, кусочно-равномерной функцией).
4. Случайная прибыль предприятия определяется по следующей зависимости:

$$Prof = Ryn \cdot Dol - Rash,$$

где Ryn – случайная величина емкости рынка;
Dol – случайная величина доли предприятия на рынке;
$Rash$ – случайная величина эксплуатационных расходов предприятия.

Выходными характеристиками модели являются:

• сумма случайных величин прибыли для N_p случайных реализаций

$$S_{prof} = \sum_{i=1}^{N_p} \Pr of;$$

• сумма квадратов случайных величин прибыли для N_p случайных реализаций.

$$S2_{prof} = \sum_{i=1}^{N_p} \Pr of^2.$$

Показателем эффективности работы предприятия является минимальная гарантированная прибыль, определяемая по следующим зависимостям:

$$M_{prof} = \frac{1}{N_p} S_{prof};$$

$$\sigma_{prof} = \sqrt{\frac{1}{N_p - 1}(S2_{prof} - N_p \cdot M_{prof}^2)};$$

$$G_{prof} = M_{prof} - K_\alpha \cdot \sigma_{prof},$$

где M_{prof} – математическое ожидание (среднее значение) прибыли;
σ_{prof} – среднее квадратическое отклонение прибыли;
K_α – квантиль нормального распределения, соответствующий заданной надежности α ($K_\alpha = 1{,}28$ при $\alpha = 0{,}9$);
G_{prof} – минимальная гарантированная прибыль.

8.3. АЛГОРИТМ МОДЕЛИ

В качестве языка программирования для разработки компьютерной модели рассматриваемого процесса выбран Visual Basic 5.0. Общий вид (макет) стартовой формы показан на рис. 8.1

В нее включены следующие объекты управления: несколько меток с заголовками объектов, двенадцать текстовых полей для корректировки исходных данных, три текстовых поля для вывода результатов моделирования и три командные кнопки для управления работой программы.

Схема алгоритма процедур обработки прерываний показана на рис. 8.2.

После нажатия кнопки «Start» активизируется стартовая форма. С этого момента программа находится в режиме ожидания действий пользователя.

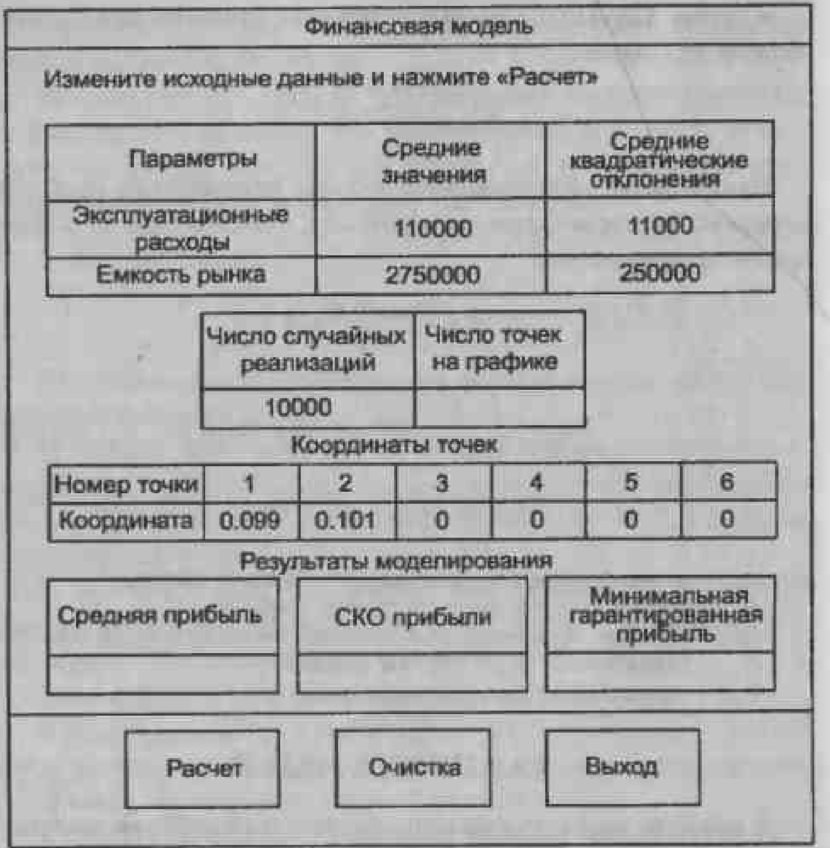


Рис. 8.1. Макет стартовой формы

Цифрой 1 обозначено действие, заключающееся в корректировке исходных данных. Необходимые изменения вносятся в соответствующие текстовые поля.

Цифрой 2 обозначено действие, заключающееся в нажатии (с помощью мыши) кнопки «Расчет». В процедуре, связанной с этой кнопкой, оператор 3 осуществляет перевод исходных данных из символьной формы в числовую. Затем оператор 4 обращается к модулю общего назначения «Model 6». Схема алгоритма этого

модуля приведена на рис. 8.3. После окончания работы модуля и выдачи на экран результатов моделирования работа процедуры, связанной с кнопкой «Расчет», заканчивается. Программа вновь переходит в режим ожидания действий пользователя.

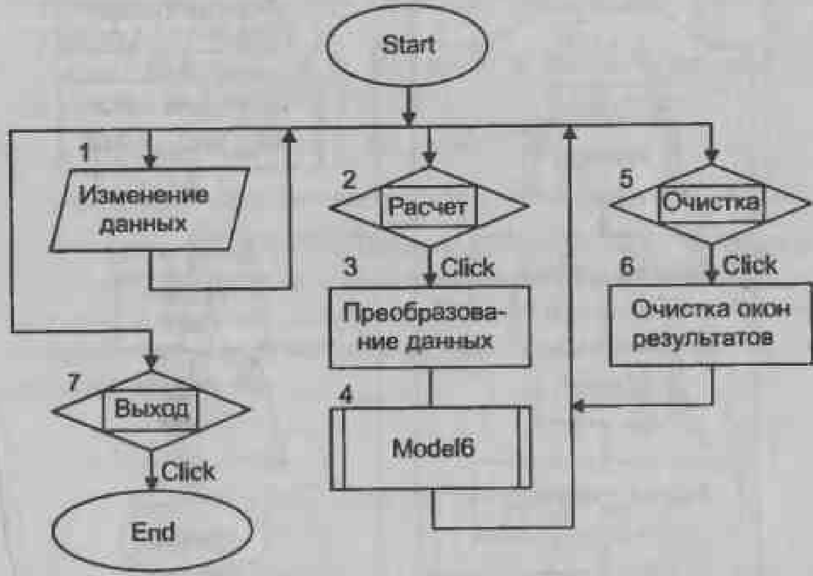


Рис. 8.2. Схема алгоритма процедур обработки прерываний

Цифрой 5 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Очистка». В процедуре, связанной с ней, производится очищение текстовых полей для вывода выходных характеристик модели. Затем может быть произведено изменение исходных данных и проведены новые расчеты с использованием кнопки «Расчет».

Цифрой 7 на схеме обозначено действие пользователя, заключающееся в нажатии кнопки «Выход». В результате работа программы прекращается.

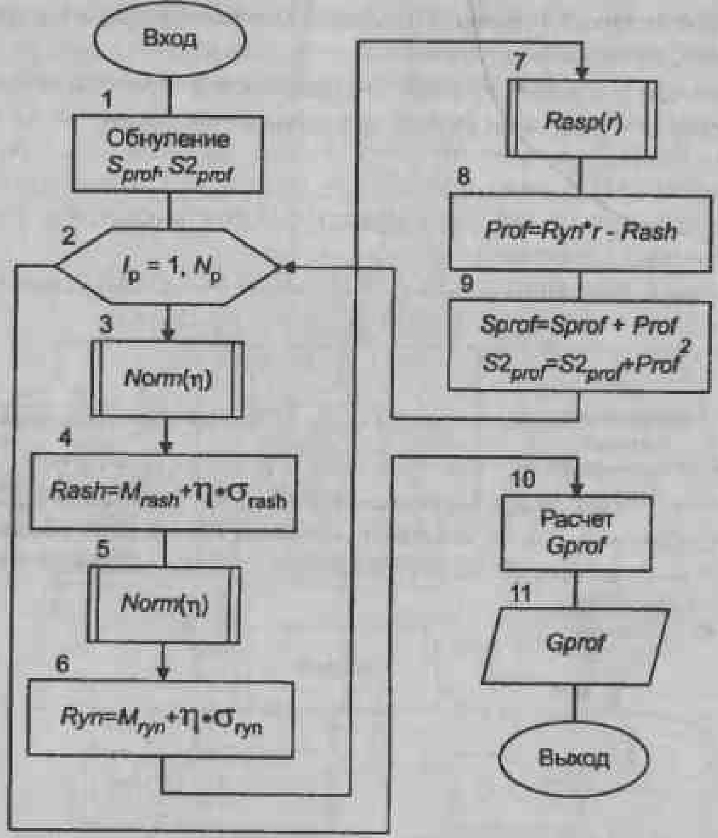


Рис. 8.3. Схема алгоритма модуля *«Model 6»*

Оператор 1 производит обнуление глобальных переменных, к которым относятся:

- сумма прибылей для всех случайных реализаций;
- сумма квадратов прибылей для всех случайных реализаций.

Оператор 2 является началом цикла случайных реализаций.

Оператор 3 обращается к процедуре, вырабатывающей возможные значения нормированных и центрированных случайных величин с нормальным распределением. Оператор 4 определяет случайное значение эксплуатационных расходов.

Операторы 5 и 6 аналогичным образом определяют случайную величину емкости рынка.

Оператор 7 обращается к процедуре, которая определяет возможное значение случайной доли предприятия в рынке. Оператор 8 определяет величину случайной прибыли для одной реализации моделируемого процесса. В операторе 9 происходит накопление сумм прибылей и сумм квадратов прибылей для всех случайных реализаций.

После окончания цикла случайных реализаций оператор 10 определяет показатель эффективности по формуле

$$G_{prof} = M_{prof} - K_{\alpha} \cdot \sigma_{prof}$$

Оператор 11 выводит результаты моделирования на экран.

Случайная доля предприятия в рынке согласно принятому допущению имеет кусочно-равномерное распределение в выбранном диапазоне. Схема алгоритма процедуры, генерирующей возможные значения случайной величины с таким распределением, показана на рис. 8.4.

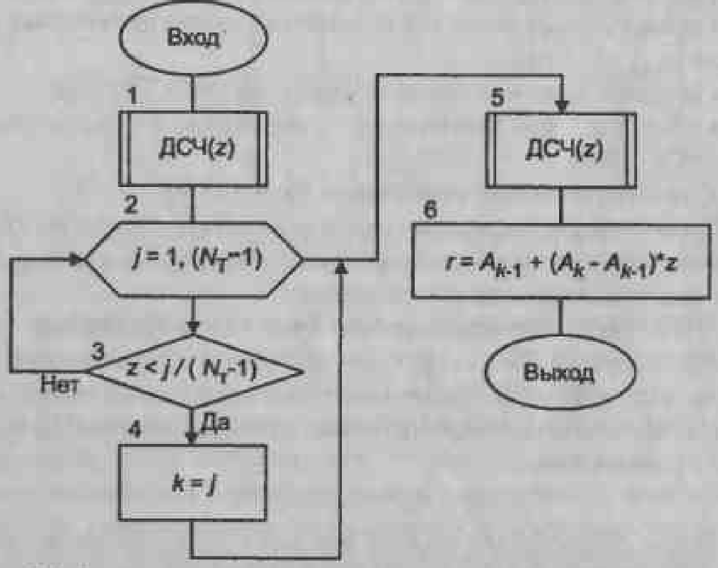


Рис. 8.4. Схема алгоритма процедуры генерирования случайных величин с кусочно-равномерным распределением

Оператор 1 обращается к стандартной процедуре генерирования случайной величины с равномерным распределением в интервале (0,1).

Оператор 2 является заголовком цикла, в котором поочередно рассматриваются все участки выбранного диапазона. Заметим, что число участков на единицу меньше числа граничных точек.

Оператор 3 проверяет условие попадания в j-й участок, а оператор 4 фиксирует номер участка.

Оператор 5 вновь обращается к генератору случайных чисел с равномерным распределением в интервале (0,1).

Оператор 6 определяет значение случайной переменной r по формуле

$$r = A_{k-1} + Z \cdot (A_k - A_{k-1}).$$

8.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Примем следующие исходные данные:

- среднее значение эксплуатационных расходов M_{rash} = $110 000;
- среднее квадратическое отклонение эксплуатационных расходов s_{rash} =$11 000;
- среднее значение емкости рынка M_{rym} =$2 750 000;
- среднее квадратическое отклонение емкости рынка σ_{rym} =$250 000;
- число случайных реализаций N_P = 10 000.

Варьируемыми переменными будем считать параметры кусочно-равномерного распределения доли предприятия в рынке. Рассмотрим три варианта распределения.

Для первого варианта примем, что число граничных точек $N_T = 2$ (диапазон состоит из одного участка). Пусть среднее значение доли равно 0,1. Граничные точки расположим симметрично относительно среднего значения. Выберем следующие значения их координат:

$$A_0 = 0{,}099 \text{ и } A_1 = 0{,}101.$$

Таким образом, для первого варианта степень неопределенности достаточно мала, доля предприятия в рынке практически постоянна и составляет 10% общей емкости рынка.

Для второго варианта примем, что число граничных точек $N_T = 6$ (диапазон состоит из пяти участков). Пусть среднее значение случайной переменной по-прежнему равно 0,1. Граничные точки расположим симметрично относительно среднего значения. Выберем следующие значения их координат:

$$A_0=0{,}035;\ A_1= 0{,}075;\ A_2 =0{,}095;\ A_3 =0{,}105;\ A_4 =0{,}125;\ A_5 =0{,}165.$$

Плотности распределения вероятностей определяются из условия, что вероятности попадания на любой из участков должны быть одинаковы и равны величине $1/(N_T - 1)$. Вид полученного распределения показан на рис. 8.5.

Таким образом, для второго варианта доля предприятия в рынке характеризуется достаточной степенью неопределенности. Случайная величина этой доли неравномерно распределена в диапазоне от 0,035 до 0,165.

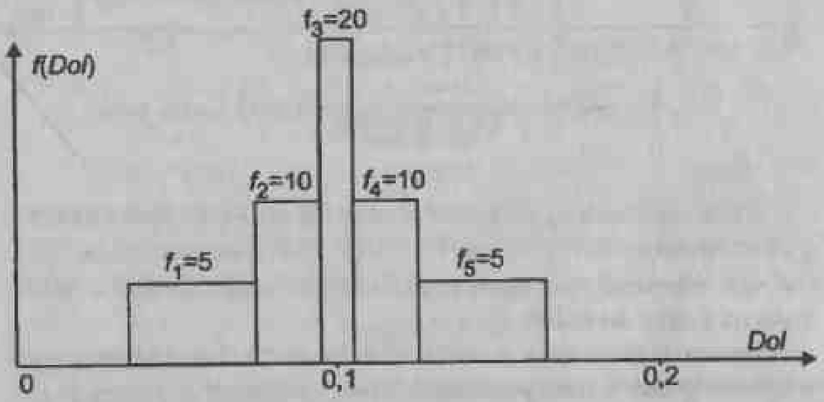


Рис. 8.5. Кусочно-равномерное распределение доли в рынке (второй вариант)

Для третьего варианта примем, что число граничных точек N_T также равно шести (диапазон состоит из пяти участков). Пусть среднее значение случайной переменной по-прежнему равно 0,1. Граничные точки расположим несимметрично относительно значения Dol = 0,1. Выберем следующие значения их координат:

$$A_0=0{,}035;\ A_1= 0{,}075;\ A_2 =0{,}095;\ A_3 =0{,}105;\ A_4 =0{,}155;\ A_5 =0{,}255.$$

Так же, как и для других вариантов, условие, которому должно удовлетворять кусочно-равномерное распределение, состоит в том, что вероятности попадания на любой из участков должны быть одинаковы и равны величине $1/(N_T-1)$. Вид полученного распределения показан на рис. 8.6.

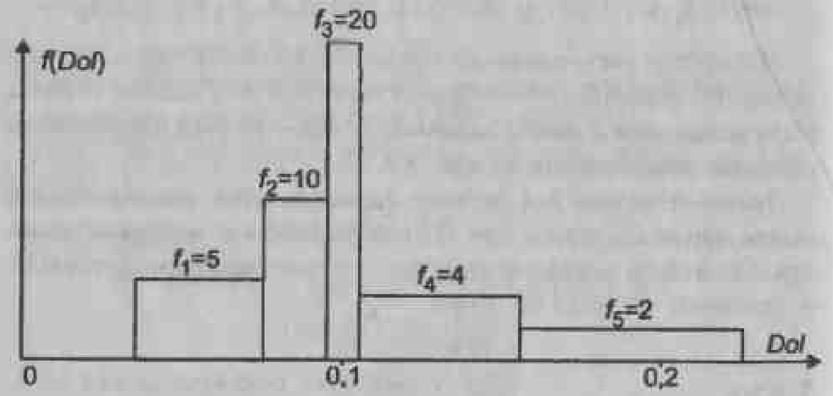


Рис. 8.6. Кусочно-равномерное распределение доли в рынке (третий вариант)

Таким образом, для третьего варианта доля предприятия в рынке характеризуется еще большей неопределенностью. Случайная величина этой доли неравномерно распределена в диапазоне от 0,035 до 0,255.

