

Министерство образования и науки Российской Федерации

Саратовский социально-экономический институт (филиал)
ФГБОУ ВПО «РЭУ им. Г.В. Плеханова»

Кафедра информационных систем в экономике

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ

Учебное пособие

для студентов, обучающихся по направлению подготовки
080500.62 «Бизнес-информатика»

Саратов
2014

УДК 004.9:33
ББК 32.97+65
И52

Авторы-составители:

д-р экон. наук, канд. физ.-мат. наук, профессор **Е.П. Бочаров**,
канд. экон. наук, доцент **О.Н. Алексенцева**

И52 Имитационное моделирование экономических процессов: учебное пособие для студентов, обучающихся по направлению подготовки 080500.62 «Бизнес-информатика» / авт.-сост. **Е.П. Бочаров**, **О.Н. Алексенцева** / Саратовский социально-экономический институт (филиал) ФГБОУ ВПО «РЭУ им. Г.В. Плеханова». – Саратов, 2014. – 160 с.

Рекомендует к печати
редакционно-издательский совет
ССЭИ (филиала) РЭУ им. Г.В. Плеханова
5 марта 2014 г.

Рецензенты:

доктор физ.-мат. наук, профессор *Т.Э. Шульга*,
доктор физ.-мат. наук, профессор *В.Н. Гусятников*

УДК 004.9:33
ББК 32.97+65

© Е.П. Бочаров, О.Н. Алексенцева, 2014
© Саратовский социально-
экономический институт (филиал)
ФГБОУ ВПО «РЭУ им. Г.В. Плеханова», 2014

Предисловие

История развития имитационного моделирования (ИМ) насчитывает уже более 50 лет. Однако только в последнее десятилетие в связи с появлением объектно-ориентированных визуальных инструментальных средств и высокопроизводительных персональных компьютеров ИМ постепенно становится повседневным инструментом для разработчиков самых различных проектов – транспортных, промышленных, медицинских, военных и т.п. Так, наличие имитационной модели и обоснование с ее помощью выбранного варианта организации является обязательным в странах Европейского Союза в комплекте документов, подаваемых на рассмотрение для проектирования или модернизации нового производства либо технологического процесса. ИМ придает все большее значение и в директивных документах Правительства РФ по технологическому развитию. Такая популярность ИМ по сравнению с другими видами математического моделирования объясняется возможностями детального описания моделируемой системы построения динамических моделей.

Цель данного учебного пособия – помочь студентам в овладении теоретическими основами имитационного моделирования экономических процессов, привить им практические навыки по разработке моделей экономических процессов, их программной реализации на ПК и анализу результатов моделирования.

Объединение в одном учебном пособии как теоретических основ имитационного моделирования экономических процессов, так и методик практической реализации имитационных моделей определяет, по нашему мнению, его ценность.

Построение данного учебного пособия отражает структуру дисциплины. В первой части (введение, темы 1–5) излагаются теорети-

ческие основы имитационного моделирования. Вторая часть содержит практикум по ИМ, включает 13 заданий, выполняемых студентами в дисплейном классе под руководством преподавателя. После освоения материала каждого из заданий студентам предлагаются задания для самостоятельной работы.

Существуют и развиваются более десятка популярных систем ИМ, как зарубежных, так и российских. Нами была выбрана для проведения практических занятий система ИМ GPSS World Student version, разработанная компанией MinutemanSoftware (США) и имеющая следующие преимущества:

- используемая версия бесплатна для образовательных учреждений, а ее дистрибутив свободно «скачивается» с сайта компании по адресу: <http://www.Minutemansoftware.com> (отметим, что все остальные системы ИМ достаточно дороги);

- система ИМ GPSS World Student version полнофункциональна и по своим возможностям не уступает другим системам ИМ, а имеющееся ограничение на количество операторов никак не отражается на учебном процессе.

Данное пособие предназначено для использования студентами, обучающимися по направлению подготовки 080500.62 «Бизнес-информатика», а также рекомендуется для студентов направления 080100.62 «Экономика», профиль «Статистика».

Введение

Понятие модели и моделирования. При использовании метода моделирования свойства и поведение объекта изучают путем применения вспомогательной системы – модели, находящейся в определенном объективном соответствии с используемым объектом.

Под *объектом исследования* понимается либо некоторая система, элементы которой в процессе достижения конечной цели реализуют один или несколько процессов, либо некоторый процесс, реализуемый элементами одной или нескольких систем.

Представления о тех или иных свойствах объектов, их взаимосвязях формируются исследователем в виде описаний этих объектов на обычном языке, в виде рисунков, графиков, формул или реализуются в виде макетов и других устройств. Подобные способы описания обобщаются в едином понятии – *модель*, а построение и изучение моделей называется *моделированием*. Модели позволяют вынести упрощенное представление о системе и получить некоторые результаты намного проще, чем при изучении реального объекта. Более того, гипотетически модели объекта могут быть исследованы и изучены перед тем, как объект будет создан.

В практике исследования производственно-экономических объектов модели могут применяться для самых разных целей, что вызывает использование моделей различных классов. Построение одной-единственной математической модели для сложной производственной системы практически не представляется возможным без разработки вспомогательных моделей. Поэтому, как правило, при создании конечной математической модели исследуемого объекта строят частные вспомогательные модели, отражающие ту или иную информацию об объекте, имеющуюся у разработчика на данном этапе построения модели.

Значение моделирования в экономике. Моделирование, особенно математическое моделирование, широко применяется в самых различных областях науки, техники, экономических и социальных приложениях.

Особенно важно значение моделирование имеет в экономике. Дело в том, что проводить натуральные эксперименты над экономическими объектами (предприятиями, банками, отраслями экономики и т.п.) очень дорого, а зачастую опасно и даже невозможно. Ведь неудачный экономический эксперимент может привести к большим потерям финансовых и материальных средств, повлиять на судьбы миллионов людей. Моделирование дает возможность не вмешиваться в текущий ход развития экономической системы, решать задачи типа «что будет, если...». Проведя расчеты по математической модели, можно выяснить, как влияют те или иные параметры, исходные значения показателей на развитие экономической системы. В случае, если результаты исследования математической модели экономической системы представляются достаточно надежными, можно выработать рекомендации для практиков, которые впоследствии можно применить для управления реальными экономическими процессами.