Варианты исходных данных, относящиеся к описанию кусочно-равномерных распределений доли предприятия в рынке, сведены в табл. 8.1.

Таблица 8.1

Параметры кусочно-равномерных распределений

№ варианта	Число точек	Координаты точек (границ участков)					
		1	2	3	4	5	6
1	2	0,099	0,101	–	–	–	–
2	6	0,035	0,075	0,095	0,105	0,125	0,165
3	6	0,035	0,075	0,095	0,105	0,155	0,255

Результаты моделирования отражены в табл. 8.2

Таблица 8.2

Результаты моделирования

№ варианта	Mprof,$	σprof, $	Gprof, $
1	164,6	27,2	129,8
2	165,2	90,8	49,0
3	205,1	150,9	11,9

Анализ приведенных данных показывает, что с увеличением степени неопределенности при описании доли предприятия в рынке средняя прибыль растет, однако минимальная гарантированная прибыль уменьшается из-за увеличения разброса случайной величины прибыли.

8.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 8.5.1

В среде Visual Basic 5.0 создайте исполняемый модуль программы «Финансовая модель Герца», текст которой приведен в приложении 6. Произведите отладку программы с целью ликвидации формальных ошибок.

Задание 8.5.2

Проведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 8.5.3

Проведите самостоятельное исследование закономерностей функционирования фирмы с помощью алгоритмической модели. Выберите исходные данные, произведите расчеты и проанализируйте результаты моделирования.

9. МОДЕЛЬ ЗВЕНА УПРАВЛЕНИЯ

9.1. РАЗГОВОР ДИРЕКТОРА С КОНСУЛЬТАНТОМ

Директор: – Я нуждаюсь в Вашей помощи в деле разработки математической модели управленческого процесса. Вместе с двумя моими заместителями по определенным дням я провожу прием посетителей. На мой взгляд, в настоящее время этот процесс построен неудачно.

Мы имеем дело с двумя категориями посетителей.

Первую группу составляют посетители, которые приходят в наше учреждение с весьма срочными делами, требующими безотлагательного решения. Их вначале принимает мой первый заместитель, а затем отправляет ко мне для окончательного решения возникшего вопроса. Число посетителей заранее не определено, время прихода следующего посетителя также точно неизвестно.

Вторую группу образуют посетители, которые обращаются к нам с менее срочными делами. Их вначале принимает мой второй заместитель, а затем также направляет ко мне для окончательного решения вопроса. Здесь также нет точных данных о числе посетителей и интервале времени между ними.

В результате создается такая ситуация, при которой мои личные временны́е затраты вдвое превышают затраты рабочего времени у каждого из моих заместителей. Кроме того, у дверей моего кабинета создается очередь, что вызывает справедливое недовольство у посетителей.

Я хотел бы перестроить порядок приема посетителей так, чтобы выровнять временну́ю нагрузку всех трех участников данного управленческого процесса. Я готов пойти на то, чтобы предоставить обоим моим заместителям право в определенном числе случаев принимать окончательные решения по рассматриваемым вопросам и только часть дел отсылать ко мне. При этом

те посетители, которые приходят со срочными делами, должны попадать ко мне в обход очереди из посетителей с менее срочными делами.

Поскольку возможностей проводить натурный эксперимент у меня нет, я хотел бы провести его на математической модели. Что для этого нужно сделать?

Консультант: – Прежде всего нужно формализовать, т. е. формально описать изучаемый процесс. В качестве типовой математической схемы в данном случае подойдет схема системы массового обслуживания с двумя каналами, двумя фазами обслуживания и двумя видами приоритета заявок. Деятельность первого заместителя описывается первой фазой первого канала, в которую поступает поток заявок высшего (первого) приоритета, имеющий заданные параметры. Деятельность второго заместителя описывается первой фазой второго канала, в которую поступает поток заявок низшего (второго) приоритета, также имеющий заданные параметры. Тогда деятельность директора учреждения описывается первой фазой второго канала, в которую могут поступать заявки, прошедшие через первые фазы обоих каналов. Второй фазы второго канала не существует. Во вторую фазу первого канала должны передаваться не все заявки, а только определенная их часть (рис. 9.1).

Процентная доля от числа посетителей, поступающих на прием к директору, составляет P_1% для первого потока и P_2% для второго потока. Что же касается доли от числа посетителей, дела

Рис. 9.1. Схема работы звена управления

которых будут окончательно решаться на уровне заместителей директора, то они будут составлять соответственно $(100 - P_1)\%$ и $(100 - P_2)\%$ для первого и второго каналов.

Директор: – Какие исходные данные нужно будет ввести в модель?

Консультант: – Логично предположить, что время между приходом двух соседних посетителей имеет показательное распределение, т. е. поток заявок – простейший. Тогда будет достаточно задать интенсивности каждого потока заявок и среднее время их обслуживания в каждой фазе каждого канала.

Можно принять допущение о том, что время обслуживания имеет показательное распределение. Тогда всего получается шесть входных параметров. Время ожидания можно, по-видимому, считать неограниченным.

Директор: – А что будут представлять собой выходные и управляющие характеристики модели?

Консультант: – Управляющими характеристиками модели являются значения P_1 и P_2. Выходными характеристиками являются относительные величины занятости каждого из трех лиц, участников управленческой деятельности, выраженные в процентах от продолжительности рабочего дня.

9.2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ

Имеется двухфазная двухканальная система массового обслуживания с двумя приоритетами заявок и неограниченным ожиданием (рис. 9.2).

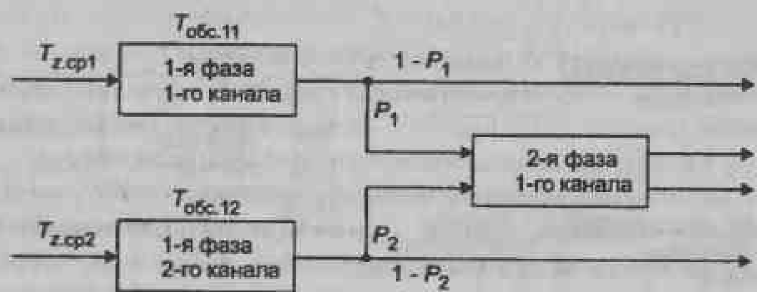


Рис. 9.2. Схема двухканальной двухфазной СМО с двумя приоритетами заявок

Поток посетителей 1-го приоритета поступает в 1-ю фазу 1-го канала. Поток – простейший со средним временем между заявками $T_{z.cp.1}$. Время обслуживания заявки имеет показательное распределение со средним временем обслуживания $T_{обс.11}$.

Далее поток заявок разделяется на два потока. С вероятностью P_1 заявка поступает во 2-ю фазу 1-го канала, а с вероятностью $(1 - P_1)$ покидает систему обслуженной.

Время обслуживания заявок 1-го приоритета во 2-й фазе 1-го канала имеет показательное распределение со средним временем обслуживания $T_{обс.21}$.

Поток посетителей 2-го приоритета поступает в 1-ю фазу 2-го канала. Поток – простейший со средним временем между заявками $T_{z.cp.2}$. Время обслуживания заявки имеет показательное распределение со средним временем обслуживания $T_{обс.12}$.

Далее поток заявок разделяется на два потока. С вероятностью P_2 заявка поступает во 2-ю фазу 1-го канала, а с вероятностью $(1 - P_2)$ покидает систему обслуженной.

Время обслуживания заявок 2-го приоритета во 2-й фазе 1-го канала имеет показательное распределение со средним временем обслуживания $T_{обс.22}$.

Дисциплина обслуживания предусматривает неограниченность ожидания для всех заявок и абсолютное превосходство заявок 1-го приоритета по отношению к заявкам 2-го приоритета. Это означает, что при поступлении заявки 1-го приоритета во 2-ю фазу 1-го канала обслуживание заявки 2-го приоритета прерывается, но затем может продолжаться после окончания обслуживания заявки 1-го приоритета и освобождения канала. Количество прерываний не ограничено.

Время окончания функционирования системы T_{fin} задано.

Показателями, характеризующими процесс функционирования системы, является относительное время занятости каждой фазы каждого канала по отношению к периоду функционирования.

Переменными управления являются величины P_1 и P_2, которые могут принимать значения в интервале (0,1). Оптимальный режим работы характеризуется сочетанием значений P_1 и P_2, при которых относительное время занятости для первых фаз каждого канала и второй фазы практически одинаково.

9.3. АЛГОРИТМ МОДЕЛИ

9.3.1. Укрупненная схема алгоритма модели

Для разработки имитационной модели используем алгоритмический язык Паскаль 7.0. Укрупненная схема алгоритма модели показана на рис. 9.3.

Оператор 1 обращается к процедуре корректировки исходных данных, к числу которых относятся переменные $T_{з.ср.1}$, $T_{з.ср.2}$, $T_{обс.11}$, $T_{обс.12}$, $T_{обс.21}$, $T_{обс.22}$, T_{fin}, N_p (число случайных реализаций).

Оператор 2 обеспечивает циклический перебор трех значений переменной управления P_1. Оператор 3 определяет эти значения по формуле P_1=0,25+(1–J_1) · 0,25, где J_1=1, 2, 3. В результате расчеты ведутся при значениях P_1, равных 0,25; 0,50 и 0,75.

Оператор 4 обеспечивает циклический перебор трех значений переменной управления P_2. Оператор 5 определяет эти значения по формуле P_2=0,25+(1 – J_2) · 0,25, где J_2=1, 2, 3. В результате расчеты ведутся при значениях P_2, равных 0,25; 0,50 и 0,75.

Таким образом, всего рассчитывается 9 вариантов сочетаний значений переменных P_1 и P_2.

Оператор 6 обнуляет значения глобальных переменных, к которым относятся:

- суммарное число поступивших заявок 1-го приоритета $NS_{з11}$;
- суммарное число поступивших заявок 2-го приоритета $NS_{з12}$;
- суммарное число обслуженных заявок 1-го приоритета в 1-й фазе 1-го канала $NS_{обс.11}$;
- суммарное число обслуженных заявок 2-го приоритета в 1-й фазе 2-го канала $NS_{обс.12}$;
- суммарное число заявок 1-го приоритета, поступивших во 2-ю фазу 1-го канала $NS_{з21}$;
- суммарное число заявок 2-го приоритета, поступивших во 2-ю фазу 2-го канала, $NS_{з22}$;
- суммарное число обслуженных заявок 1-го приоритета во 2-й фазе 1-го канала, $N_{обс.21}$;

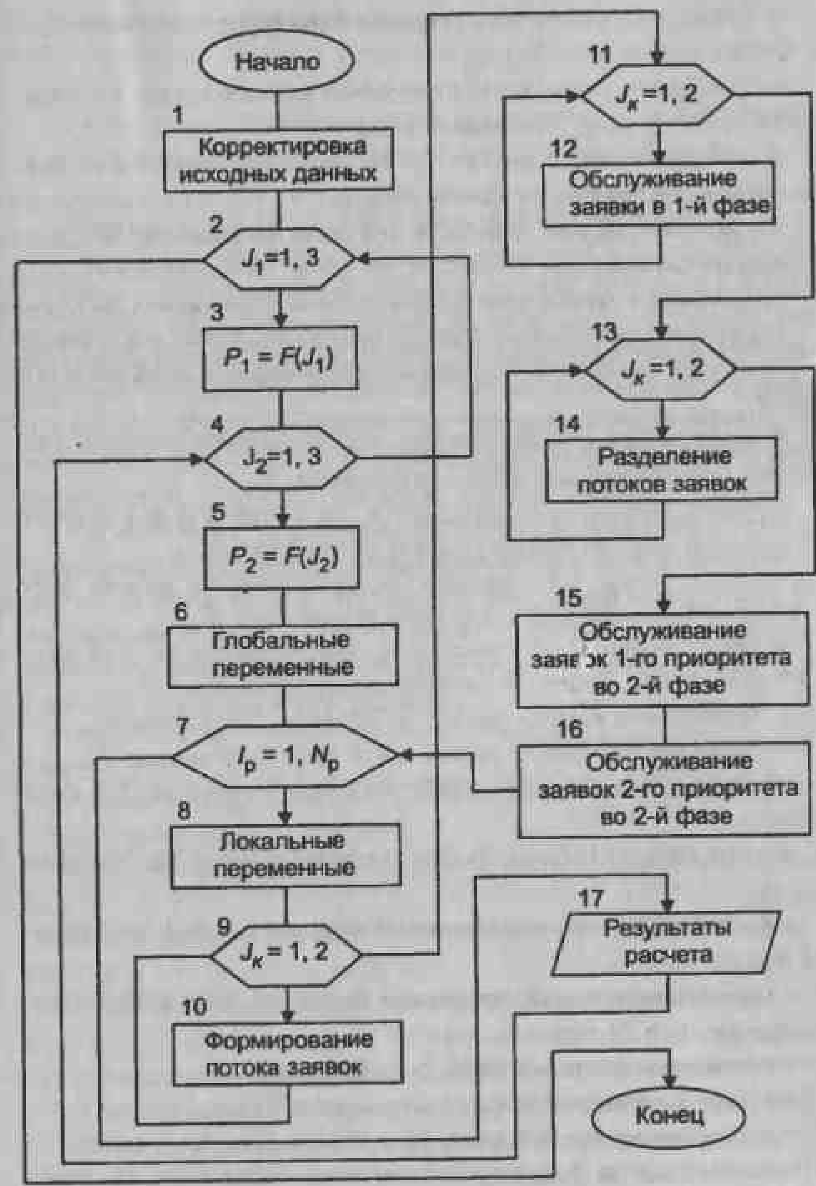


Рис. 9.3. Укрупненная схема алгоритма модели

• суммарное число обслуженных заявок 2-го приоритета во 2-й фазе 1-го канала $N_{обс.22}$;

• суммарное число полностью обслуженных заявок 1-го приоритета в 1-й фазе 1-го канала $NS_{обс.1}$;

• суммарное число полностью обслуженных заявок 2-го приоритета в 1-й фазе 2-го канала $NS_{обс.2}$;

• суммарное время занятости 1-й фазы 1-го канала $SDT_{обс.11}$;

• суммарное время занятости 1-й фазы 2-го канала $SDT_{обс.12}$;

• суммарное время занятости 2-й фазы 1-го канала $SDT_{обс.2}$.

Оператор 7 является заголовком цикла случайных реализаций.

Оператор 8 обнуляет локальные переменные, к которым относятся:

• число заявок 1-го приоритета, поступивших в 1-ю фазу 1-го канала для данной случайной реализации N_{z11};

• число заявок 2-го приоритета, поступивших в 1-ю фазу 2-го канала для данной случайной реализации N_{z12};

• число заявок 1-го приоритета, поступивших во 2-ю фазу 1-го канала для данной случайной реализации N_{z21};

• число заявок 2-го приоритета, поступивших во 2-ю фазу 1-го канала для данной случайной реализации N_{z22};

• число обслуженных заявок в 1-й фазе 1-го канала $N_{обс.11}$;

• число обслуженных заявок в 1-й фазе 2-го канала $N_{обс.12}$;

• число обслуженных заявок 1-го приоритета во 2-й фазе $N_{обс.21}$;

• число обслуженных заявок 2-го приоритета во 2-й фазе $N_{обс.22}$;

• число полностью обслуженных заявок в 1-й фазе 1-го канала $N_{обс.1}$;

• число полностью обслуженных заявок в 1-й фазе 2-го канала $N_{обс.2}$;

• время занятости 1-й фазы 1-го канала $TD_{обс.11}$;

• время занятости 1-й фазы 2-го канала $TD_{обс.12}$;

• время занятости 2-й фазы 1-го канала $TD_{обс.2}$.

Оператор цикла 9 дважды обращается к оператору 10, который вызывает процедуру формирования потоков заявок как процедуру с параметрами:

1) первый раз для формирования потока заявок 1-го приоритета с заданным средним временем между соседними заявками $T_{z.ср.1}$;

2) второй раз для формирования потока заявок 2-го приоритета с заданным средним временем между соседними заявками $T_{z.ср.2}$.

Оператор цикла 11 дважды обращается к оператору 12, который вызывает процедуру обслуживания заявок. Сначала моделируется процесс обслуживания заявок 1-го приоритета в 1-й фазе 1-го канала. Определяются общее количество обслуженных заявок и время окончания обслуживания каждой заявки. Затем аналогичным образом рассчитываются общее количество обслуженных заявок и время окончания обслуживания каждой заявки 2-го приоритета для 1-й фазы 2-го канала. Здесь же определяется суммарное время занятости 1-й фазы каждого канала.