Поэтому во всех странах с достаточно высоким уровнем развития науки (США, страны ЕС, СССР, а затем Россия и страны СНГ) существуют и развиваются многочисленные школы математического, в частности имитационного, моделирования.

Из отечественных ученых можно отметить вклад в моделирование экономических систем лауреата Нобелевской премии академика Л. Канторовича, академиков В. Макарова, Н. Федоренко, А. Гранберга. Всемирную известность приобрели работы русского ученого, эмигрировавшего после революции в США, лауреата Нобелевской премии В. Леонтьева.

Классификация моделей. Приведем классификацию моделей, отражающую в первую очередь методологические вопросы построения математических моделей и нахождения их решения с помощью персонального компьютера (ПК).

По форме представления модели делят на *физические, символические и смешанные*.

Физические модели подразделяются на модели подобия и аналоговые. Модели подобия характеризуются некоторыми масштабными изменениями, выбираемыми в соответствии с критериями подобия

(например, *глобус* – модель земного шара). Природа процесса, его физическая сущность одинаковы как для модели, так и для исследуемого оригинала. Аналоговые модели основаны на известных аналогиях между протеканием процессов в механических, тепловых, электрических, пневматических, гидравлических и других динамических системах и предназначены для исследования статических и динамических свойств объекта (например, изучение колебаний пружинного маятника на модели, реализуемой электрическим колебательным контуром).

Символические модели характерны тем, что параметры реального объекта и отношения между ними представлены символами: семантическими (словами), математическими, логическими. Класс символических моделей весьма широк. Наряду со словесными описаниями функционирования объектов – сценариями, сюда также относятся схематические модели: чертежи, графики, блок-схемы, логические блок-схемы (например, алгоритмы программ) и таблицы решений, кривые, таблицы и монограммы, а также математические описания – математические модели.

Среди *смешанных* моделей особое значение в экономической практике имеют *человеко-машинные модели* (программа, реализующая на ПК некоторую математическую модель, плюс человек, принимающий решение за счет обмена информацией с моделью).

Форма модели определяет и метод работы с ней. При исследовании различного рода объектов применяются три вида моделирования:

- физическое, когда модель воспроизводит изучаемый процесс (оригинал) с сохранением его физической природы;
- аналоговое, основанное на известных аналогиях между протеканием механических, тепловых, электрических, ядерных и других динамических процессов (изучение свойств колебаний пружинного маятника на модели электрического колебательного контура представляет собой аналоговое моделирование с использованием принципов прямой аналогии);
- математическое, в основе которого лежит исследование математического описания (математической модели) изучаемого объекта.

В связи с бурным развитием персональных компьютеров появилась возможность реализовать даже очень сложные математические модели. Именно поэтому наиболее важное значение приобрел такой класс символических моделей, как математические.

**ОСНОВНЫЕ ПОНЯТИЯ
ЭКОНОМИКО-МАТЕМАТИЧЕСКОГО
МОДЕЛИРОВАНИЯ**

1.1. Целевое назначение модели. Виды моделей

По целевому назначению экономико-математические модели подразделяются на модели структуры, функционирования и стоимостные (модели расхода ресурсов).

Модели структуры отображают связи между компонентами объекта и внешней средой и подразделяются на:

- каноническую модель, характеризующую взаимодействие объекта с окружением через входы и выходы;
- модель внутренней структуры, характеризующую состав компонентов объекта и связи между ними;
- модель иерархической структуры (дерево системы), в которой объект (целое) расчленяется на элементы более низкого уровня, действия в которых подчинены интересам целого.

Модели функционирования включают широкий спектр символических моделей, например:

- модель жизненного цикла системы, описывающая процессы существования системы от зарождения замысла ее создания до прекращения функционирования;
- модели операций, выполняемых объектом и представляющих описание взаимосвязанной совокупности процессов функционирования отдельных элементов объекта при реализации тех или иных функций объекта. Так, в состав моделей операций могут входить модели надежности, характеризующие выход элементов системы из строя под влиянием эксплуатационных факторов, и модели живуче-

сти, характеризующие выход элементов системы из строя под влиянием целенаправленного воздействия внешней среды;

- информационные модели, отображающие во взаимосвязи источники и потребители информации, виды информации, характер ее преобразования, а также временные и количественные характеристики данных;

- процедурные модели, описывающие порядок взаимодействия элементов исследуемого объекта при выполнении различных операций, например обработки материалов, деятельности персонала, использования информации, в том числе и реализации процедур принятия управленческих решений;

- временные модели, описывающие процедуру функционирования объекта во времени и распределение ресурса «время» по отдельным компонентам объекта.

Стоимостные модели, как правило, сопровождают модели функционирования объекта и по отношению к ним вторичны, «питаются» от них информацией и совместно с ними позволяют проводить комплексную технико-экономическую оценку объекта или его оптимизацию по экономическим критериям.

При анализе и оптимизации производственно-экономических объектов проводится объединение построенных математических функциональных моделей с математическими стоимостными моделями в единую экономико-математическую модель.

Экономико-математической моделью (ЭММ) называется выражение, состоящее из совокупности связанных между собой математическими зависимостями (формулами, уравнениями, неравенствами, логическими условиями) величин – факторов, все или часть которых имеют экономический смысл.

По своей роли в ЭММ эти факторы целесообразно подразделить на *параметры* и *характеристики* (рис. 1.1). При этом параметрами объекта называются факторы, характеризующие свойства объекта или составляющих его элементов. В процессе исследования объекта ряд параметров может изменяться, поэтому они называются переменными, которые в свою очередь подразделяются на переменные состояния и переменные управления. Как правило, переменные состояния объекта являются функцией переменных управления и воздействий внешней среды.

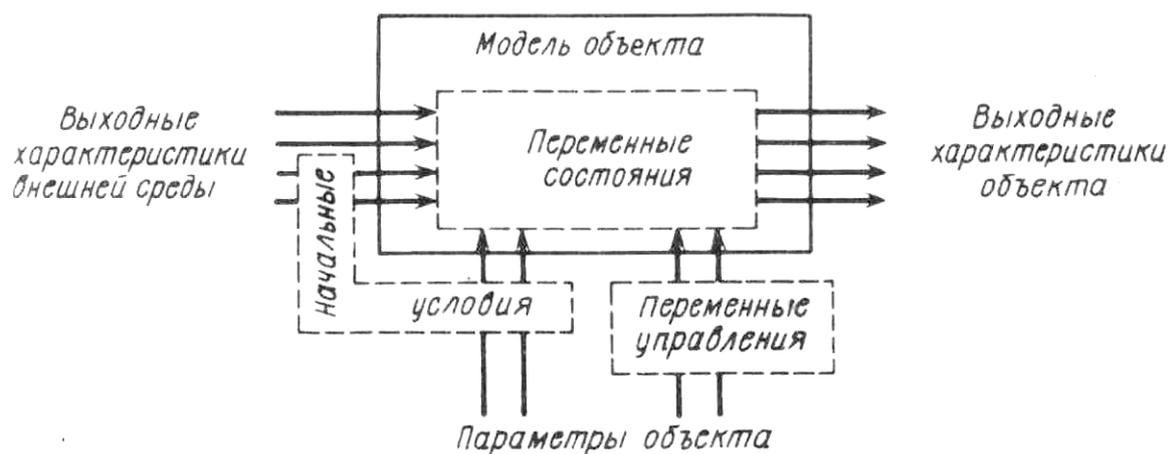


Рис. 1.1. Классификация факторов по их роли в ЭММ

Характеристиками (выходными характеристиками) называются интересующие исследователя непосредственные конечные результаты функционирования объекта (естественно, что выходные характеристики являются переменными состояниями).

Соответственно характеристики внешней среды описывают свойства внешней среды, которые сказываются на процессе и результате функционирования объекта. Значения ряда факторов, определяющие начальное состояние объекта или внешней среды, называются начальными условиями.

1.2. Критерии, ограничения, уравнения связи, решение модели

При рассмотрении ЭММ оперируют следующими понятиями: критерий оптимальности, целевая функция, система ограничений, уравнения связи, решение модели.

Критерием оптимальности называется некоторый показатель, имеющий экономическое содержание, служащий формализацией конкретной цели управления и выражаемый при помощи целевой функции через факторы модели. Критерий оптимальности определяет смысловое содержание целевой функции. В ряде случаев в качестве критерия оптимальности может выступать одна из выходных характеристик объекта.

Целевая функция математически связывает между собой факторы модели, и ее значение определяется значениями этих величин. Со-

держательный смысл целевой функции придает только критерий оптимальности.

Не следует смешивать критерий оптимальности и целевую функцию. Так, например, критерий прибыли и критерий стоимости произведенной продукции могут описываться одной и той же целевой функцией:

$$F = \sum_{i=1}^n c_i x_i \rightarrow \max, \quad (1.1)$$

где $i = 1, n$ – номенклатура производимой продукции; x_i – объем выпуска i -й номенклатуры; c_i – прибыль от выпуска i -й номенклатуры, или стоимость единицы i -й номенклатуры, в зависимости от смысла критерия оптимальности.

При наличии нескольких критериев оптимальности каждый из них будет формализован своей частной целевой функцией F_k , где $k = 1, \dots, K$ – число критериев оптимальности. Такая *многокритериальная оптимизационная задача* не имеет строгого математического решения.

Один из известных подходов к решению подобных оптимизационных задач состоит в формировании новой целевой функции:

$$F = f(E_1, E_2, \dots, E_K) \rightarrow \text{extr}. \quad (1.2)$$

Однако эта целевая функция может уже не нести экономического смысла. Отметим, что сам способ конструирования новой целевой функции (т.е. вид функции f) неоднозначен и зависит от субъективных предпочтений исследователя.

Другой подход к решению многокритериальных оптимизационных задач состоит в нахождении так называемого *множества Парето* G_f – некоторой поверхности в K -мерном пространстве частных целевых функций. Каждая точка f^* , принадлежащая этой поверхности, обладает следующим свойством: не найдется другой такой точки множества допустимых значений G_f , которая была бы не хуже f^* по всем показателям и превосходила ее хотя бы по одному показателю. Можно сформулировать данное определение «от обратного»: точка f^o из K -мерного пространства частных целевых функций не является оптимальной по Парето, если найдется хотя бы одна другая точка f^* , которая не хуже f^o по $K - 1$ частной целевой функции, а по одной частной целевой функции превосходит ее.

На рис. 1.2 (случай двух максимизируемых частных целевых функций f_1 и f_2) замкнутой линией представлена область G_f допустимых значений целевых функций $\{f_1, f_2\}$. Двойными линиями отмечены решения, оптимальные по Парето. На этом же рисунке изображена одна из эффективных (оптимальных по Парето) точек $f^* \{f_1^*, f_2^*\}$. В отличие от нее точка $\{f_1^0, f_2^0\}$ не является оптимальной по Парето, поскольку перемещение из точки $\{f_1^0, f_2^0\}$ к точке $\{f_1^*, f_2^*\}$ сопровождается улучшением значений обеих целевых функций.

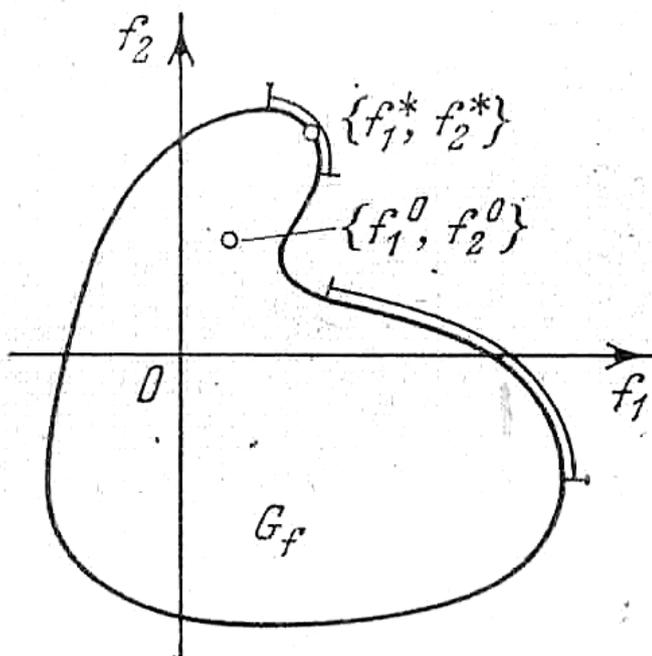


Рис. 1.2. Множество допустимых значений целевых функций и оптимальное по Парето множество

Система ограничений определяет пределы, сужающие область осуществимых, приемлемых или допустимых решений и фиксирующие основные внешние и внутренние свойства объекта. Ограничения определяют область протекания процесса, пределы изменения параметров и характеристик объекта.