Оператор цикла 13 дважды обращается к оператору 14, который вызывает процедуру разделения потока заявок, обслуженных в 1-й фазе, на два потока. При этом попадание заявки во 2-ю фазу моделируется как событие с заданной вероятностью (равной P_1 – для 1-го канала и P_2 – для 2-го канала). В результате формируются четыре новых потока:

• поток из N_{z21} заявок 1-го приоритета, направляемых во 2-ю фазу;

• поток из $N_{обс.1}$ заявок 1-го приоритета, полностью обслуженных в 1-й фазе 1-го канала;

• поток из N_{z22} заявок 2-го приоритета, направляемых во 2-ю фазу;

• поток из $N_{обс.2}$ заявок 2-го приоритета, полностью обслуженных в 1-й фазе 2-го канала.

Оператор 15 обращается к процедуре обслуживания заявок, т. е. к той же процедуре, которую вызывал оператор 12. Однако на этот раз входными параметрами процедуры являются характеристики части потока заявок 1-го приоритета, направленных во 2-ю фазу.

Оператор 16 обращается к специальной процедуре обслуживания заявок 2-го приоритета, построенной с учетом того, что заявки 1-го приоритета при своем поступлении во 2-ю фазу приостанавливают процесс обслуживания заявок 2-го приоритета.

Оператор 17 обеспечивает вывод результатов расчета одного варианта моделирования на экран.

9.3.2. Алгоритм процедуры формирования заявок

Процедура формирования входных потоков заявок, вызываемая оператором 10 (см. рис. 9.3), имеет ту же структуру, что и процедура формирования заявок в модели бензоколонки (см. п. 4.3). В данной модели эта процедура используется дважды:

1) для формирования потока заявок 1-го приоритета со средним временем между заявками T_{z11};

2) для формирования потока заявок 1-го приоритета со средним временем между заявками T_{z12}.

9.3.3. Алгоритм процедуры обслуживания однородных заявок

Процедура обслуживания однородных заявок, вызываемая операторами 12 и 15 (см. рис. 9.3), имеет ту же структуру, что и процедура обслуживания заявок в модели бензоколонки (см. п. 4.5). В данной модели процедура используется трижды:

1) для моделирования процесса обслуживания заявок 1-го приоритета в 1-й фазе 1-го канала (со средним временем обслуживания заявки $T_{обс.11}$);

2) для моделирования процесса обслуживания заявок 2-го приоритета в 1-й фазе 2-го канала (со средним временем обслуживания заявки $T_{обс.12}$);

3) для моделирования процесса обслуживания заявок 1-го приоритета во 2-й фазе 1-го канала (со средним временем обслуживания заявки $T_{обс.21}$).

9.3.4. Алгоритм процедуры разделения потока заявок

Процедура разделения потока заявок, вызываемая оператором 14 (см. рис. 9.3), используется дважды:

1) для разделения потока заявок 1-го приоритета и определения общего числа заявок, направляемых во 2-ю фазу (N_{z21}), а также для определения времени начала и окончания обслуживания каждой заявки в 1-й фазе 1-го канала;

2) для разделения потока заявок 2-го приоритета и определения общего числа заявок, направляемых во 2-ю фазу (N_{z22}), а также для определения времени начала и окончания обслуживания каждой заявки в 1-й фазе 2-го канала.

Схема алгоритма работы процедуры показана на рис. 9.4.

Оператор 1 обеспечивает поочередное рассмотрение потоков заявок, прошедших через 1-ю фазу 1-го и 2-го каналов.

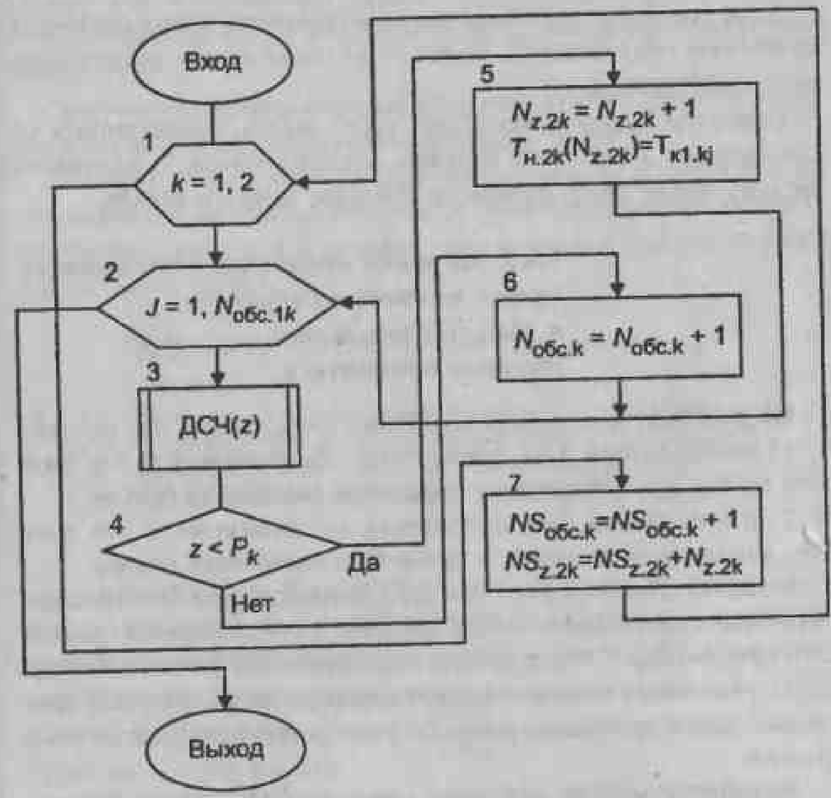


Рис. 9.4. Схема алгоритма процедуры разделения потоков заявок

Оператор 2 производит циклическую обработку заявок, обслуженных в 1-й фазе k-го канала.

Оператор 3 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1).

Относительное количество заявок, направляемых во 2-ю фазу, численно равно P_k. Относительное число случаев, когда случайная величина z будет меньше P_k, также равно P_k. Поэтому выполнение неравенства в операторе 4 является условием попадания текущей заявки во 2-ю фазу. В этом случае оператор 5 определяет число заявок, попавших во 2-ю фазу. Здесь же фиксируется время поступления заявки, равное времени окончания ее обслуживания в 1-й фазе.

Если условие в операторе 4 не выполняется, заявка считается полностью обслуженной. Количество таких заявок подсчитывается оператором 6.

Оператор 7 определяет общее число заявок, направленных во 2-ю фазу из 1-го и 2-го каналов, а также заявок, покинувших систему после обслуживания в 1-й фазе каждого канала.

9.3.5. Алгоритм процедуры обслуживания заявок низшего приоритета в присутствии заявок высшего приоритета

Во 2-ю фазу 1-го канала поступают параллельно два потока:

1) поток заявок 1-го приоритета, обслуженных в 1-й фазе 1-го канала и отобранных в процедуре разделения потока;

2) поток заявок 2-го приоритета, обслуженных в 1-й фазе 2-го канала и отобранных в процедуре разделения потока.

В соответствии с принятым допущением заявка 1-го приоритета приостанавливает обслуживание ранее принятой заявки 2-го приоритета. С точки зрения классификации систем массового обслуживания в данной модели реализуется абсолютный приоритет заявок высокого уровня по отношению к заявкам низкого уровня.

Разработку модели для этого варианта СМО можно построить на основе следующих положений:

1) поток заявок 1-го приоритета, обслуженных в 1-й фазе 1-го канала и направленных во 2-ю фазу, рассматривается в качестве входного потока заявок для 2-й фазы;

2) с помощью процедуры обслуживания однородных заявок моделируется процесс обслуживания заявок 1-го приоритета (без учета заявок 2-го приоритета);

3) поток заявок 2-го приоритета, обслуженных в 1-й фазе 2-го канала и направленных во 2-ю фазу, рассматривается в качестве входного потока заявок для 2-й фазы;

4) моделирование процесса обслуживания заявок 2-го приоритета производится в условиях, когда на временной оси располагаются уже обслуженные заявки 1-го приоритета, т. е. заявки 2-го приоритета могут занимать на этой оси только свободные промежутки.

При взаимодействии заявок двух различных приоритетов могут возникнуть три возможные ситуации (см. п. 1.9.4).

В ситуации 1 ни одна из N_{z21} заявок первого приоритета не препятствует обслуживанию рассматриваемой заявки второго приоритета. Логическое условие, при котором создается 1-я ситуация, записывается так:

$$[T_{кj}(2) <= T_{нi}(1)]\ OR\ [T_{кi}(1) <= T_{нj}(2)]\ (i=1,N_{z21}),$$

где $T_{нi}(1)$ – время начала обслуживания i-й заявки 1-го приоритета;
$T_{кi}(1)$ – время окончания обслуживания i-й заявки 1-го приоритета;
$T_{нj}(2)$ – время начала обслуживания j-й заявки 2-го приоритета;
$T_{кj}(2)$ – время окончания обслуживания j-й заявки 2-го приоритета.

В ситуации 2 система приняла к обслуживанию заявку 2-го приоритета, и она начала обслуживаться. Однако до окончания расчетного времени окончания обслуживания поступила заявка первого приоритета, которая вытесняет данную заявку второго приоритета. Логическое условие, при котором создается 2-я ситуация, записывается так:

$$[T_{нj}(2) < T_{нi}(1)]\ AND\ [T_{нi}(1) < T_{кj}(2)],\ (i=1,N_{z21}), \qquad (9.1)$$

Если условие (9.1) выполняется хотя бы для какой-либо комбинации значений переменных $T_{нj}(2)$, $T_{кj}(2)$, $T_{нi}(1)$ и $T_{кi}(1)$ ($i=1,N_{z21}$), то продолжение процесса обслуживания заявки второго приоритета откладывается до момента освобождения канала.

В ситуации 3 заявка второго приоритета поступила в период обслуживания заявки первого приоритета. Следовательно, заявка второго приоритета не может быть принята к обслуживанию. Логическое условие, при котором создается любой из вариантов 3-й ситуации, записывается так:

$$[T_{нi}(1) < T_{нj}(2)]\ AND\ [T_{нj}(2) < T_{кi}(1)]\ (i=1,N_{z21}). \qquad (9.2)$$

Если условие (9.2) выполняется хотя бы для какой-либо комбинации значений переменных $T_{нj}(2)$, $T_{кj}(2)$, $T_{нi}(1)$ и $T_{кi}(1)$ ($i=1,N_{z21}$), то производится «сдвиг» времени начала и окончания обслуживания заявки 2-го приоритета по формулам:

$$T_{нj} = T_{к.fix};$$

$$T_{кj}(2) = T_{к.fix} + [T_{кj}(2) - T_{нj}(2)],$$

где $T_{к.fix}$ – фиксированное время окончания обслуживания одной из заявок 1-го приоритета, которая помешала принятию к обслуживанию данной заявки 2-го приоритета.

После этого вновь рассматривается возникшая ситуация для «сдвинутой» заявки.

«Дообслуживаемая», или «сдвинутая» заявка, в свою очередь, может оказаться в одной из трех перечисленных выше возможных ситуаций.

Анализ возникающих ситуаций производится в специальной процедуре, схема которой показана на рис. 9.5.

Оператор 1 обнуляет числовые признаки F_1, F_2 и F_3.

Оператор 2 является заголовком циклического перебора всех заявок 1-го приоритета, число которых равно M.

Оператор 3 проверяет условие возникновения ситуации 2, описываемое выражением (9.1). Если это условие выполняется, то оператор 4 присваивает числовому признаку F_2 значение «единица», а также фиксирует время начала и окончания обслуживания той заявки 1-го приоритета, которая вызвала прерывание процесса обслуживания рассматриваемой заявки 2-го приоритета. После этого происходит выход из процедуры.

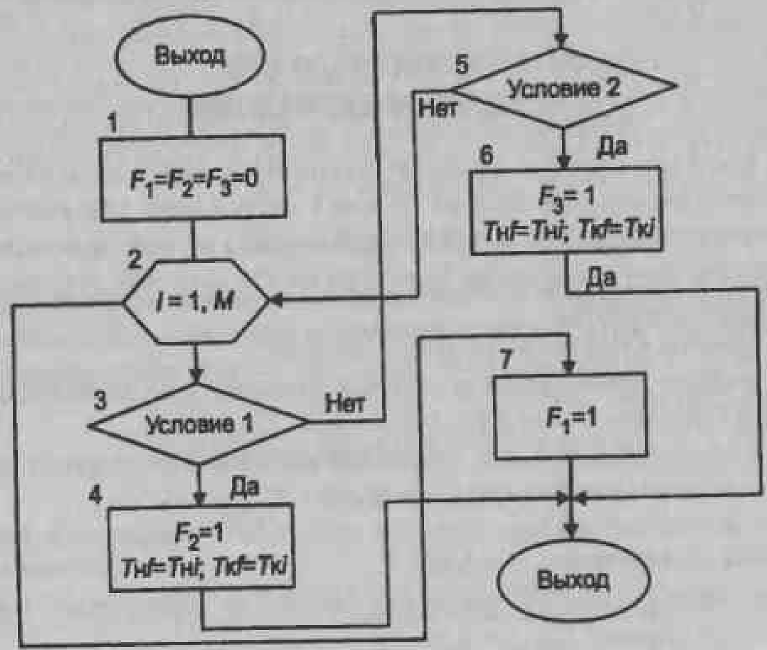


Рис. 9.5. Схема алгоритма процедуры анализа ситуаций

Если условие (9.1) не выполняется, то оператор 5 проверяет условие возникновения ситуации 3, описываемое выражением (9.2). Если это условие выполняется, то оператор 6 присваивает числовому признаку F_3 значение «единица», а также фиксирует время начала и окончания обслуживания той заявки 1-го приоритета, которая обслуживалась в момент появления рассматриваемой заявки 2-го приоритета. После этого происходит выход из процедуры.

Если ни одна из заявок 1-го приоритета не «встретится» с рассматриваемой заявкой 2-го приоритета, то после окончания цикла оператор 7 присваивает числовому признаку F_1 значение «единица», после чего происходит выход из процедуры.

Выработанные в процедуре числовые признаки используются в дальнейшем для корректировки времени начала или продолжения обслуживания рассматриваемой заявки 2-го приоритета.

9.4. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ

Поставим задачу: с помощью алгоритмической модели определить условия, при которых во время работы звена управления учреждением степень загруженности каждого из трех должностных лиц отличалась бы не более чем на 1%, т. е. была бы практически одинакова.

Примем следующие исходные данные:

среднее время между соседними заявками 1-го приоритета, поступающими в 1-ю фазу 1-го канала, $T_{z.\text{ср}.1} = 1$ ч;

• среднее время между соседними заявками 2-го приоритета, поступающими в 1-ю фазу 2-го канала, $T_{z.\text{ср}.2} = 1$ ч;

• среднее время обслуживания заявок 1-го приоритета в 1-й фазе 1-го канала $T_{\text{обс}.11} = 1$ ч;

• среднее время обслуживания заявок 2-го приоритета в 1-й фазе 2-го канала $T_{\text{обс}.12} = 1$ ч;

• среднее время обслуживания заявок 1-го приоритета во 2-й фазе 1-го канала $T_{\text{обс}.21} = 1$ ч;

• среднее время обслуживания заявок 2-го приоритета во 2-й фазе 1-го канала $T_{\text{обс}.22} = 1$ ч;

• период функционирования звена управления $T_{fin} = 10$ ч;

• число случайных реализаций $N_{\text{р}} = 5000$.

В качестве управляющих переменных используем следующие величины:

1) относительную долю заявок, поступающих из 1-й фазы 1-го канала во 2-ю фазу 1-го канала, P_1;

2) относительную долю заявок, поступающих из 1-й фазы 2-го канала во 2-ю фазу 1-го канала, P_2.

Решение задачи проведем в три этапа, постепенно сужая диапазон изменения управляющих переменных.

9.4.1. Первый этап

Проведем вначале расчеты в широком диапазоне изменения управляющих параметров. За исходную точку в пространстве значений управляющих параметров примем точку с координатами: P_1=0,50 и P_2=0,50.Выберем шаг изменения параметров ΔP=0,25. Изменение значений параметров будем производить в обе стороны от начальной точки. Тогда расчетные параметры будут равны:

$$P_1 = 0{,}25;\ 0{,}50;\ 0{,}75;$$

$$P_2 = 0{,}25;\ 0{,}50;\ 0{,}75.$$

Проведем 9 вариантов расчетов при всех возможных сочетаниях значений выбранных параметров. Результаты расчетов представлены в табл. 9.1.

Таблица 9.1

Относительное время занятости (1-й этап)

№ варианта	P_1	P_2	Относительное время занятости, %		
			1-я фаза 1-го канала	1-я фаза 2-го канала	2-я фаза 1-го канала
1	0,25	0,25	0,61	0,61	0,25
2		0,50	0,61	0,61	0,35
3		0,75	0,61	0,61	0,45
4	0,50	0,25	0,61	0,61	0,36
5		0,50	0,61	0,61	0,44
6		0,75	0,61	0,61	0,53
7	0,75	0,25	0,61	0,61	0,45
8		0,50	0,61	0,61	0,53
9		0,75	0,61	0,61	0,60

Анализ данных, помещенных в таблицу, показывает, что первые фазы обоих каналов загружены одинаково, а загрузка 2-й фазы 1-го канала изменяется и достигает практического совпадения с загрузкой первых фаз при сочетании значений управляющих параметров: P_1=0,75 и P_2=0,75. Эту расчетную точку примем за исходную при проведении второго этапа расчетов.

9.4.2. Второй этап

За исходную точку в пространстве значений управляющих параметров примем точку с координатами: P_1=0,75 и P_2=0,75. Выберем шаг изменения параметров ΔP=0,15. Изменение значений параметров будем производить в обе стороны от начальной точки. Тогда расчетные параметры будут равны:

$$P_1 = 0{,}60;\ \ 0{,}75;\ \ 0{,}90;$$

$$P_2 = 0{,}60;\ \ 0{,}75;\ \ 0{,}90.$$

Проведем 9 вариантов расчетов при всех возможных сочетаниях значений выбранных параметров. Результаты расчетов представлены в табл. 9.2.

Таблица 9.2

Относительное время занятости (2-й этап)

№ варианта	P_1	P_2	Относительное время занятости, %: 1-я фаза 1-го канала	1-я фаза 2-го канала	2-я фаза 1-го канала
1	0,60	0,60	0,61	0,61	0,51
2		0,75	0,61	0,61	0,56
3		0,90	0,61	0,61	0,60
4	0,75	0,60	0,61	0,61	0,56
5		0,75	0,62	0,61	0,60
6		0,90	0,61	0,61	0,63
7	0,90	0,60	0,61	0,61	0,60
8		0,75	0,61	0,61	0,63
9		0,90	0,61	0,60	0,66

Анализ данных, помещенных в таблицу, показывает, что первые фазы обоих каналов загружены одинаково, а загрузка 2-й фазы 1-го канала изменяется и достигает практического совпадения с загрузкой первых фаз при сочетании значений управляющих параметров: P_1=0,75 и P_2=0,75. Эту расчетную точку примем за исходную при проведении третьего этапа расчетов.

9.4.3. Третий этап

За исходную точку в пространстве значений управляющих параметров примем точку с координатами: P_1=0,75 и P_2=0,75. Выберем шаг изменения параметров ΔP=0,02. Изменение значений параметров будем производить в обе стороны от начальной точки. Тогда расчетные параметры будут равны:

$$P_1 = 0{,}73;\ \ 0{,}75;\ \ 0{,}77;$$

$$P_2 = 0{,}73;\ \ 0{,}75;\ \ 0{,}77.$$

Проведем 9 вариантов расчетов при всех возможных сочетаниях значений выбранных параметров. Результаты расчетов представлены в табл. 9.3.

Таблица 9.3

Относительное время занятости (3-й этап)

№ варианта	P_1	P_2	Относительное время занятости, %: 1-я фаза 1-го канала	1-я фаза 2-го канала	2-я фаза 1-го канала
1	0,73	0,73	0,61	0,61	0,59
2		0,75	0,61	0,61	0,59
3		0,77	0,61	0,61	0,60
4	0,75	0,73	0,61	0,61	0,59
5		0,75	0,61	0,61	0,60
6		0,77	0,61	0,61	0,60
7	0,77	0,73	0,61	0,61	0,60
8		0,75	0,61	0,62	0,61
9		0,77	0,61	0,61	0,61

Анализ данных, помещенных в таблицу, показывает, что первые фазы обоих каналов загружены одинаково, а загрузка 2-й фазы 1-го канала практически совпадает с загрузкой первых фаз при сочетании значений управляющих параметров: P_1=0,77 и P_2=0,77.

Итак, для равномерной загрузки всех трех должностных лиц в звене управления учреждением необходимо, чтобы количество посетителей каждого потока, которое после рассмотрения дел заместителями направляется для окончательного решения вопроса к руководителю учреждения, составляло 77% от общего числа посетителей.

9.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 9.5.1

В среде Турбо-Паскаля 7.0 создайте исполняемый модуль программы «Модель звена управления», текст которой приведен в приложении 7. Произведите отладку программы с целью ликвидации формальных ошибок.

Задание 9.5.2

Проведите проверку программы расчетом. Подставьте те же исходные данные, которые были выбраны в приведенном выше примере. Убедитесь в том, что результаты расчетов практически совпадают.

Задание 9.5.3

Проведите самостоятельное исследование закономерностей функционирования звена управления с помощью алгоритмической модели. Выберите исходные данные, произведите расчеты и проанализируйте результаты моделирования.

ЧАСТЬ 2 СПРАВОЧНЫЕ СВЕДЕНИЯ ИЗ ТЕОРИИ ВЕРОЯТНОСТЕЙ, МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ И ТЕОРИИ ЭФФЕКТИВНОСТИ

10. СВЕДЕНИЯ ИЗ ТЕОРИИ ВЕРОЯТНОСТЕЙ

10.1. СЛУЧАЙНЫЕ ВЕЛИЧИНЫ

Случайной величиной называется величина, которая в результате опыта может принять некоторое не известное заранее значение. Различают дискретные и непрерывные случайные величины. ***Дискретная случайная величина*** (ДСВ) принимает конечное, или счетное, множество возможных значений. ***Непрерывная случайная величина*** (НСВ) может принимать любые значения из некоторого промежутка (интервала).

Случайная величина задается с помощью ***функции распределения***, представляющей собою вероятность того, что случайная величина X в результате испытания примет значение меньше x, т. е.

$$F(x) = P(X < x).$$

Если функция $F(x)$ непрерывна и дифференцируема (по крайней мере кусочно), то непрерывную случайную величину можно задать с помощью ***плотности распределения вероятностей*** $f(x)$, которая является производной от функции распределения $F(x)$. Таким образом,

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

Свойства функции распределения:

1. Значения функции $F(x)$ при изменении x от $-\infty$ до $+\infty$ лежат в промежутке от 0 до 1, т. е. $0 \le F(x) \le 1$.

2. Функция распределения $F(x)$ есть неубывающая функция, т. е. если $x_2 > x_1$, то $F(x_2) > F(x_1)$.

3. Имеют место равенства: $F(\infty)=1$ и $F(-\infty)=0$.

4. Вероятность попадания случайной величины x в произвольный интервал (a,b) равна: $P(a \le x \le b)=F(b)-F(a)$.

Из свойств функции распределения $F(x)$ вытекают свойства плотности распределения вероятностей:

1. Плотность распределения вероятностей неотрицательна, т. е. $f(x) \ge 0$.

2. Интеграл от плотности распределения по всей оси $0X$ равен единице, т. е.

$$\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1.$$

Таким образом, площадь фигуры под графиком плотности равна единице (рис. 10.1).

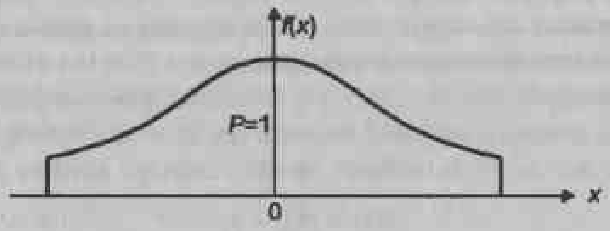


Рис. 10.1. График плотности распределения вероятности

3. Вероятность попадания в интервал (a,b) равна интегралу от плотности распределения в пределах от a до b, т. е.

$$P(a \le x < b) = \int_a^b f(x)dx.$$

10.2. ЧИСЛОВЫЕ ХАРАКТЕРИСТИКИ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

Функция распределения $F(x)$, или плотность распределения $f(x)$ исчерпывающе определяет непрерывную случайную величину. Однако случайная величина может быть задана только несколькими числовыми характеристиками (параметрами), к которым относятся прежде всего ***математическое ожидание*** и ***дисперсия***.

Математическим ожиданием, или средним значением непрерывной случайной величины X, называется число, определяемое по формуле

$$M(X) = \int_{-\infty}^{\infty} xf(x)dx.$$

Дисперсией непрерывной случайной величины X называется математическое ожидание квадрата ее отклонения от среднего значения, т. е.

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

Положительный квадратный корень из дисперсии называется ***средним квадратическим отклонением*** (СКО). Оно определяется по формуле

$$\sigma(X) = +\sqrt{D(X)}.$$

Среди свойств математического ожидания отметим следующие.

1. Математическое ожидание суммы случайных величин равно сумме их математических ожиданий:

$$M(X+Y) = M(X) + M(Y).$$

2. Математическое ожидание произведения независимых случайных величин равно произведению их математических ожиданий:

$$M(XY) = M(X) \cdot M(Y).$$

3. Среднее квадратическое отклонение среднего арифметического для $\sqrt{n}$ одинаково распределенных и взаимно независимых случайных величин в n раз меньше среднего квадратического отклонения для каждой случайной величины, т. е.

$$\sigma[M(X)] = \frac{\sigma(X)}{\sqrt{n}}.$$

Дисперсия обладает следующими свойствами.

1. Дисперсия постоянной величины равна нулю:

$$C = const; \qquad D(C)=0.$$

2. Дисперсия суммы или разности двух независимых случайных величин равна сумме их дисперсий:

$$D(X+Y) = D(X) + D(Y);$$
$$D(X-Y) = D(X) + D(Y).$$

10.3. РАСПРЕДЕЛЕНИЯ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

Наиболее распространенными являются следующие распределения непрерывных случайных величин: равномерное, показательное (экспоненциальное), нормальное, усеченное нормальное.

10.3.1. Равномерное распределение

Непрерывная случайная величина z, принимающая значения в интервале (a,b), имеет равномерное распределение, если плотность распределения имеет вид:

$$f(z) = \begin{cases} 0 & \text{при } z \le a; \\ \dfrac{1}{b-a} & \text{при } a < z \le b; \\ 0 & \text{при } z > b. \end{cases}$$

Функция распределения случайной величины z имеет вид:

$$F(z) = \begin{cases} 0 & \text{при } z \le a; \\ \dfrac{z-a}{b-a} & \text{при } a < z \le b; \\ 1 & \text{при } z > b. \end{cases}$$

Графики функций $f(z)$ и $F(z)$ равномерного распределения в интервале (a,b) показаны на рис. 10.2.

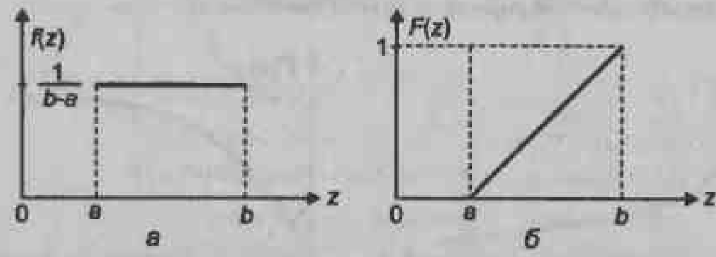


Рис. 10.2. Графики функций $f(z)$ и $F(z)$ равномерного распределения: a – плотность распределения; $б$ – функция распределения

Числовые характеристики случайной величины z, равномерно распределенной в интервале (a,b), имеют следующие значения:

$$M(z) = (a+b)/2; \quad \sigma(z) = \frac{b-a}{\sqrt{12}}.$$

10.3.2. Показательное распределение

Непрерывная случайная величина x, принимающая неотрицательные значения в полубесконечном интервале $(0,\infty)$, имеет показательное распределение, если плотность распределения имеет вид:

$$f(x) = \lambda e^{-\lambda x}.$$

Функция распределения в этом случае имеет вид:

$$F(x) = \int_{-\infty}^{x} f(x)dx = 1 - e^{-\lambda x}.$$

Графики функций $f(x)$ и $F(x)$ приведены на рис. 10.3.

Числовые характеристики показательного распределения определяются по следующим формулам:

$$M(X) = 1/\lambda; \quad D(X) = 1/\lambda^2; \quad \sigma(X) = 1/\lambda.$$

Обращает на себя внимание то обстоятельство, что математическое ожидание и среднее квадратическое отклонение численно равны. Следовательно, показательное распределение имеет только один параметр – интенсивность λ.

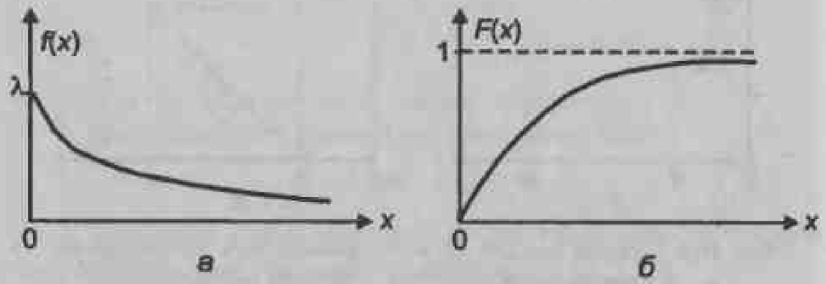


Рис. 10.3. Графики показательного распределения:
а – плотность распределения; *б* – функция распределения

10.3.3. Нормальное распределение

Нормальным называют распределение непрерывной случайной величины, которая имеет плотность

$$f(y) = \frac{1}{\sigma_y\sqrt{2\pi}} EXP[-\frac{(y-m_y)^2}{2\sigma_y^2}],$$

где m_y – математическое ожидание случайной величины y;
σ_y – среднее квадратическое отклонение случайной величины y.

Функция распределения в этом случае определяется по следующей формуле:

$$F(y) = \frac{1}{\sigma_y\sqrt{2\pi}} \int_{-\infty}^{y} EXP[-\frac{(y-m_y)^2}{2\sigma_y^2}]dy.$$

Графики функций $f(y)$ и $F(y)$ для нормального распределения приведены на рис. 10.4.

Введем нормированную и центрированную случайную величину с нормальным распределением:

$$t = \frac{y-M_y}{\sigma_y}.$$

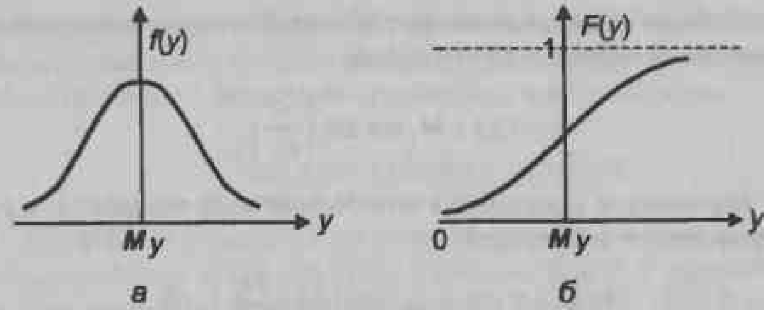


Рис. 10.4. Графики функций нормального распределения:
а – плотность распределения; *б* – функция распределения

Для нее составлена табличная функция Лапласа, имеющая вид:

$$\Phi_0(t) = \frac{1}{\sqrt{2\pi}} \int_0^t EXP(-t^2/2)dt.$$

С помощью табличной функции Лапласа можно определить вероятность попадания случайной величины y в заданный интервал (a,b) по формуле:

$$P(a \le y < b) = \Phi_0\left(\frac{b-M_y}{\sigma_y}\right) - \Phi_0\left(\frac{a-M_y}{\sigma_y}\right). \qquad (10.1)$$

Часто требуется вычислить вероятность того, что отклонение нормально распределенной случайной величины Y по абсолютной величине меньше заданного числа δ, т. е. требуется найти вероятность выполнения неравенства

$$| Y - M_y | < \delta.$$

Заменим это неравенство равносильным ему двойным неравенством

$$M_y - \delta < Y < M_y + \delta.$$

Тогда по формуле (10.1) получим:

$$P(M_y - \delta < Y < M_y + \delta) = \Phi_0\left(\frac{\delta}{\sigma_y}\right) - \Phi_0\left(-\frac{\delta}{\sigma_y}\right).$$

Функция Лапласа нечетная, это значит, что знак минус можно вынести за скобки. Следовательно,

$$P(|Y - M_y|) = 2\Phi_0\left(\frac{\delta}{\sigma_y}\right).$$

Вероятность попадания в полубесконечный интервал $(-\infty, y_\alpha)$ определяется по формуле

$$P(-\infty < Y < y_\alpha) = \alpha = \Phi_0\left(\frac{y_\alpha - M_y}{\sigma_y}\right) + 0{,}5.$$

Зависимость между вероятностью попадания в полубесконечный интервал α и значением квантиля $K_\alpha = (y_\alpha - M_y)/\sigma_y$ приведена в табл. 10.1

Таблица 10.1

K_α	–1,645	–1,282	–0,674	0	0,674	1,282	1,645
P_α	0,05	0,10	0,25	0,50	0,75	0,90	0,95

10.3.4. Усеченное нормальное распределение

Усеченное нормальное распределение непрерывной случайной величины x задается четырьмя параметрами: математическим ожиданием M_x, средним квадратическим отклонением σ_x, а также максимальными и минимальными значениями x_1 и x_2 (точками усечения).