Уравнения связи являются математической формализацией системы ограничений. Между понятиями «система ограничений» и «уравнения связи» существует точно такая же аналогия, как и между понятиями «критерий оптимальности» и «целевая функция»: различные по смыслу ограничения могут описываться одинаковыми уравнениями связи, а одно и то же ограничение в разных моделях может записываться различными уравнениями связи.

Таким образом, именно критерий оптимальности и система ограничений в первую очередь определяют концепцию построения будущей математической модели, т.е. концептуальную модель, а их формализация, т.е. целевая функция и уравнения связи, представляет собой математическую модель.

Решением математической модели называется такой набор (совокупность) значений переменных, который удовлетворяет ее уравнениям связи. Решения, имеющие экономический смысл, называются структурно допустимыми. Модели, имеющие много решений, называются вариантными в отличие от безвариантных, имеющих одно решение. Среди структурно допустимых решений вариантной модели, как правило, находится одно решение, при котором целевая функция в зависимости от смысла модели имеет наибольшее или наименьшее значение. Такое решение, как и соответствующее значение целевой функции, называется оптимальным (в частности, наибольшим или наименьшим).

По характеру цели исследования ЭММ делятся на *оптимизационные* (нормативные) и *описательные* (дескриптивные или ЭММ прямого счета).

Характерной чертой оптимизационных моделей является наличие одной или нескольких целевых функций. При этом в первом случае оптимизационные ЭММ называются монокритериальными, а во втором – многокритериальными. В общем виде монокритериальная ЭММ может быть представлена следующей системой отношений:

$$E = f(x_1, x_2, \dots, x_i, \dots, x_n; a_1, a_2, \dots, a_h, \dots, a_q) \rightarrow \text{extr}; \quad (1.3)$$

$$g_i(x_1, x_2, \dots, x_i, \dots, x_n; a_1, a_2, \dots, a_h, \dots, a_q) \{ \leq, \geq, <, >, = \} 0, \quad (1.4)$$

где E – критерий оптимальности объекта; x_i – управляемые переменные, $i = 1, \dots, n$; a_h – неуправляемые факторы модели, $h = 1, \dots, q$; g_j – уравнения связи, представляющие собой формализацию системы ограничений, $j = 1, \dots, m$; f – целевая функция – формализованное выражение критерия оптимальности.

Выражение $\{ \leq, \geq, <, >, = \}$ означает, что в ограничениях может стоять любое из приведенных в фигурных скобках логических условий.

Решение модели, заданной соотношениями (1.3) и (1.4), заключается в нахождении совокупности значений переменных

$$x^* = \{x_1^*, \dots, x_n^*\},$$

обращающей в максимум или минимум целевую функцию E при заданных уравнениях связи g_i .

В зависимости от того, содержит ли экономико-математическая модель (ЭММ) случайные факторы, она может быть отнесена к классу стохастических или детерминированных. В *детерминированных* моделях ни целевая функция f , ни уравнения связи g_i не содержат случайных факторов. Следовательно, для данного множества входных значений модели на выходе может быть получен только один-единственный результат. Для *стохастических* ЭММ характерно наличие среди факторов a_n модели, описываемой соотношениями (1.3) и (1.4), таких, которые имеют вероятностную природу и характеризуются какими-либо законами распределения, причем среди функций f и g_i могут быть и случайные функции. Значения выходных характеристик в таких моделях могут быть предсказаны только в вероятностном смысле. Реализация стохастических ЭММ в большинстве случаев осуществляется на ПК методами имитационного статистического моделирования.

Следующим признаком, по которому можно различать ЭММ, является связь с фактором времени. Модели, в которых входные факторы, а следовательно, и результаты моделирования явно зависят от времени, называются *динамическими*, а модели, в которых зависимость от времени t либо отсутствует совсем, либо проявляется слабо или неявно, называются *статическими*. Интересны в этом отношении имитационные модели: по механизму функционирования они являются динамическими (в модели идет имитация работы объекта в течение некоторого периода времени), а по результатам моделирования – статическими (например, ищется средняя производительность объекта за моделируемый период времени).

Статические модели представляют собой известную степень приближения к реальным объектам и системам, функционирующим во времени. Во многих случаях степень такого приближения, проявляющаяся в допущениях о неизменности или различного рода усреднениях факторов во времени (косвенно или приблизительно учитывающих фактор времени в определенных границах его изменения), является достаточной для практического применения статических моделей.

1.3. Принципы построения экономико-математических моделей

Современное промышленное предприятие, являющееся объектом моделирования, обладает исключительной сложностью. Процесс изготовления изделий характеризуется прежде всего движением во времени и пространстве огромного числа материальных, трудовых и информационных потоков, связанных с подготовкой производства, доставкой сырья и полуфабрикатов, выполнением множества взаимосвязанных операций по обслуживанию производства, хозяйственно-финансовому обеспечению, сбыту и реализации продукции. При этом поведение производственной системы не может быть оценено каким-либо одним показателем.

Не менее сложны и другие экономические объекты моделирования, например банки, страховые компании, торговые предприятия.

Ввиду чрезвычайной сложности рассматриваемых экономических систем моделирование их деятельности с помощью единой экономико-математической модели невозможно по двум основным причинам: из-за огромной размерности модели и неустойчивости ее решений к многочисленным реально существующим возмущениям. Поэтому целесообразно строить систему взаимосвязанных экономико-математических моделей, моделирующих деятельность отдельных подсистем и используемых при построении общей модели.

Общие принципы системного экономико-математического моделирования вытекают из общих принципов системного анализа, т.е. они должны являться ответами на следующие вопросы: 1) что должно быть сделано; 2) когда должно быть сделано; 3) при помощи кого должно быть сделано; 4) на основе какой информации осуществляется действие; 5) какой результат должен быть получен в результате действий.

Поэтому в качестве общих принципов системного экономико-математического моделирования целесообразно принять: 1) принцип достаточности используемой информации; 2) принцип инвариантности используемой информации; 3) принцип преемственности моделей; 4) принцип эффективной реализуемости комплекса экономико-математических моделей.

Рассмотрим более подробно каждый из них.

Принцип достаточности используемой информации означает, что в каждой частной модели должна использоваться только та информация, которая известна с требуемой для результатов моделирования точностью. Под известной информацией понимаются нормативные, справочные и прочие данные о реальной производственной системе, имеющиеся к моменту моделирования, точность которых можно оценить. В связи с последовательной разработкой комплекса моделей, характеризующей сложный объект, к моменту решения некоторой задачи, формализуемой частной моделью, вся информация о моделируемой системе может быть еще не известна.

Принцип инвариантности информации требует, чтобы используемая в модели входная информация была независима от параметров моделируемой системы, которые еще не известны на данной стадии исследования. Использование этого принципа позволяет избежать при построении экономико-математических моделей нередко встречающегося замкнутого круга, когда в модели используется информация, которая может быть известна лишь по результатам моделирования. Например, достаточно распространенной является модель определения длительности производственного цикла, в которой размеры партии деталей считаются известными, в то время как для выбора размеров партий необходимы данные о длительности производственного цикла.

Применение принципов достаточности и инвариантности приводит к формированию иерархии экономико-математических моделей для сложного объекта, позволяет строго определить входные параметры уравнения связи и целевые функции, формализующие критерии оптимальности и ограничения для каждой частной модели.

Суть *принципа преемственности моделей* сводится к тому, что каждая последующая модель не должна нарушать свойств объекта, установленных или отраженных в предыдущих моделях комплекса. Следовательно, выбор критериев и модели должен в первую очередь базироваться на принципе преемственности при условии, что обеспечивается выполнение принципов достаточности и инвариантности используемой информации. Если же последующая модель не является преемственной предыдущим моделям (а это зачастую бывает из-за использования при ее построении новой, дополнительной информации), то ранее построенные модели должны быть скорректированы для обеспечения принципа преемственности.

Для выполнения *принципа эффективной реализуемости комплекса моделей* необходимо, чтобы каждая частная модель могла быть реализована при помощи современных и характерных для большинства предприятий вычислительных систем. Выполнение этого принципа требует обеспечения соответствия точности исходных данных, точности решения задачи и той точности результирующей информации, которая достаточна для практических целей.

Как показывают проводимые исследования, вследствие неточности используемых данных, а также наличия определенной погрешности вычисляемых по моделям показателей поиск точного оптимума оказывается бессмысленным и, более того, иллюзорным. Обычно, исходя из погрешности исходных данных, погрешности применяемых вычислительных методов и конкретных свойств оптимизируемых показателей, можно найти погрешность целевой функции, в пределах которой все планы могут считаться оптимальными.

1.4. Процедура построения и реализации модели на ЭВМ

Процедуру построения модели и подготовки управленческого решения с помощью моделирования на ПК можно представить состоящей из ряда этапов (рис. 1.3), хотя в конкретных случаях некоторые этапы могут опускаться, а ряд работ по построению модели – вестись параллельно.

Содержательной основой для построения математической модели объекта является его *концептуальная модель*. Под концептуальной моделью объекта понимается совокупность качественных зависимостей критериев оптимальности и различного рода ограничений от факторов, существенных для отражения функционирования объекта. Концептуальная модель отражает следующие основные моменты:

- условия функционирования объекта, определяемые характером взаимодействий объекта и его окружения, элементов объекта;
- цели исследования объекта и направления улучшения его функционирования;
- возможности управления объектом, определяющие состав управляемых переменных объекта.



Рис. 1.3. Этапы построения и исследования ЭММ

В связи с этим при формулировке концептуальной модели объекта возникают проблемы:

- составления упрощенного и в то же время адекватного поставленной цели описания исследуемой ситуации – сценария функционирования объекта;
- формулировки и уточнения формулировки целей, стоящих перед объектом при его функционировании;
- формализации целей в критерии оптимальности;
- формализации внешних и внутренних ограничений;
- выбора факторов, описывающих объект и его окружение, которые должны быть учтены в исследовании и соответственно включены в математическую модель;

- классификации факторов и выделения из них в первую очередь управляемых переменных.

Построение концептуальной модели – важнейший этап моделирования, так как именно оно определяет направление, цели, область исследования. Заключительным шагом построения концептуальной модели является оценка ее адекватности исследуемой ситуации.

На следующем этапе на основе концептуальной модели **формируется математическая модель объекта**. Главная проблема этого этапа – определение количественных, математических соотношений, формализующих качественные зависимости концептуальной модели. Даже при наличии детально проработанного сценария эти соотношения могут быть неочевидны. В связи с этим зачастую возникает необходимость в выполнении промежуточного этапа между построением концептуальной и математической модели объекта – преобразования сценария в алгоритм, описывающий (моделирующий) взаимодействие элементов между собой и с окружением во времени.

Для реализации математической модели на ЭВМ она должна быть представлена численно, т.е. заданы числовые значения констант, диапазоны изменения неопределенных факторов и управляемых переменных, законы распределения случайных величин. При этом зачастую возникают проблемы эффективного представления числовой информации в математической модели для ее реализации на ЭВМ, например сжатия табличной информации методами интерполяции, аппроксимации и экстраполяции, обработки статистических данных для получения формы и характеристик законов распределения случайных величин, получения эмпирических зависимостей между факторами модели и т.п.

Заключительным шагом формирования математической модели является оценка ее адекватности по отношению к концептуальной модели.

Этап **исследования математической модели** начинается с ее анализа (отнесения к тому или иному классу) и выбора соответствующего метода ее решения. Главная проблема этого этапа – разработка алгоритма поиска решения на модели, действительно обеспечивающего поиск оптимального или наилучшего в заданных условиях решения в пределах времени счета ПК.

Заключительным шагом последнего этапа является оценка точности полученного на модели результата.

1.5. Цели построения экономико-математических моделей

Цель исследования определяет цель построения модели. Модели могут строиться для следующих целей:

1. *Выявление функциональных соотношений* – определение количественных зависимостей между входными факторами модели x , a и выходными характеристиками исследуемого объекта ω . Подобного рода модели по своему характеру являются описательными; задача выявления функциональных соотношений присутствует при построении математических моделей любых типов.

2. *Анализ чувствительности* – установление из большого числа действующих факторов тех, которые в большей степени влияют на интересующие исследователя выходные характеристики. Модели анализа чувствительности должны обязательно предусматривать возможность варьирования интересующими исследователя факторами: характеристиками внешней среды, начальными условиями, переменными управления. Помимо основной математической модели – системы функциональных соотношений – они могут включать модели входов, обеспечивающие целенаправленную вариацию входных факторов, и модель выхода, проводящую обработку результатов вычислений и дающую итоговую информацию о степени чувствительности модели к тому или иному фактору.