Функция распределения случайной величины x определяется равенством

$$F(x) = \begin{cases} 0 & \text{при } X \le x_1; \\ A[\,\Phi_0(t) - \Phi_0(t_1)\,] & \text{при } x_1 \le X \le x_2; \\ 1 & \text{при } X > x_2, \end{cases}$$

где $$A = \frac{1}{\Phi_0(t_2) - \Phi_0(t_1)};$$

$$t = \frac{x - M_x}{\sigma_x}; \quad t_1 = \frac{x_1 - M_x}{\sigma_x}; \quad t_2 = \frac{x_2 - M_x}{\sigma_x}.$$

Существуют также формулы для расчета математического ожидания, дисперсии и среднего квадратического отклонения случайной величины x с усеченным нормальным распределением.

10.4. СЛУЧАЙНЫЕ СОБЫТИЯ

Случайным называется событие, которое при определенной совокупности условий (во время испытаний) может произойти или не произойти. Каждому событию из множества возможных соответствует вероятность события.

Вероятность достоверного события, которое должно обязательно произойти, равна единице. Вероятность невозможного события равна нулю. Вероятность любого случайного события есть положительное число, заключенное между нулем и единицей.

События называются ***несовместными***, если появление одного из них исключает появление других событий в одном и том же испытании. События называются ***независимыми***, если появление одного события не изменяет вероятности появления другого события.

Несколько событий образуют ***полную группу***, если в результате испытаний появится хотя бы одно из них. Если при этом события ***попарно несовместны***, то в результате испытаний появится только одно из них.

10.5. ПОТОКИ СОБЫТИЙ

Потоком событий называется последовательность событий, которые наступают в случайные моменты времени. Поток называется ***однородным***, если он характеризуется только моментами наступления событий $t_1, t_2, \ldots, t_n$, $(t_i \le t_{i+1})$. Поток ***неоднородных событий*** характеризуется моментами времени наступления событий и набором признаков $\psi_1, \psi_2, \ldots, \psi_n$. К числу признаков может, например, относиться приоритет заявки.

Поток событий может обладать свойством ***стационарности***, которое заключается в том, что вероятность появления K-событий на любом промежутке времени зависит только от числа K и от длительности промежутка времени и не зависит от положения промежутка на оси времени.

Поток событий может обладать свойством ***отсутствия последействия***, если вероятность появления К событий на любом промежутке времени не зависит от предыстории, т. е. от того, появлялись ли события в предыдущие моменты времени.

Поток событий может обладать свойством ***ординарности***, если появление двух и более событий за малый промежуток времени практически невозможно.

Если поток событий обладает свойствами стационарности, отсутствия последействия и ординарности, то его называют ***простейшим потоком***.

Интенсивностью потока называется среднее число событий, которые появляются в единицу времени. Для простейшего потока время между двумя соседними событиями является случайной величиной с показательным распределением. Его можно задать функцией распределения

$$F(\tau) = 1 - e^{-\lambda\tau},$$

где τ – случайная величина времени между соседними событиями;
λ – параметр распределения (интенсивность потока).

Математическое ожидание времени между соседними событиями $M(\tau)=1/\lambda$. Среднее квадратическое отклонение времени между соседними событиями $\sigma(t)=1/\lambda$.

Пусть поток событий представляет собой композицию (сумму) двух простейших потоков, имеющих показательные распределения с плотностями:

$$f(\tau_a) = \lambda_a\ exp(-\lambda_a\tau_a);$$

$$f(\tau_b) = \lambda_b\ exp(-\lambda_b\tau_b).$$

При этом события a и b следуют друг за другом чередуясь (рис.10.5).

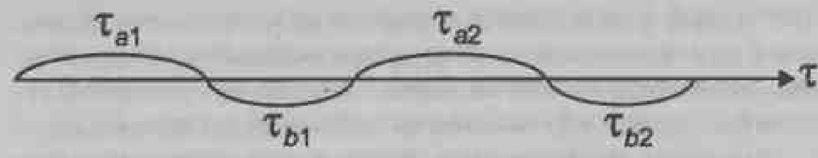


Рис. 10.5. Композиция потоков событий

Тогда результирующий поток будет представлять собой поток Эрланга 2-го порядка с плотностью распределения, определяемой по формуле

$$f(\tau) = \frac{\lambda_a\lambda_b(\exp(-\lambda_a\tau) - \exp(-\lambda_b\tau))}{\lambda_b - \lambda_a}.$$

В частном случае при $\lambda_a = \lambda_b = \lambda$ плотность распределения потока Эрланга определяется формулой

$$f(t) = \lambda^2\ \tau\ (exp(-\lambda\tau)).$$

10.6. ЦЕНТРАЛЬНАЯ ПРЕДЕЛЬНАЯ ТЕОРЕМА ТЕОРИИ ВЕРОЯТНОСТЕЙ

Центральная предельная теорема теории вероятностей (***теорема Ляпунова***) содержит доказательства того, что если случайные величины $x_1, x_2, \dots x_n$ независимы, одинаково распределены и имеют конечные математические ожидания $M(x_i)$ и дисперсии $s(x_i)$, то распределение суммы этих случайных величин при неограниченном увеличении n неограниченно приближается к нормальному распределению.

На практике теорему Ляпунова можно использовать уже при $n > 10$.

11. СВЕДЕНИЯ ИЗ МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ

Математическая статистика используется для обработки результатов испытаний (статистических данных) методами теории вероятностей. К задачам математической статистики относятся:

1) оценка неизвестной вероятности события; оценка неизвестной функции распределения; оценка параметров распределения, вид которого известен; оценка зависимости случайной величины от одной или нескольких случайных величин и др.;

2) проверка статистических гипотез о виде неизвестного распределения или величине параметров распределения, вид которого известен.

Для изучения совокупности однородных объектов или явлений в математической статистике чаще всего используется выборочное исследование. ***Выборкой*** называется совокупность случайно отобранных объектов. Множество всех объектов, из которых производится выборка, называют ***генеральной совокупностью.***

11.1. СТАТИСТИЧЕСКИЕ ОЦЕНКИ ПАРАМЕТРОВ РАСПРЕДЕЛЕНИЯ

Генеральным средним $\overline{X}_г$ называют среднее арифметическое значений некоторого признака генеральной совокупности. Если все значения $x_1, x_2, \dots, x_N$ различны, то

$$\overline{X}_г = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

где N – объем генеральной совокупности.

Выборочным средним $\overline{X}_в$ называют среднее арифметическое значений признака выборочной совокупности. Если все значения $x_1, x_2, \dots, x_n$ различны, то

$$\overline{X}_в = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

где n – объем выборки.

Выборочное среднее является случайной величиной. Его математическое ожидание равно генеральному среднему, т. е.

$$M(\overline{X}_в) = \overline{X}_г,$$

а это в свою очередь означает, что выборочное среднее является ***несмещенной оценкой*** генерального среднего.

Генеральной дисперсией $D_г$ называют среднее арифметическое квадратов отклонений значений признака генеральной совокупности от генерального среднего. Если все значения $x_1, x_2, \dots, x_N$ различны, то

$$D_г = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{X}_г)^2.$$

Генеральным средним квадратическим отклонением (стандартом) называется квадратный корень из генеральной дисперсии:

$$\sigma_г = \sqrt{D_г}.$$

Выборочной дисперсией $D_в$ называют среднее арифметическое квадратов отклонений наблюдаемых значений признака от выборочного среднего. Если все значения $x_1, x_2, \dots, x_n$ различны, то

$$D_в = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{X}_в)^2.$$

Математическое ожидание выборочной дисперсии:

$$M(D_в) = \frac{n-1}{n} D_г,$$

т. е. выборочная дисперсия не является несмещенной оценкой генеральной дисперсии. Поэтому вместо выборочной дисперсии обычно рассматривают ***исправленную дисперсию:***

$$S^2 = \frac{n}{n-1} D_в = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{X}_в)^2,$$

которая является ***несмещенной*** оценкой генеральной дисперсии.

Выборочное среднее квадратическое отклонение определяется по формуле

$$S_в = \sqrt{\frac{1}{n-1}\left[\sum_{i=1}^{n}(x_i - \overline{X}_в)^2\right]}. \qquad (11.1)$$

Однако вычисление $S_в$ по формуле (11.1) неудобно, так как $\overline{X}_в$ вычисляется в процессе накопления x_i, а это требует запоминания всех n значений x_i.

Поэтому при проведении расчетов на ЭВМ используется преобразованная формула для среднего квадратического отклонения следующего вида:

$$S_в = \sqrt{\frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - n\,\overline{X}_i^2\right)}.$$

11.2. ОПРЕДЕЛЕНИЕ ТРЕБУЕМОГО ОБЪЕМА ВЫБОРКИ

Найденная по данным выборки величина $\overline{X}_в$ представляет собой статистическую оценку неизвестного параметра $\overline{X}_г$.

Доверительной вероятностью (надежностью) называют вероятность α того, что абсолютная величина отклонения оценки от истинного значения не превышает некоторой заданной характеристики точности γ, т. е.

$$\alpha = P(|\overline{X}_г - \overline{X}_в| < \gamma).$$

Иначе говоря, α – есть вероятность того, что интервал ($\overline{X}_в - \gamma$, $\overline{X}_в + \gamma$) заключает в себе истинное значение параметра $\overline{X}_г$. Этот интервал называют ***доверительным.***

Если случайная величина X имеет нормальное распределение, то для определения математического ожидания выборочного

среднего и среднего квадратического отклонения справедливы соотношения:

$$M(\overline{X}_в) = \overline{X}_г;$$

$$\sigma(\overline{X}_в) = \frac{\sigma_г}{\sqrt{n}}.$$

Будем считать, что среднее квадратическое отклонение $\sigma_г$ известно. Для упрощения зависимостей будем в дальнейшем использовать ее без индекса σ. Тогда вероятность того, что истинное математическое ожидание случайной величины X попадет в заданный интервал, будет равна:

$$\alpha = P(|\,\overline{X}_г - \overline{X}_в| < \gamma) = P\left(\left|\frac{\overline{X}_г - \overline{X}_в}{\sigma/\sqrt{n}}\right| < \frac{\gamma}{\sigma/\sqrt{n}}\right) = 2\Phi_0\left(\frac{\gamma\sqrt{n}}{\sigma}\right).$$

Отсюда следует, что

$$\frac{\gamma\sqrt{n}}{\sigma} = \Phi_0^{-1}\left(\frac{\alpha}{2}\right),$$

где Φ_0^{-1} – обратная функция Лапласа.

Тогда минимальный объем выборки (число испытаний), который обеспечивает оценку математического ожидания с заданной точностью γ и надежностью α, определяется по формуле

$$n = \frac{\sigma^2}{\gamma^2}\left[\Phi_0^{-1}\left(\frac{\alpha}{2}\right)^2\right] = \frac{\sigma^2}{\gamma^2} K_{\alpha/2}^2.$$

Для наиболее часто используемых уровней надежности α значения коэффициента $K_{\alpha/2}$ приведены в табл. 11.1.

Таблица 11.1

$K_{\alpha/2}$	0,900	0,950	0,975	0,990	0,999
α	1,64	1,96	2,24	2,58	3,29

12. СВЕДЕНИЯ ИЗ ТЕОРИИ ЭФФЕКТИВНОСТИ

12.1. ПОНЯТИЕ ЭФФЕКТИВНОСТИ

Экономические системы создаются в результате целенаправленной деятельности человека для удовлетворения его потребностей. Созданная человеком система используется им путем выполнения некоторых действий или проведения операций. Каждая операция объединена каким-либо общим замыслом или целью.

Реальный, или фактический, результат операции может отличаться от желаемого (ожидаемого или требуемого) результата. Например, фактическая прибыль может отличаться от ожидаемой прибыли.

Эффективность операции с экономической системой – это степень соответствия реального результата желаемому.

На эффективность влияют свойства системы (полезные с точки зрения целевого назначения системы), условия ее функционирования и способы управления системой.

К основным свойствам экономической системы можно отнести устойчивость к воздействию неблагоприятных факторов (стабильность) и управляемость системой.

К условиям функционирования экономической системы можно отнести емкость рынка, наличие конкуренции, обеспеченность сырьевыми ресурсами и т. п.

Способы управления системой могут включать распределение выделенных средств (структура системы), последовательность проведения операций по управлению системой, способ использования резервов.

12.2. ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ

Показатель эффективности – это мера степени соответствия реального результата экономической операции требуемому результату.

На практике встречаются пять ситуаций, в каждой из которых используется свой показатель эффективности.

Ситуация 1. Желаемый результат операции состоит в наступлении некоторого события. Показатель эффективности определяется так:

$$W(u) = P(A),$$

где $P(A)$ – вероятность события A;
u – способ проведения операции (из множества возможных).

Примером такой ситуации может быть случай, когда вкладываются денежные средства в приобретение билетов выигрышной лотереи с одним крупным призом. Показателем эффективности является вероятность выигрыша.

Ситуация 2. Желаемый результат не определен или экстремален. Показатель эффективности – средний результат. Он определяется по формуле

$$W(u) = M[y(u)],$$

где $y(u)$ – случайный результат для u-го способа проведения операции;
$M[y(u)]$ – математическое ожидание (среднее значение) случайного результата.

Примером такой ситуации может быть случай, когда в результате проведения многократной реализации продукции получается случайная прибыль. Эффективность операции оценивается по средней прибыли.

Ситуация 3. Желаемый результат– достижение требуемого результата – $y_{тр}$. Показатель эффективности – вероятность достижения требуемого результата, определяется по формуле

$$W(u) = P[y(u) \geq y_{тр}] = 1 - F_u(y_{тр}),$$

где F_u – функция распределения случайного результата $y(u)$ для u-го способа проведения операции.

Такой показатель принято называть вероятностной гарантией требуемого результата.

Примером такой ситуации может служить случай, когда оценивается вероятность получения прибыли, не меньшей, чем заданная.

Ситуация 4. Желаемый результат – достижение гарантированного минимального результата с заданной вероятностью. Показатель эффективности для этого случая определяется из соотношения

$$\alpha = P(y \geq y_\alpha) = 1 - F(y_\alpha), \qquad (12.1)$$

где α – уровень гарантии (надежность) достижения заранее неизвестного результата y_α.

Из соотношения (12.1) может быть найдена величина y_α по формуле

$$y_\alpha = F^{-1}(1 - \alpha).$$

Обычно предполагается, что результат представляет собой случайную величину с нормальным распределением. Поэтому

$$y_\alpha = M(y) - K_\alpha\, \sigma_y,$$

где K_α – квантиль нормального распределения, определяемый по таблице функции Лапласа;

σ_y – среднее квадратическое отклонение случайного результата.

Примером такой ситуации может быть случай, когда показателем эффективности является величина минимальной прибыли, которая будет получена с заданной вероятностью.

Ситуация 5. Желаемый результат – достижение гарантированного максимального результата с заданной вероятностью. Показатель эффективности для этого случая определяется из соотношения

$$\alpha = P(y < y_\alpha) = F(y_\alpha), \qquad (12.2)$$

где α – уровень гарантии (надежности) достижения заранее неизвестного результата y_α.

Из соотношения (12.2) может быть найдена величина y_α по формуле

$$y_\alpha = F^{-1}(\alpha).$$

Обычно предполагается, что результат представляет собой случайную величину с нормальным распределением. Поэтому

$$y_\alpha = M(y) + K_\alpha\, \sigma_y,$$

где K_α – квантиль нормального распределения, определяемый по таблице функции Лапласа.

Примером такой ситуации может быть случай, когда показателем эффективности является величина максимального убытка, которая будет получена с заданной вероятностью.

12.3. КРИТЕРИИ ЭФФЕКТИВНОСТИ

Если существует несколько вариантов действий, направленных на достижение поставленной цели (способов проведения операции), то можно поставить задачу выбора наилучшего варианта.

В общем случае критерий эффективности есть правило, позволяющее сопоставлять способы достижения цели и осуществлять направленный выбор наилучшего способа из множества возможных.

Существуют две концепции, на основе которых производится построение критериев эффективности: ***пригодность*** и ***оптимальность***.

В рамках концепции пригодности рекомендуется выбор стратегии по следующим критериям.

1. Критерий приемлемого результата. Правило выбора наилучшего способа проведения операции состоит в том, что выбирается один из возможных вариантов действий, для которого выполняется условие: величина показателя эффективности W не меньше заданной $W_{тр}$, т. е.

$$K(u^*) : W(u) \geq W_{тр} \quad (u \in U),$$

где u – порядковый номер способа действия;

u^* – номер оптимального способа действия;

U – множество возможных способов действия.

Если существует несколько способов проведения операций, каждый из которых приводит к выполнению приведенного выше условия, то в соответствии с данным критерием эффективности любой из этих способов является приемлемым.

2. Критерий допустимой гарантии. Правило выбора наилучшего способа действий заключается в том, что выбирается один из возможных вариантов, для которого выполняется условие: вероятность того, что величина показателя эффективности окажется не меньше заданной, превышает некоторое заданное значение уровня гарантии, т. е.

$$K(u^*) : P[W(u) \geq W_{тр}\,] \geq P_{тр} \;\; (u \in U),$$

Если существует несколько способов проведения операций, каждый из которых приводит к выполнению приведенного выше условия, то в соответствии с данным критерием эффективности любой из этих способов является приемлемым.