3. *Прогноз* – оценка поведения объекта при некотором предполагаемом сочетании внешних условий. Обычно задачи прогноза являются динамическими относительно входов, и в качестве независимой (неуправляемой) переменной в них выступает время.

4. *Оценка* – определение, насколько хорошо исследуемый объект будет соответствовать некоторым критериям. В отличие от рассмотренных выше типов моделей, модели оценки включают расчеты интересующих исследователя интегральных характеристик – критериев, формализующих цели исследования. Модели оценки занимают промежуточное положение между описательными и оптимизационными моделями – в них заданы критерий и его некоторое «критическое» значение, но проводится не оптимизация, а лишь сравнение расчетного значения с «критическим», после чего принимается решение об удовлетворении характеристик объекта предъявленным требованиям.

5. *Сравнение* – сопоставление ограниченного числа альтернативных вариантов систем или же сопоставление нескольких предлагаемых принципов или методов действия. Задача сравнения предусматривает оценку каждого варианта по одному или нескольким критериям и дальнейший выбор наилучшего. Число вариантов предполагается незначительным, в связи с чем оцениваются все варианты (прямой перебор). Хотя модели этого типа близки к оптимизационным, специальный блок оптимизации в них отсутствует.

6. *Оптимизация* – точное определение такого сочетания переменных управления, при котором обеспечивается экстремальное (максимальное или минимальное в зависимости от смысла критерия оптимальности) значение целевой функции. Главное отличие от вышерассмотренного случая – наличие специального блока оптимизации, позволяющего целенаправленно и наиболее эффективно с вычислительной точки зрения просматривать множество альтернативных вариантов, число которых зачастую близко к бесконечности.

Вопросы для самоконтроля

1. Как классифицируются экономико-математические модели по целевому назначению?
2. Дайте определение параметров и характеристик модели объекта.
3. Чем отличается критерий оптимальности от целевой функции?
4. Какие методы решения многокритериальных оптимизационных задач вам известны?
5. Поясните оптимальное по Парето решение оптимизационной задачи.
6. Какие принципы экономико-математического моделирования целесообразно применять при построении модели экономического объекта?
7. Каковы цели построения экономико-математических моделей?



**ПРЕДПОСЫЛКИ СОЗДАНИЯ
ИМИТАЦИОННЫХ МОДЕЛЕЙ**

2.1. Границы возможностей классических математических методов в экономике

Классические оптимизационные методы и методы математического программирования, которые рассматривались в предыдущей теме, в принципе являются мощным аналитическим средством. Однако число реальных задач, которые можно сформулировать так, чтобы не возникало противоречий между предположениями, лежащими в основе этих методов, сравнительно невелико. В связи с этим аналитические модели, в первую очередь модели математического программирования, до настоящего времени не стали практическим инструментом управленческой деятельности. По большей части они широко используются в высшем экономическом образовании. Аналитические методы имеют определенную познавательную ценность, они позволяют в сжатом и точном виде выражать экономические закономерности, но без учета многих важных подробностей. А, как гласит известная поговорка, «дьявол скрывается в деталях». Именно для учета таких важных деталей особенно полезно имитационное моделирование.

Развитие вычислительной техники породило новое направление в исследовании сложных процессов – *имитационное моделирование*. Имитационные модели, являющиеся особым классом математических моделей, принципиально отличаются от аналитических тем, что персональные компьютеры в их реализации играют одну из самых важнейших ролей.

2.2. Понятие имитационного моделирования. Особенности имитационных моделей

Современные ПК обладают колоссальным быстродействием и памятью, развитыми внешними устройствами, эффективным системным и прикладным программным обеспечением. Все это дает возможность эффективно организовать диалог человека и машины в рамках имитационной системы.

Идея метода имитационного моделирования состоит в том, что вместо аналитического описания связей между входами, состояниями и выходами строят алгоритм, отображающий последовательность развития процессов внутри исследуемого объекта, а затем «проигрывают» поведение объекта на ПК.

Имитационная система – совокупность модели, имитирующей изучаемое явление, и систем внешнего и внутреннего обеспечения. Имитационная модель – вычислительная процедура, формализованно описывающая изучаемый объект и имитирующая его поведение. При ее составлении нет необходимости упрощать описание явления, отбрасывая порой даже существенные детали, чтобы втиснуть его в рамки модели, удобной для применения тех или иных известных математических методов анализа. Для имитационного моделирования характерна имитация элементарных явлений, составляющих исследуемый процесс, с сохранением их логической структуры, последовательности протекания во времени, характера и состава информации о состояниях процесса. Модель по своей форме является логико-математической (алгоритмической).

2.3. Классификация моделей имитационного моделирования

Имитационные модели как подкласс математических моделей можно классифицировать на: статические и динамические; детерминированные и стохастические; дискретные и непрерывные.

Класс задачи предъявляет определенные требования к имитационной модели. Так, например, при статической имитации расчет повторяется несколько раз в различных условиях проведения эксперимента – исследование поведения «в определенный короткий период време-

ни». При динамической имитации моделируется поведение системы «в течение продолжительного периода времени» без изменений условий. При стохастической имитации в модель включаются случайные величины с известными законами распределения; при детерминированной имитации эти возмущения отсутствуют, т.е. их влияние не учитывается.

Порядок построения имитационной модели и ее исследования в целом соответствует схеме построения и исследования аналитических моделей. Однако специфика имитационного моделирования приводит к ряду специфических особенностей выполнения тех или иных этапов. Рекомендуется следующий порядок имитации:

1. Определение системы – установление границ, ограничений и измерителей эффективности системы, подлежащей изучению.

2. Формулирование модели – переход от реальной системы к некоторой логической схеме (абстрагирование).

3. Подготовка данных – отбор данных, необходимых для построения модели, и представление их в соответствующей форме.

4. Трансляция модели – описание модели на языке, приемлемом для используемого ПК.

5. Оценка адекватности – повышение до приемлемого уровня степени уверенности, с которой можно судить относительно корректности выводов о реальной системе, полученных на основании обращения к модели.

6. Стратегическое планирование – планирование эксперимента, который должен дать необходимую информацию.

7. Тактическое планирование – определение способа проведения каждой серии испытаний, предусмотренных планом эксперимента.

8. Экспериментирование – процесс осуществления имитации с целью получения желаемых данных и анализа чувствительности.

9. Интерпретация – построение выводов по данным, полученным путем имитации.

10. Реализация – практическое использование модели и (или) результатов моделирования.

11. Документирование – регистрация хода осуществления проекта и его результатов, а также документирование процесса создания и использования.

Вопросы для самоконтроля

1. Перечислите основные отличия имитационного моделирования от других математических методов моделирования экономических объектов.
2. Почему применение персонального компьютера является принципиально необходимым при имитационном моделировании?
3. Как классифицируются имитационные модели?
4. Почему для имитационного моделирования экономических систем применяется дискретный подход?
5. Каков порядок построения имитационной модели и ее исследования?

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ
ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ****3.1. Метод Монте-Карло как математическая основа имитационного моделирования**

Многим специалистам термин «метод Монте-Карло» иногда представляется синонимом термина «имитационное моделирование», что в общем случае неверно. Имитационное моделирование – это более широкое понятие, и метод Монте-Карло является важным, но далеко не единственным методическим компонентом имитационного моделирования.

Согласно методу Монте-Карло, проектировщик может моделировать работу тысячи сложных систем, управляющих тысячами разновидностей подобных процессов, и исследовать поведение всей группы, обрабатывая статистические данные. Другой способ применения этого метода заключается в том, чтобы моделировать поведение системы управления на очень большом промежутке модельного времени (несколько лет), причем астрономическое время выполнения моделирующей программы на компьютере может составить доли секунды. Рассмотрим метод Монте-Карло подробнее.

В различных задачах, встречающихся при создании сложных систем, могут использоваться величины, значения которых определяются случайным образом.

Примерами таких величин являются:

- случайные моменты времени, в которые поступают заказы на фирму;
- загрузка производственных участков или служб объекта экономики;

- внешние воздействия (требования или изменения законов, платежи по штрафам и др.);
- оплата банковских кредитов;
- поступление средств от заказчиков.

В качестве соответствующих им переменных могут использоваться число, совокупность чисел, вектор или функция. Одной из разновидностей метода Монте-Карло при численном решении задач, включающих случайные переменные, является метод статистических испытаний, который заключается в моделировании случайных событий.

Метод Монте-Карло основан на статистических испытаниях и по природе своей является экстремальным, может применяться для решения полностью детерминированных задач, таких как обращение матриц, решение дифференциальных уравнений в частных производных, отыскание экстремумов и численное интегрирование. При вычислениях методом Монте-Карло статистические результаты получаются путем повторяющихся испытаний. Вероятность того, что эти результаты отличаются от истинных не более чем на заданную величину, есть функция количества испытаний.

В основе вычислений по методу Монте-Карло лежит случайный выбор чисел из заданного вероятностного распределения. При практических вычислениях эти числа берут из таблиц или получают путем некоторых операций, результатами которых являются псевдослучайные числа с теми же свойствами, что и числа, получаемые путем случайной выборки. Имеется большое число вычислительных алгоритмов, которые позволяют получить длинные последовательности псевдослучайных чисел.

Применение метода Монте-Карло может дать существенный эффект при моделировании развития процессов, натурное наблюдение которых нежелательно или невозможно, а другие математические методы применительно к этим процессам либо не разработаны, либо неприемлемы из-за многочисленных оговорок и допущений, которые могут привести к серьезным погрешностям или неправильным выводам.

Основной вопрос, который возникает при анализе полученных результатов методом Монте-Карло, – каков объем выборки необходим, чтобы получить достоверные результаты? Обозначим через Π случайную величину, которая рассчитывается однократным расчетом по методу имитационного моделирования. Нас интересует ее среднее

значение (вернее, оценка среднего значения по выборке), а также близость выборочного среднего к действительному, соответствующему генеральной совокупности.

В табл. 3.1 в качестве примера приведены данные о выборочном среднем значении \bar{P} (будем обозначать среднее значение, определенное по генеральной совокупности P через \bar{P} , выборочное среднее значение через \tilde{P}), дисперсии (обозначаем через σ^2 , выборочную дисперсию через $\tilde{\sigma}^2$), среднеквадратичном отклонении (обозначаем через S , выборочное среднеквадратичное отклонение обозначаем через \tilde{S}) в зависимости от числа испытаний (объем выборки) N .

Таблица 3.1

Данные о результатах статистической обработки выборки

N	\tilde{P}	$\tilde{\sigma}^2$	\tilde{S}	δ	δ / \tilde{P}
10	39,41	1,53	1,24	0,89	0,022
20	38,99	1,05	1,025	0,48	0,0123
50	38,96	1,81	1,34	0,382	0,0098
100	39,09	1,64	1,281	0,25	0,0064
150	39,05	1,53	1,24	0,20	0,0051
200	39,083	1,47	1,212	0,169	0,0043
250	39,11	1,37	1,17	0,145	0,0037

В качестве меры достоверности полученного в расчетах выборочного среднего \tilde{P} будем рассматривать доверительный интервал δ на уровне значимости 95%. Смысл его состоит в том, что вероятность того, что истинное среднее значение \bar{P} (среднее значение для генеральной совокупности всех значений случайной величины) с вероятностью 0,95 находится в интервале от $\tilde{P} - \delta$ до $\tilde{P} + \delta$,

где $\delta = t_{N-1;0,975} \sqrt{\frac{\tilde{\sigma}^2}{N}}$.

Это означает, что \bar{P} находится в диапазоне:

$$\tilde{P} \pm t_{N-1;0,975} \sqrt{\frac{\tilde{\sigma}^2}{N}}. \quad (3.1)$$

Входящий в (3.1) параметр $t_{N-1;0,975}$ – известный t – критерий Стьюдента. Имеются соответствующие таблицы значений этого критерия. При применении пакетов статистических программ или программы «Анализ данных» табличного процессора EXCEL расчет этого критерия производится автоматически. Также можно использовать пакет программ STATISTICA.

Поясним расчет доверительного интервала, используя данные из

табл. 3.1. При $N = 200$:
$$\sqrt{\frac{\tilde{\sigma}^2}{N}} = \sqrt{\frac{1,47}{200}} = 0,0857, \quad t_{N-1;0,975} = 1,96.$$

Следовательно, $t_{N-1;0,975} \sqrt{\frac{\tilde{\sigma}^2}{N}} = \delta = 0,169.$

Это означает, что истинное среднее значение \bar{K} (соответствующее ее генеральной совокупности) случайной величины K находится в интервале, определенном соотношением (3.1), с вероятностью 0,95.