3. Критерий допустимого гарантированного результата. Правило выбора наилучшего способа действий заключается в том, что выбирается один из возможных вариантов, для которого выполняется условие: гарантированная величина показателя эффективности не меньше заданного значения, т. е.

$$K(u^*) : W_{\alpha}(u) \;\; \geq W_{тр} \;\; (u \in U)$$

где $W_{\alpha}(u) = F^{-1}(1 - \alpha)$.

В рамках концепции оптимизации используются следующие критерии

1. Критерий экстремального результата. Правило выбора наилучшего способа действий заключается в том, что из возможных вариантов выбирается один, для которого показатель эффективности имеет наибольшее (или наименьшее) значение, т. е.

$$K(u^*) : MAX\{W(u)\} \;\; (u \in U),$$

$$\text{или} \quad K(u^*) : MIN\{W(u)\} \;\; (u \in U).$$

Примерами таких критериев могут служить наибольшая случайная прибыль или наименьшие случайные убытки.

2. Критерий экстремального среднего результата. Правило выбора наилучшего способа действий заключается в том, что из возможных вариантов выбирается один, для которого показатель эффективности имеет наибольшее (или наименьшее) среднее значение, т. е.

$$K(u^*) : MAX\{\, M[y(u)]\} \;\; (u \in U)$$

$$\text{или} \quad K(u^*) : MIN\{\, M[y(u)]\} \;\; (u \in U).$$

Примерами таких критериев может служить наибольшая средняя прибыль или наименьший средний убыток.

3. Критерий наибольшей вероятностной гарантии. Правило выбора наилучшего способа действия заключается в том, что из возможных вариантов выбирается один, для которого достигает максимума следующее условие:

$$K(u^*) : MAX\{\, P[W(u) \;\; \geq W_{тр}\,]\} \;\; (u \in U).$$

4. Критерий наибольшего гарантированного результата. Правило выбора наилучшего способа действия заключается в том, что из возможных вариантов выбирается один, для которого достигает максимума следующее условие:

$$K(u^*) \;\; : \;\; MAX\{W_{\alpha}(u)\} \;\; (u \; О \; U),$$

где $W_{\alpha}(u) = F^{-1}\,(1 - \alpha)$.

ЛИТЕРАТУРА

1. **Бусленко Н. П.** Моделирование сложных систем. – М.: Наука, 1978.
2. **Варфоломеев В. И.**Имитационное моделирование экономических систем. -М.: МГУК, 1997
3. **Емельянов А. А., Власов Е. А.** Имитационное моделирование в экономических информационных системах. – М.: МЭСИ, 1996.
4. **Лабскер Л. Г.** и др. Математическое моделирование финансово-экономических ситуаций с применением компьютера. – М.: МЭСИ, 1998.
5. Надежность и эффективность в технике: Справочник 10 т./Ред. совет: В. С. Авдуевский (председатель) и др. – М.: Машиностроение, 1988, Т. 3. Эффективность технических систем / Под общ. ред. В. Ф. Уткина, Ю. В. Крючкова.
6. **Нейлор Т.** Машинные имитационные эксперименты с моделями экономических систем. – М.: Мир, 1975.
7. **Ойхман Е. Г., Попов Э. В.** Реинжиниринг бизнеса. – М.: Финансы и статистика. 1997.
8. **Подиновский В. В.** Теоретические основы выработки решений в сложных ситуациях. – М.:МО СССР, 1978.
9. **Романцев В. В., Яковлев С. А.** Моделирование систем массового обслуживания. – СПб.: Поликом, 1995.
10. **Сабинин О. Ю.** Статистическое моделирование технических систем. – СПб.: Изд. ЭТУ, 1993.
11. **Советов Б. Я., Яковлев С. А.** Моделирование систем. Учебник для вузов. – 2-е изд. перераб. и доп. – М.: Высшая школа, 1998.
12. **Шеннон Р.** Имитационное моделирование систем – искусство и наука. – М.: Мир, 1978.

ПРИЛОЖЕНИЯ

Приложение 1

Программа «паутинообразной» модели на языке Visual Basic 5.0

1. Стартовая форма frmForm1

Макет стартовой формы показан на рис. 3.1, с. 61.

1.1. Таблица свойств

Объект	Свойства	Установки
Форма	Name Caption	frmForm1 Паутинообразная модель
Метка 1	Name Caption	lblInputDat Измените исходные данные и нажмите кнопку <Расчет>
Командная кнопка 1	Name Caption	cmdRashet Расчет
Командная кнопка 2	Name Caption Enabled	CmdGrafic График False
Командная кнопка 3	Name Caption	CmdExit Выход

1.2. Процедуры обработки прерываний

```
Private Sub cmdExit_Click()
    End
End Sub

Private Sub cmdGrafic_Click()
    Line(600,3500)-(6600,3500)                    ' горизонтальная ось
    Line(600,3500)-(600,1500)                     ' вертикальная ось
    My=1000                                       ' масштаб по оси Y
    X1 = 600 : Y1 = 3500 –  P(1) * My
    Dx = 100                                      ' шаг по времени
    S = 0: For j = 1 To Tk: S = S + P(j): Next
    Pcp = S / Tk
```

```
    CurrentX = 500: CurrentY = 1500: Print "P"
    CurrentX = 6500: CurrentY = 3500: Print "T"
    CurrentX = 650: CurrentY = 3300 - Pcp * My
    Print "Pcp"
    Xk = 600 + Tk * Dx: Ycp = 3500 - Pcp * My
    Line (X1, Ycp)-(Xk, Ycp)
    For j = 1 To Tk - 1
        X2 = X1 + Dx: Y2 = 3500 - P(j) * My
        Line (X1, Y1)-(X2, Y2)
        X1 = X2: Y1 = Y2
    Next
    cmdGrafic.Enabled = False
    cmdRashet.Enabled = True
End Sub

Private Sub cmdRashet_Click()
    A = Val(Text1): B = Val(Text2): C = Val(Text3)
    E = Val(Text4): Ro = Val(Text5): Su = Val(Text6)
    Cv = Val(Text7): Sw = Val(Text8)
    Mu = 0: Mv = 0: Mw = 0
    Call Model1
    cmdRashet.Enabled = False: cmdGrafic.Enabled = True
    Cls
End Sub

Private Sub Form_Load()
        frmForm1.Show: Cls
End Sub
```

2. Модуль общего назначения Model1

```
Public Const Tk = 60
Public P(Tk) As Single
Public T As Integer, i As Integer, j As Integer
Public A, B, C, E, Ro, Su, Sv, Sw, Mu, Mv, Mw, Et
Public Sub Model1()
    P(1) = (A - C) / (B + E)            'цена на первом временном отрезке
    P(2) = (A - C) / B - E / B * P(1)   ' цена на втором временном отрезке
    For T = 3 To Tk                      ' цикл временных участков
      ' генерирование трех возможных значений случайной величины
      ' с нормальным распределением :
        For j = 1 To 3                   'цикл временных участков
        Et = 0
        For i = 1 To 12
            Et = Et + Rnd
        Next
```



```
            If j = 1 Then Ut = Mu + (Et - 6) * Su
            If j = 2 Then Vt = Mv + (Et - 6) * Sv
            If j = 3 Then Wt = Mw + (Et - 6) * Sw
        Next
        DP = P(T - 1) - P(T - 2)
        P(T) = (A-C-E*(P(T-1)-Ro*DP) + Ut + Wt - Vt) / B
        If P(T) < 0 Then P(T) = 0
        P(T - 2) = P(T - 1): P(T - 1) = P(T)
    Next
End Sub
```

Приложение 2

Программа модели бензоколонки на языке Visual Basic 5.0

1. Стартовая форма frmForm1

Макет стартовой формы приведен на рис. 4.2, с. 75.

1.1. Таблица свойств формы 1

Объект	Свойства	Установки
Форма 1	Name Caption	frmForm1 Модель СМО с 1, 2 или 3 каналами
Командная кнопка 1	Name Caption	Command1 Расчет
Командная кнопка 2	Name Caption	Command2 Расчет
Командная кнопка 3	Name Caption	Command3 Выход

1.2. Процедуры обработки прерываний

```
Private Sub Command1_Click()
    Nkan = Val(txtNk): TZcp = Val(txtTzs): TObcp = Val(txtTobs)
    TWmax = Val(txtTwm): Nr = Val(txtNp)
    Call Model2
End Sub
```

```
Private Sub Command2_Click()
    frmForm1.txtResult = ""
End Sub

Private Sub Command3_Click()
    End
End Sub
```

2. Форма frmForm2

Ниже приведен макет формы 2.

2.1. Таблица свойств формы 2

Объект	Свойства	Установки
Форма 2	Name Enabled	Form2 False

2.2. Процедура обработки прерываний

```
Private Sub Form2_Load()

End Sub
```

2. Модули общего назначения

```
'------------------------------ Описание констант и массивов: - ----------------

Public Const Tfin = 10                        'время окончания работы
Public Const Nzmax = 40                       'максимальное число заявок
Public Tz(Nzmax)                              'массив времени поступления заявок
Public Nob(3) As Integer                      'число обслуженных заявок в каналах
Public TKO(3)                                 'время окончания обслуживания заявок
Public TZcp, Tobcp, Twmax, Tkmin, TH, TK, z, Ts
Public Snob As Long, Iz As Integer, Nz As Integer, Ir As Integer
Public Nr As Integer, J As Integer, Nkan As Integer, Jmin As Integer
180
Public Sub Model2()
'------------------------------ Главный модуль: - -------------------------------
    SNob = 0                                  'сумматор числа обслуженных заявок
    frmForm1.Enabled = False: frmForm1.Visible = False
    frmForm2.Enabled = True: frmForm2.Visible = True
```



```
    For Ir = 1 To Nr                          'начало цикла случайных реализаций
        frmForm2.Cls                          ' очистка окна формы 2
        frmForm2.CurrentX = 600: frmForm2.CurrentY = 200
        frmForm2.Print "Расчет " & Ir & "-й реализации"     'вывод показаний
                                         ' счетчика числа ' реализаций в окно 'формы 2
        '..................... обнуление локальных переменных: ..........................
        Nz = 0                                ' обнуление числа заявок
            Nob(1) = 0: Nob(2) = 0: Nob(3) = 0' обнуление числа обслуженных
                                                               'заявок
        TKO(1) = 0: TKO(2) = 0: TKO(3) = 0                 ' время окончания
                                   ' обслуживания 'заявок в 1, 2 и 3-м каналах
        Call ZAJAVKA                          ' процедура "Поток заявок"
        For Iz = 1 To Nz                      ' начало цикла обслуживания заявок
            '-------------------выбор номера канала:----------------------------
            TKmin = TKO(1)
            For J = 1 To Nkan
            If TKO(J) < TKmin Then TKmin = TKO(J): Jmin = J
        Next J
        Call SERVICE                          ' процедура обслуживания заявки
    Next Iz                                   ' конец цикла обслуживания заявок
    '                                         суммарное число обслуженных заявок:
    Snob  = Snob + Nob(1) + Nob(2) + Nob(3)
    Next Ir                                   ' конец цикла реализаций
    frmForm2.Enabled = False: frmForm2.Visible = False
    frmForm1.Enabled = True: frmForm1.Visible = True
    '                                         показатель эффективности:
    Cotn = SNob / Nr -  1 + 0.5 * Nkan -  0.5 * Nkan * Nkan
    frmForm1.txtResult = Format$(Cotn, "#.##")
End Sub

Sub ZAJAVKA()
    ' Процедура "Поток заявок"
    T = 0                                     ' модельное время
    For J = 1 To Nzmax'                       ' начало цикла формирования заявок
        z = Rnd(1)          ' случайная величина с равномерным распределением
        Ts = T -  TZcp * Log(z)               ' случайное время поступления заявки
        If Ts > Tfin Then Exit For            ' условие прекращения приема заявок
        Nz = Nz + 1                           ' счетчик числа заявок
        Tz(Nz) = Ts                           ' фиксированное время поступления заявки
        T = Ts                                ' изменение модельного времени
    Next J                                    ' конец цикла формирования заявок
End Sub

Sub SERVICE()
    '.............Процедура "Обслуживание заявок"
    J = Jmin                                  номер канала
```

```
   DTWait = 0                          ' начальное значение времени ожидания
   TH = Tz(Iz)                                     'время начала обслуживания
      If Tz(Iz) < TKO(J) Then     ' проверка необходимости корректировки
      ' корректировка времени начала обслуживания:
      DTWait = TKO(J) - Tz(Iz)                           ' период ожидания
      If DTWait > TWmax Then Exit Sub

      TH = TKO(J)                                 ' время начала обслуживания
   End If
   z = Rnd(1)            'случайная величина с равномерным распределением
                                                         ' в интервале (0,1)
   TK = TH - TObcp * Log(z)                    ' время окончания обслуживания
   If TK > Tfin Then
      TKO(J) = Tfin: Exit Sub
   End If
   Nob(J) = Nob(J) + 1                   ' увеличение числа обслуженных заявок
   TKO(J) = TK                                 ' время окончания обслуживания
End Sub
```

Приложение 3

Программа модели управления запасами на языке Visual Basic 5.0

1. Стартовая форма frmZapasy

Макет стартовой формы приведен на рис. 5.1, с. 89

1.1. Таблица свойств

Объект	Свойства	Установки
Форма 1	Name Caption	frmZapasy Модель управления запасами
Командная кнопка 1	Name Caption	cmdRashet Расчет
Командная кнопка 2	Name Caption	cmdClear Очистка
Командная кнопка 3	Name Caption	cmdExit Выход

1.2. Процедуры обработки прерываний

```
Private Sub cmdRaschet_Click()
   Nur = Val(txtbegUr): Part = Val(txtParty)
   URmin = Val(txtURmn):MTVZ=Val(txtMTdos)
   STVZ=Val(txtSNdos): MDS=Val(txtMDcp)
   SDS=Val(txtSKOD): C1=Val(txtC_1): C2=Val(txtC_2)
   C3=Val(txtC_3): TD=Val(txtTskl): Nr=Val(txtNreal)
   Call Model3
End Sub

Private Sub cmdClear_Click()
   frmZapasy.txtResult = ""
End Sub

Private Sub cmdExit_Click()
   End
End Sub
```

2. Модуль общего назначения Model3.bas

```
Public Sub Model3()
   ' Главный модуль программы "управление запасами"
   MSC = 0                                    ' средние суммарные издержки
   SSC = 0                                      ' СКО суммарных издержек
   TP0 = TD + 1                  ' время выполнения заявки на поставку
   For Ir = 1 To Nr                               ' начало цикла реализаций
      TP=TP0     ' исходное значение времени поступления партии товара
      Sc1=0              ' исходная сумма затрат на хранение товара
      Sc2=0              ' исходная сумма затрат на поставку товара
      Sc3=0     ' исходная сумма затрат, связанных с нехваткой товара
      T=0                             ' счетчик модельного времени, дни
      Zajav=0                  ' признак отсутствия запроса на поставку
      V=Nur                         ' начальный уровень текущего запаса
      Do                                       ' начало цикла DO...LOOP
         T = T + 1                                      'модельное время
         If T > TD Then Exit Do                  ' условие окончания расчета
                                                     ' текущей реализации
         N = Norm    ' обращение к функции, вырабатывающей случайную
                    величину с нормальным распределением – Norm
         D = Int(MDS + N * SDS)                      ' случайный дневной спрос
         If T >= TP Then
            TP = TP0         ' восстановление исходного времени  поставки
            Zajav = 0        'восстановление признака отсутствия заявки
            V = V + Part                  ' увеличение запаса после поставки
         End If
```

```
            V = V - D                     ' уменьшение запаса в результате спроса
            If V < 0 Then                     ' условие появления дефицита
             Sc3 = Sc3 - V * C3      'определение издержек, связанных с
                                                                    дефицитом товара
            V = 0                                         'обнуление запаса товара
        End If
        Sc1 = Sc1 + V * C1                         ' затраты на хранение товара
        If V < Urmin Then                      ' снижение запаса ниже нормы
            If Zajav = 0 Then                 ' условие отсутствия оформленной
                                                  'заявки на поступление товара
                Sc2 = Sc2 + C2 * Part  ' затраты на организацию поставки
                N = Norm                  ' обращение к функции, вырабатывающей
                          ' случайную ' величину с нормальным распределением
                TZ = Int(MTVZ + N * STVZ) ' случайное время выполнения
                                                                  'очередного заказа
                Zajav = 1                          ' фиксация оформления заявки
            End If
        End If
    Loop
    SC = Sc1 + Sc2 + Sc3
    MSC = MSC + SC                    ' суммарные издержки в Ir-реализациях
    SSC = SSC + SC * SC                           ' сумма квадратов издержек
    Next Ir                                            ' окончание цикла реализаций
              ' результаты моделирования:
    Ccp = MSC / Nr                                                        'сред-
                                                                  ние затраты
    If Nr > 1 Then
        DispC = (SSC-Nr*Ccp*Ccp)/(Nr-1)               ' дисперсия затрат
        ' определение среднего квадратического отклонения затрат:
        If DispC>0 Then SigC = Sqr(DispC) Else SigC = 0
    End If
    Cgar = Ccp+1.28*SigC            ' максимальные гарантированные затраты
        'вывод на экран результата расчета:
    frmZapasy.txtResult = Format$(Cgar, "#####")
End Sub
Function Norm()
    'генератор нормального распределения
    Sz = 0
    For j = 1 To 12
        z = Rnd(1) ' датчик случайных чисел с равномерным распределением
                                                               ' в интервале (0,1)
        Sz = Sz + z
    Next j
    Norm = Sz - 6                          ' центрирование случайной величины
End Function
```


Программа модели производственной фирмы на языке Visual Basic 5.0

1. Стартовая форма frmForm1

Макет стартовой формы приведен на рис. 6.1, с. 104.

1.1. Таблица свойств

Объекты	Свойства	Установки
Командная кнопка 1	Name Caption	Command1 Расчет
Командная кнопка 2	Name Caption	Command2 Очистка
Командная кнопка 3	Name Caption	Command3 Выход

1.2. Процедуры обработки прерываний

```
Private Sub Command1_Click()
    Lz = Val(Text1): P = Val(Text2): C = Val(Text3)
    TD = Val(Text4): STotn = Val(Text5): Nr = Val(Text6)
    For j = 1 To 4: MTobs(j) = Val(Text7(j - 1)): Next j
    Call Model4
End Sub

Private Sub Command2_Click()
    Text8 = "": Text9 = ""
End Sub

Private Sub Command3_Click()
    End
End Sub
```

2. Модуль общего назначения Model4.bas

```
Public MTobs(4), TH(4), TK(4)
Public Mprof As Double, Cprof As Double, Sum2 As Double
```

```
Public SNM As Double, SigProf As Double
Public Lz, P, C, TD, STotn, Nr
Public Ir, Nz, Tz0, Nobs, i, j, z, Tz, N, Tobs, Sz, Min
Public Max, SMT, Fact
Const K = 4                                                   ' число цехов

Public Sub Model4()
    Randomize
        'обнуление глобальных переменных:
    Mprof = 0                           ' начальное значение средней прибыли
    Sum2 = 0                       ' начальное значение суммы квадратов прибыли
    For Ir = 1 To Nr                         ' начало цикла случайных реализаций
        ' обнуление локальных переменных
        Nz = 0: Tz0 = 0: Nobs = 0
        For j = 1 To K: TK(j) = 0: Next j
        Do                                  ' начало внутреннего цикла DO..LOOP
            z = Rnd        ' случайная величина с равномерным распределением
            Tz = Tz0 - Log(z) / Lz                       ' время поступления заказа
            Tz0 = Tz
            If Tz > TD Then Exit Do
            Nz = Nz + 1                                     ' счетчик числа заказов
            For j = 1 To K                       ' начало цикла выполнения заказов
                If Tz > TK(j) Then TH(j)=Tz Else TH(j)=TK(j)
                N = NORM                   ' функция "Нормальное распределение"
                Tobs = MTobs(j) * (1 + N * STotn)
                TK(j) = TH(j) + Tobs                  ' время окончания обслуживания
                If TK(j) > TD Then Exit Do
                Tz = TK(j)
            Next j
            Nobs = Nobs + 1
        Loop
                ' результаты расчета одной реализации:
        Prof = P * Nobs - C                                            ' прибыль
        Mprof = Mprof + Prof                                 ' суммарная прибыль
        Sum2 = Sum2 + Prof * Prof                      ' сумма квадратов прибыли
    Next Ir                                   ' конец цикла случайных реализаций
    Cprof = Mprof / Nr                                         ' средняя прибыль
    Disp = (Sum2 - Nr * Cprof * Cprof) / (Nr - 1)          ' дисперсия прибыли
    If Disp > 0 Then SigProf = Sqr(Disp) Else SigProf = 0
    Gprof = Cprof - 1.28 * SigProf   ' минимальная гарантированная прибыль
    Call Factor                              ' расчет величины числового фактора
    Form1.Text8 = Format$(Fact, "0.000")
    Form1.Text9 = Format$(Gprof, "#####")
End Sub
```



```
Function NORM()
    'процедура-функция "Нормальное распределение"
    Sz = 0
    For i = 1 To 12
        z = Rnd : Sz = Sz + z
    Next i
    NORM = Sz - 6
End Function

Public Sub Factor()
    'процедура расчета числового фактора
    Min = MTobs(1): Max = MTobs(1): SMT = 0
    For f = 1 To K
        SMT = SMT + MTobs(f)
        If MTobs(f) > Max Then Max = MTobs(f)

        If MTobs(f) < Min Then Min = MTobs(f)
    Next
    Fact = (Max - Min) / SMT
End Sub
```

Приложение 5

Программа модели торговой точки на языке Visual Basic 5.0

1. Стартовая форма frmForm1

Макет стартовой формы приведен на рис. 7.1, с. 117.

1.1. Таблица свойств

Объект	Свойства	Установки
Командная кнопка 1	Name Caption	Command1 Расчет
Командная кнопка 2	Name Caption	Command2 Очистка
Командная кнопка 3	Name Caption	Command1 Выход

1.2. Процедуры обработки прерываний

```
Private Sub Command1_Click()
    T(1, 1) = Val(Text1): T(1, 2) = Val(Text2): T(1, 3) = Val(Text3)
    U(1) = Val(Text4): P(1) = Val(Text5): T(2, 1) = Val(Text6):
    T(2, 2) = Val(Text7): T(2, 3) = Val(Text8): U(2) = Val(Text9)
    P(2) = Val(Text10): T(3, 1) = Val(Text11): T(3, 2) = Val(Text12)
    T(3, 3) = Val(Text13): U(3) = Val(Text14): P(3) = Val(Text15)
    Sigm = Val(Text16): Tmin = Val(Text17): Tmax = Val(Text18)
    Nr = Val(Text19)
    Call Model5                        ' обращение к модулю общего назначения
End Sub

Private Sub Command2_Click()
    Text20 = "": Text21 = "": Text22 = ""
End Sub

Private Sub Command3_Click()
    End
End Sub
```

2. Модуль общего назначения Model5.bas

```
Public T(3, 3)              ' массив значений средней выручки для различных
                            ' товаров и ' различных пунктов торговли
Public U(3)                 ' массив значений убытков для различных пунктов
Public P(3)                 ' вероятности внеплановых убытков для разных пунктов
Public Tsum(3) As Double    ' прибыль для различных пунктов для всех
                            ' случайных реализаций
Public Tsum2(3) As Double   ' сумма квадратов прибылей для различных
                            ' пунктов для всех случайных реализаций
Public Mcp(3)               ' средняя прибыль для различных пунктов
Public Gar(3)               ' минимальная гарантированная прибыль
                            ' для 'различных пунктов
Public Sigma(3)             ' средние квадратические отклонения прибыли
                            ' для различных пунктов
Public i, j, Ir, Nr, Tmin, Tmax, Sigm, Eta, TS, z
Public Disp, E, k
Const R = 1.645

Public Sub Model5()
    'обнуление сумм, используемых для расчета МО и СКО:
    For i = 1 To 3: Tsum(i) = 0: Tsum2(i) = 0: Next i
        'начало цикла случайных реализаций
        For Ir = 1 To Nr
            'начало цикла перебора торговых точек:
            For i = 1 To 3
                Eta = UNR: TS = T(i, 1) * (1+Sigm*Eta)
                Eta = UNR: TS = TS + T(i, 2) * (1+Sigm*Eta)
                Eta = UNR: TS = TS + T(i, 3) * (1+Sigm*Eta)
                z = Rnd
                If z < P(i) Then
                    Eta = UNR: TS = TS - U(i) * (1+Sigm*Eta)
                End If
                Tsum(i) = Tsum(i) + TS
                Tsum2(i) = Tsum2(i) + TS * TS
            Next i                    ' конец цикла перебора торговых точек
        Next Ir                       ' конец цикла случайных реализаций
        ' расчет показателей (МО И СКО) для каждой точки:
    For i = 1 To 3
        Mcp(i) = Tsum(i) / Nr                ' расчет средней прибыли
        ' расчет дисперсии и СКО прибыли:
        If Nr > 1 Then
            Disp = (Tsum2(i) - Nr * Mcp(i)*Mcp(i))/(Nr - 1)
            Sigma(i) = Sqr(Disp)
        Else
            Sigma(i) = 0
        End If
        ' минимальная гарантированная прибыль:
        Gar(i) = Mcp(i) - R * Sigma(i)
    Next i
    ' вывод результатов ' расчета:
    Form1.Text20.Text = Format$(Gar(1), "#####")
    Form1.Text21.Text = Format$(Gar(2), "#####")
    Form1.Text22.Text = Format$(Gar(3), "#####")
End Sub                   ' конец главной программы

Public Function UNR()
    ' Функция "Усеченное нормальное распределение"
    Do
        E = 0
        For k = 1 To 12
            z = Rnd: E = E + z
        Next k
        E = E - 6
        If E >= Tmin And E <= Tmax Then
            UNR = E: Exit Do
        End If
    Loop
End Function
```

Приложение 6

Программа финансовой модели Герца на языке Visual Basic 5.0

1. Стартовая форма frmForm1

Макет стартовой формы приведен на рис. 8.1, с. 128.

1.1. Таблица свойств

Объект	Свойства	Установки
Командная кнопка 1	Name Caption	Command1 Расчет
Командная кнопка 2	Name Caption	Command2 Очистка
Командная кнопка 3	Name Caption	Command3 Выход

1.2. Процедуры обработки прерываний

```
Private Sub Command1_Click()
    Mrash = Val(Text1): SigRash = Val(Text2): Mrynok = Val(Text3)
    SigRynok = Val(Text4): Nr = Val(Text5): NT = Val(Text6)
    a(0) = Val(Text7): a(1) = Val(Text8): a(2) = Val(Text9)
    a(3) = Val(Text10): a(4) = Val(Text11): a(5) = Val(Text12)
    Ver = 1 / (NT - 1)                      ' вероятность попадания в участок
    Call Modul6                             ' обращение к модулю
End Sub

Private Sub Command2_Click()
    Text13 = "": Text14 = "": Text15 = ""
End Sub

Private Sub Command3_Click()
    End
End Sub
```

2. Модуль общего назначения

```
Public a(5) As Single, n As Single, z As Single
Public r As Single, NT As Integer, Ver As Single
```



```
Public Sprof As Double, Sprof2 As Double, Mrash
Public SigRash, Mrynok, SigRynok, Nr, I, Dol, Prof
Public Rynok, Rash, Mprof, SigProf, Disp, Gprof

Public Sub Modul6()
    ' Обнуление глобальных переменных:
    Sprof = 0: Sprof2 = 0
    ' Цикл случайных реализаций:
    For Ir = 1 To Nr
        Call Norm: Rash = Mrash + n * SigRash
        Call Norm: Rynok = Mrynok + n * SigRynok
        Call Rasp: Dol = r
        Prof = Rynok * Dol - Rash
        Sprof = Sprof + Prof
        Sprof2 = Sprof2 + Prof * Prof
    Next Ir
    Mprof = Sprof / Nr
    Disp = (Sprof2 - Nr * Mprof * Mprof) / (Nr - 1)
    If Disp > 0 Then SigProf = Sqr(Disp) Else SigProf = 0
    Gprof = Mprof - 1.28 * SigProf
    ' Результаты моделирования:
    Form1.Text13 = Format$(Mprof, "#######")
    Form1.Text14 = Format$(SigProf, "#######")
    Form1.Text15 = Format$(Gprof, "#######")
End Sub

Sub Norm()
    S = 0
    For j = 1 To 12
        z = Rnd(1): S = S + z
    Next
    n = S - 6
End Sub

Sub Rasp()
    z = Rnd(1)
    For j = 1 To NT - 1
        If z <= Ver * j Then
            M = j: GoTo Met
        End If
    Next
Met:
    z = Rnd(1)
    r = a(M - 1) + (a(M) - a(M - 1)) * z
End Sub
```

Паскаль-программа модели звена управления

1. Главная программа Model7_0

```
Program Model7_0;
Uses TPCrt, Model7_1, Model7_2;
Label M0,M1;
Const MZmax=200; Nk=2;        {максимальное число заявок и число каналов}
  {Type NN=array[1..14] of real;              }   описано в Model7_1
  NNN=array[1..25,1..8] of real;              }   }
  {MasR=array[1..Nk] of real;                 }
  MasI=array[1..Nk] of integer;               }   описано в Model7_2
  MasMZ=array[1..MZmax] of real;              }
  MMR=array[1..Nk,1..MZmax] of MasMZ;  }      }
Var ID: NN; Res: NNN;
 Jm,Jn,Jv,Kv,i,j,k,M,N : byte;
 Mode,Ireal,Sreal : integer;
 Tfinish,DTs2f,SDTs2f: real;
 DTs1f,SDTs1f,TZcp,Ts1cp,Ts2cp,P : MasR;
 Nz1f,Nz2f,Ns1f,Ns2f,NSz1f,NSz2f,Ns,NSs,NSs1f,
 NSs2f : MasI;
 TH1,TK1,TH2,TK2 : MasMZ;
 TH1f,TK1f,TH2f,TK2f : MMR;
 Ch,Ndisk : char;
{------------------------------------------------------------}
Procedure ZERO_GLOB_MAS;
    {процедура обнуления глобальных массивов}
Begin
    for k:=1 to Nk do
        begin
            NSz1f[k]:=0; NSz2f[k]:=0; NSs1f[k]:=0;
            NSs2f[k]:=0;NSs[k]:=0; SDTs1f[k]:=0;
        end;
    SDTs2f:=0;
End;     {Zero_Glob_Mas}
{------------------------------------------------------------}
Procedure ZERO_LOC_MAS;
      {процедура обнуления локальных массивов}
Begin
    for k:=1 to Nk do
```



```
        begin
            Nz1f[k]:=0; Nz2f[k]:=0; Ns1f[k]:=0; Ns2f[k]:=0;
            Ns[k]:=0; DTs1f[k]:=0;
        end;
    DTs2f:=0;
End;     {Zero_Loc_Mas}
{------------------------------------------------------------}
Procedure STATISTIC;
      {процедура переименования параметров}
Begin
    Res[Jv,3]:= NSs[1]/Sreal; Res[Jv,4]:= NSs[2]/Sreal;
    Res[Jv,5]:=(NSs2f[1] + NSs2f[2])/Sreal; Res[Jv,6]:= SDTs1f[1]/Sreal/Tfinish;
    Res[Jv,7]:= SDTs1f[2]/Sreal/Tfinish ;
    Res[Jv,8]:= SDTs2f/Sreal/Tfinish ;
End;     {Statistic}
{******************** Main Program *********************}
BEGIN
    TextBackGround(1);                              {цвет фона – черный}
    TextColor(14);                                  {цвет текста – желтый}
    ClrScr;                                         {очистка экрана}
    BASE_DATA(ID);                                  {процедура TPU-модуля Model7_1}
    Randomize;                                      {инициализация датчика случайных чисел}
M0: ClrScr;                                         {очистка экрана}
    CORRECT_DATA(ID);                               {процедура TPU-модуля Model7_1}
    ClrScr;
    TZcp[1]:=ID[1]; TZcp[2]:=ID[2]; Ts1cp[1]:=ID[3];
    Ts2cp[1]:=ID[4]; Ts1cp[2]:=ID[5]; Ts2cp[2]:=ID[6];
    M:=Round(ID[9]); N:=Round(ID[12]); Tfinish:=ID[13];
    Sreal:=Round(ID[14]);
    Kv:=M*N;                                        {число вариантов}
    Jv:=0;                                          {обнуление счетчика числа вариантов}
    for Jm:=1 to M do                               {цикл значений P[1]}
      begin
        P[1]:=ID[7] + (Jm – 1)*ID[8];
          for Jn:=1 to N do                         {цикл значений P[2]}
            begin
              Jv:=Jv + 1;                           {счетчик числа вариантов}
              GotoXY(35,1); Writeln('ВАРИАНТ ',Jv:2);
              P[2]:=ID[10] + (Jn – 1)*ID[11];
              Res[Jv,1]:=P[1]; Res[Jv,2]:=P[2];
              ZERO_GLOB_MAS;                        {обнуление глобальных массивов}
                      {начало цикла случайных реализаций}
              for Ireal:=1 to Sreal do
                begin
                  ZERO_LOC_MAS;                     {обнуление локальных массивов}
```

```
          GotoXY(30,2);
          Writeln('Расчет ',Ireal,'-й реализации');
          {обращение к процедуре формирования потоков заявок}
          FORM_POTOK_ZAJAVKA(Tfinish,Nz1f,NSz1f,TZcp,TH1f);
          for k:=1 to Nk do
           begin
             for j:=1 to Nz1f[k] do TH1[j]:=TH1f[k,j];
             {обращение к процедуре обслуживания заявок в 1-й фазе}
             SERVICE_PHASE(1,Nz1f[k],TH1,TK1,Ns1f[k],NSs1f[k],
                    Ts1cp[k],Tfinish,DTs1f[k],SDTs1f[k]);
             for j:=1 to Ns1f[k] do TK1f[k,j]:=TK1[j];
           end;    {for k}
          {обращение к процедуре разделения потоков заявок}
          SELECT_POTOK(Ns1f,P,Nz2f,NSz2f,Ns,NSs,TH2f,TK1f);
          for j:=1 to Nz2f[1] do TH1[j]:=TH2f[1,j];
          {обращение к процедуре обслуживания заявок 1-го приоритета}
          SERVICE_PHASE(2,Nz2f[1],TH1,TK1,Ns2f[1],NSs2f[1]
                               Ts2cp[1],Tfinish,DTs2f,SDTs2f);
          for j:=1 to Ns2f[1] do
           begin
             TH2f[1,j]:=TH1[j]; TK2f[1,j]:=TK1[j];
           end;
          for j:=1 to Nz2f[2] do TH2[j]:=TH2f[2,j];
         {обращение к процедуре обслуживания заявки 2-го приоритета}
          SERVICE_LOW_PRIORITY(Ns2f[1],Nz2f[2],Ns2f[2],NSs2f[2],
           TH1,TK1,TH2,TK2,Ts2cp[2],Tfinish,DTs2f,SDTs2f);
        end;     {for Ireal}
       STATISTIC;                          {накопление расчетных данных}
      end;     {for Jn}
    end;     {for Jm}
M1: ClrScr;
  GLOBRESULT(ID,Res,Kv);              {процедура TPU-модуля Model7_2}
  Writeln; Write(' ':9,'Будете ли продолжать расчеты? (Y/N) '); Readln(Ch);
  if (Ch='Y') or (Ch='y') or (Ch='H') or (Ch='н') then goto M0;
  ClrScr;                                               {очистка экрана}
END.    {Model7_0}
```

2. Программный модуль Model7_1

```
Unit Model7_1;
Interface
Uses TPCrt,TPWindow;
Type NN=array[1..14] of real;
   NNN=array[1..25,1..8] of real;
{--------------------------------------------------}
```



```
Procedure BASE_DATA(var ID:NN);
Procedure CORRECT_DATA(var ID:NN);
Procedure GLOBRESULT(ID: NN; RES: NNN; Kv: byte);
Implementation
Procedure BASE_DATA(var ID:NN);
   {процедура задания базовых значений исходных данных}
Begin
   ID[1]:=1.0; ID[2]:=1.0; ID[3]:=1.0; ID[4]:=1.0;
   ID[5]:=1.0; D[6]:=1.0; ID[7]:=0.25; ID[8]:=0.25;
   ID[9]:=3; ID[10]:=0.25; ID[11]:=0.25; ID[12]:=3;
   ID[13]:=10.0; ID[14]:=1000;
End;     {BASE_DATA}
{--------------------------------------------------}
Procedure CORRECT_DATA(var ID: NN);
  {процедура корректировки исходных данных}
Label Met;
Type S1=array[1..21] of string[70];
Const T: array[1..21] of integer = (0,0,0,1,2,0,3,4,5,6,0,0.7,8,9,0,10,11,12,13,14);
'  ┌─────┬─────────────────────────────────────────┬──────┐',
'  │     │                                         │      │',
'  │     │ Среднее время между заявками:           │      │',
'  │  1  │     1-го приоритета, ч                  │      │',
'  │  2  │     2-го приоритета, ч                  │      │',
'  │     │ Среднее время обслуживания заявок:      │      │',
'  │  3  │     1-го приоритета в 1-й фазе,ч        │      │',
'  │  4  │     1-го приоритета во 2-й фазе, ч      │      │',
'  │  5  │     2-го приоритета в 1-й фазе, ч       │      │',
'  │  6  │     2-го приоритета во 2-й фазе,  ч     │      │',
'  │     │ Вероятность перехода во 2-ю фазу        │      │',
'  │     │ для заявок 1-го приоритета:             │      │',
'  │  7  │      начальное значение                 │      │',
'  │  8  │      шаг изменения                      │      │',
'  │  9  │      количество шагов                   │      │',
'  │     │ для заявок 2-го приоритета:             │      │',
'  │ 10  │      начальное значение                 │      │',
'  │ 11  │      шаг изменения                      │      │',
'  │ 12  │      количество шагов                   │      │',
'  │ 13  │ Период функционирования системы, ч      │      │',
'  │ 14  │ Число случайных реализаций              │      │',
'  └─────┴─────────────────────────────────────────┴──────┘');
 Var i,N : byte;
         Par: real;
         Ch: char;
 Begin
   ClrScr;
```

```
    Writeln(' ':20,'Исходные данные имеют следующие значения:');
    While True do
      begin
       for i:=1 to 21 do Writeln(' ':10,C1[i]);
       for i:=1 to 20 do
        begin
          if (i>3)and(i<>6)and(i<>11)and(i<>12)and(i<>16) then
            begin
              GotoXY(62,i); Write(ID[T[i]]:4:2);
            end;
          end;
      GotoXY(60,21); Write(Round(ID[14]):6);
      GotoXY(15,23);
      Write('Хотите изменить какой-либо параметр?(Y/N)');
      Readln(Ch);
      if (Ch='Y')or(Ch='y')or(Ch='Н')or(Ch='н') then
          begin
            GotoXY(15,23); Writeln(' ':50);
            GotoXY(15,23);
            Write('Введите порядковый номер параметра: ');
            Readln(N);
            GotoXY(15,24);
            Write('Введите новое значение параметра: ');
            Readln(Par); ID[N]:=Par;
            GotoXY(15,23); Writeln(' ':50);
            GotoXY(15,24); Writeln(' ':50);
            GotoXY(20,1);
            Writeln('Теперь исходные данные имеют следующие значения:
            ');
          end else goto Met;
      end;     {While}
Met:;
End;     {WWOD_DATA}
{------------------------------------------------}
Procedure GLOBRESULT(ID: NN; RES: NNN; Kv: byte);
     {процедура выдачи сводной таблицы результатов моделирования}
Type S2=array[1..6] of string[78];
Const C2:S2 = (
```

№ варианта	P[1]	P[2]	Среднее число обслуженных заявок			Среднее относительное время занятости		
			1ф 1к	1ф 2к	2ф 1к	1ф 1к	1ф 2к	2ф 1к

```
');
```



```
Var i,Iv : byte;
Begin
   TextColor(14);                              {цвет текста –  желтый}
   GotoXY(32,1); Writeln('ИСХОДНЫЕ ДАННЫЕ:');
   GotoXY(12,2);
   Write('TZcp[1]=',ID[1]:4:1,'; TZcp[2]=',ID[2]:4:1);
   Writeln('; Ts1f[1]=',ID[3]:4:1,'; Ts1f[2]=',ID[4]:4:1);
   Write(' ':11,'Ts2f[1]=',ID[5]:4:1,';
   Ts2f[2]=',ID[6]:4:1);
   Writeln('; Tfinish=',ID[13]:4:1,';
   Sreal=Round(ID[14]):4);
   Writeln;
   GotoXY(32,4); Writeln('РАСЧЕТНЫЕ ДАННЫЕ');
   for i:=1 to 7 do Writeln(' ':9,C2[i]);
     for Iv:=1 to Kv do
       begin
          Write(' ':9,'|' ,Iv:2, '|',Res[Iv,1]:4:2, '|',Res[Iv,2]:4:2);
          Write('| ',Res[Iv,3]:5:2,'| ',Res[Iv,4]:5:2,'| ',Res[Iv,5]:5:2);
          Write('| ',Res[Iv,6]:4:2,' | ',Res[Iv,7]:4:2);
          Writeln(' | ',Res[Iv,8]:4:2,' | ');
       end;
     Writeln(' ':9,C2[8]);
End;    {GLOBRESULT}
End.    {Model7_1}
```

3. Программный модуль Model7_2

```
Unit Model7_2;
Interface
Uses TPCrt;
Const MZmax=200; Nk=2;
Type MasR=array[1..Nk] of real;
        MasI=array[1..Nk] of integer;
MasMZ=array[1..MZmax] of real;
MMR=array[1..Nk] of MasMZ;
{------------------------------------------------------------------------}
Procedure FORM_POTOK_ZAJAVKA(Tfinish:real; var Nz1f,NSz1f:MasI;
                                              TZcp:MasR; var TH1f:MMR);
Procedure SERVICE_PHASE(f,Nz:integer; var TH,TK:MasMZ;
                    var Nsf,NSsf:integer;  Ts,Tfinish:real; var DTs, SDTs:real);
Procedure SELECT_POTOK(Ns1f:MasI; P:MasR; var Nz2f,
                          NSz2f,  Ns, NSs:MasI; var TH2f:MMR; TK1f:MMR);
Procedure SERVICE_LOW_PRIORITY(Ns1,Nz2:integer; var Ns2,NSs2: integer;
                 var TH1,TK1,TH2,TK2:MasMZ;Ts2,Tfinish:real; var DTs,SDTs:real);
```

```
Procedure ANALIZ(N:integer; T1,T2:real; TH,TK:MasMZ;
                                          var F1,F2,F3:integer;var THF,TKF:real);
Implementation
{-------------------------------------------------------------------------------}
Procedure FORM_POTOK_ZAJAVKA(Tfinish:real; var Nz1f,
                                   NSz1f:MasI; TZcp:MasR; var TH1f:MMR);
           {процедура формирования потока заявок}
Var k,n: integer;
   T,z: real;
Begin
   for k:=1 to Nk do
     begin
       T:=0;
       While T < Tfinish do
        begin
          Nz1f[k]:=Nz1f[k] + 1; z:=Random;
          n:=Nz1f[k]; TH1f[k,n]:=T - TZcp[k]*ln(z);
          T:=TH1f[k,n];
        end;    {While}
       Nz1f[k]:=Nz1f[k] - 1;                        {ликвидация лишней заявки}
       NSz1f[k]:=NSz1f[k] + Nz1f[k];
     end;     {for k}
End;    {FORM_POTOK_ZAJAVKA}
{------------------------------------------}
Procedure SERVICE_PHASE(f,Nz:integer; var TH,TK:MasMZ;
              var Nsf,NSsf:integer; Ts,Tfinish:real;
              var DTs,SDTs:real);
              {процедура обслуживания однородных заявок}
Label Met;
Var DT,TH0,TK0,z: real;
             j: integer;
Begin
   if Nz > 0 then
     begin
       for j:=1 to Nz do
         begin
           z:=Random; TK[j]:=TH[j] - Ts*ln(z);
           if TK[j] > Tfinish then goto Met;
           if (j >1) and (TH[j] < TK0) then
             begin
               DT:=TK[j] - TH[j]; TH[j]:=TK0;
               TK[j]:=TH[j] + DT;
               if TK[j] > Tfinish then goto Met;
             end;   {if}
           TH0:=TH[j]; TK0:=TK[j]; Nsf:=Nsf + 1;
```



```
           DTs:=DTs + TK0 - TH0;
         end;{for j}
     end;     {if Nz}
Met: if f=1 then SDTs:=SDTs + DTs;
           NSsf:=NSsf + Nsf;
End;     {SERVICE_PHASE}
{--------------------------------------------}
Procedure SELECT_POTOK(Ns1f:MasI; P:MasR; var Nz2f,
        NSz2f,Ns,NSs:MasI; var TH2f:MMR; TK1f:MMR);
            {процедура разделения потоков}
Var j,k,n,nz:integer;
           z: real;
Begin
   nz:=0;
     for k:=1 to Nk do
       begin
        if Ns1f[k] > 0 then
          begin
           for j:=1 to Ns1f[k] do
            begin
              z:=Random;
              if z < P[k] then
               begin
                 Nz2f[k]:=Nz2f[k] + 1; n:=Nz2f[k]; TH2f[k,n]:=TK1f[k,j];
               end else Ns[k]:=Ns[k] + 1;
              end;{for j}
           end;     {if Nz1f}
         NSs[k]:=NSs[k] + Ns[k];
         NSz2f[k]:=NSz2f[k] + Nz2f[k];
       end;     {for k}
End;     {SELECT_POTOK}
{--------------------------------------------}
Procedure SERVICE_LOW_PRIORITY(Ns1,Nz2:integer;
        var Ns2,NSs2:integer; var TH1,TK1,TH2,TK2:MasMZ;
             Ts2,Tfinish:real; var DTs,SDTs:real);
             {процедура обслуживания потока низкого приоритета}
Label Met1,Met2,Met3;
Var j,F1,F2,F3: integer; DT,TH0,TK0,THV,THF,TKF,z: real;
Begin
   if Nz2 > 0 then
     begin
      for j:=1 to Nz2 do
       begin
         z:=Random; TK2[j]:=TH2[j] - Ts2 * ln(z);
         if TK2[j] > Tfinish then goto Met3;
```

```
          if (j > 1) and (TH2[j] < TK0) then
            begin
             DT:=TK2[j] - TH2[j]; TH2[j]:=TK0; TK2[j]:=TH2[j] + DT;
             if TK2[j] > Tfinish then goto Met3;
            end;
          THV:=TH2[j];
          if Ns1 > 0 then
            begin
Met1:        ANALIZ(Ns1,THV,TK2[j],TH1,TK1,F1,F2,F3,THF,TKF);
             if F1=1 then goto Met2;
             if F2=1 then
              begin
               TK2[j]:=TKF + TK2[j] - THF; DTs:=DTs + THF - THV;
              end else TK2[j]:=TKF + TK2[j] - THV;
             THV:=TKF;
             if TK2[j] < Tfinish then goto Met1
             else goto Met3;
            end;    {if Ns1}
Met2:     TH0:=THV; TK0:=TK2[j]; Ns2:=Ns2 + 1;
          DTs:=DTs + TK0 - TH0;
        end;    {for j}
      end;    {if Nz2}
Met3: NSs2:=NSs2 + Ns2; SDTs:=SDTs + DTs;
End;    {SERVICE_LOW_PRIORITY}
{--------------------------------------------------}
Procedure ANALIZ(N:integer; T1,T2:real; TH,TK: MasMZ; var F1,F2,F3:integer;
                 var THF,TKF:real);    {процедура анализа состояния}
Label Met;
Var i: integer;
Begin
   F1:=0; F2:=0; F3:=0;
     for i:=1 to N do
       begin
        if (T1 < TH[i]) and (TH[i] < T2) then
          begin
             F2:=1; THF:=TH[i]; TKF:=TK[i]; goto Met
          end else if (TH[i] <= T1) and (T1 < TK[i]) then
          begin
             F3:=1; THF:=TH[i]; TKF:=TK[i]; goto Met
          end;
       end;    {for i}
     F1:=1;
Met:
End;    {ANALIZ}
End    {Model7_2}
```

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

СОДЕРЖАНИЕ

Учебное пособие

Варфоломеев Валентин Иванович

АЛГОРИТМИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТОВ ЭКОНОМИЧЕСКИХ СИСТЕМ

Заведующая редакцией *Л. А. Табакова*
Редактор *Л. Д. Григорьева*
Младший редактор *Н. А. Федорова*
Технические редакторы *Е. В. Кузьмина, И. В. Завгородняя*
Корректоры *Т. М. Колпакова, Н. П. Сперанская*
Художественное оформление *О. В. Толмачева*

ИБ № 3420
Лицензия ЛР № 010156 от 29.01.97

Подписано в печать 31.01.2000. Формат 60×88/16
Гарнитура «Таймс». Печать офсетная
Усл. печ. л. 12,74. Уч.-изд. л. 11,24. Тираж 4000 экз.
Заказ 3041. «С» 023

Издательство «Финансы и статистика»,
101000, Москва, ул. Покровка, 7
Телефон (095) 925-35-02, факс (095) 925-09-57
E-mail: mail@finstat.ru http://www.finstat.ru

Великолукская городская типография
Комитета по средствам массовой информации и связям с общественностью администрации Псковской области,
182100, Великие Луки, ул. Полиграфистов, 78/12
Тел./факс: (811-53) 3-62-95
E-mail: VTL@MART.RU

51 г

Варфоломеев Валентин Иванович родился в 1924 г. в Бобруйске. Более 40 лет прослужил в Вооруженных силах СССР. Участник Великой Отечественной войны. В 1952 г. окончил Артиллерийскую академию им. Ф.Э. Дзержинского. С 1953 г. занимается педагогической деятельностью. Общий стаж научно-педагогической работы 44 года. В 1957 г. защитил кандидатскую диссертацию. Доцент по кафедре теплотехники с 1961 г. Профессор по кафедре проектирования летательных аппаратов с 1980 г. Автор более 90 научных трудов, в том числе 6 монографий, 2 учебников и 36 учебных пособий.

В настоящее время Варфоломеев В.И. — профессор кафедры экономической информатики Московского государственного университета коммерции. Специалист в области моделирования сложных систем.

ISBN 5-279-02123-7
9 785279 021239