3.2. Формула Поллачека-Хинчина

Большинство задач, решаемых методом имитационного моделирования, относится к так называемой «теории массового обслуживания». Эта теория исследует процессы обслуживания некоторым устройством (например, станком, кассиром супермаркета и т.п.) потока заявок на обслуживание. В имитационном моделировании такие запросы на обслуживание носят название «транзактов». Фактически теория массового обслуживания явилась предтечей методов имитационного моделирования.

Среднее время задержки запросов в очереди на обслуживание вычисляется по формуле Поллачека-Хинчина:

$$t_q = \frac{t_s \rho (1 + c_s^2)}{2(1 - \rho)}, \quad (3.2)$$

где t_q – искомая средняя задержка в очереди, t_s – среднее время обслуживания, c – загрузка обслуживающего узла ($c = \frac{t_s}{t_g} \leq 1$), c_s – коэффициент вариации времени обслуживания.

Если известно среднее квадратичное отклонение времени обслуживания σ_s , то $c_s = \frac{\sigma_s}{t_s}$.

Формула Поллачека-Хинчина полезна при отладке программ имитационного моделирования.

3.3. Формализованная схема процесса имитационного моделирования

Для имитационного моделирования процесса на ПК необходимо преобразовать его математическую модель в специальный моделирующий алгоритм, в соответствии с которым в ПК будет вырабатываться информация, описывающая элементарные явления исследуемого процесса с учетом их связей и взаимных влияний. Определенная часть циркулирующей информации выводится и используется для определения тех характеристик процесса, которые требуется получить в результате моделирования (рис. 3.1).

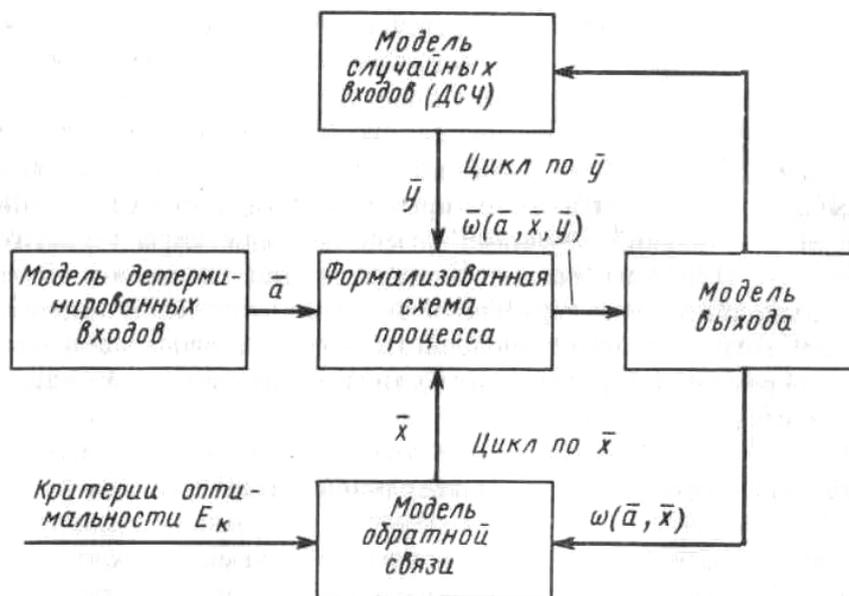


Рис. 3.1. Структура моделирующего алгоритма для оптимизационной модели со случайными факторами

Центральным звеном моделирующего алгоритма является собственно имитационная модель – формализованная схема процесса. Формализованная схема представляет собой формальное описание процедуры функционирования сложного объекта в исследуемой опе-

рации и позволяет для любых задаваемых значений входных факторов модели (переменных \bar{x} , детерминированных \bar{a} , случайных \bar{y}) просчитать соответствующие им числовые значения выходных характеристик \bar{w} .

Остальные блоки модели (рис. 3.1) представляют собой внешнее математическое обеспечение процесса имитации.

Модели входов обеспечивают задание тех или иных значений входных факторов. Статические модели детерминированных входов элементарны: это массивы значений констант, соответствующих определенным факторам модели. Динамические модели входов обеспечивают изменение значений детерминированных факторов во времени по известному закону $\bar{a}(t)$.

Модели случайных входов (иначе – датчики случайных чисел) имитируют поступление на вход изучаемого объекта случайных воздействий с заданными (известными) законами распределения $p(\bar{y})$. Динамические модели случайных входов учитывают, что законы распределения случайных величин являются функциями времени, т.е. для каждого периода времени либо форма, либо характеристика закона распределения (например, математическое ожидание, дисперсия и т.д.) будут своими.

В связи с тем, что результат, полученный при воспроизведении единственной реализации из-за наличия случайных факторов, не может характеризовать исследуемый процесс в целом, приходится анализировать большое число таких реализаций, так как только тогда по закону больших чисел получаемые оценки приобретают статистическую устойчивость и могут быть с определенной точностью приняты за оценки искомых величин. Модель выхода обеспечивает накопление, обработку и анализ полученного множества случайных результатов. Для этого с ее помощью организуется многократный просчет значений выходных характеристик при постоянных значениях факторов \bar{a} , \bar{x} и различных значениях случайных факторов \bar{y} (в соответствии с заданными законами распределения) – «цикл по y ». В связи с этим модель выхода включает программы тактического планирования эксперимента на ПК – определение способа проведения каждой серии прогонов, соответствующей конкретным значениями \bar{a} и \bar{x} . Кроме того, модель решает задачу обработки случайных значений выходных характеристик, в результате которой они «очищаются» от

влияния случайных факторов и поступают на вход модели обратной связи, т.е. модель выхода реализует сведение стохастической задачи к детерминированной методом «осреднения по результату».

3.4. Принципы построения имитационных моделирующих алгоритмов

Имитационная модель является, как правило, динамической моделью, отражающей последовательность протекания элементарных процессов и взаимодействие отдельных элементов по оси «модельного» времени t^M .

Процесс функционирования объекта в течение некоторого интервала времени T можно представить как случайную последовательность дискретных моментов времени t_i^M . В каждый из этих моментов происходят изменения состояния элементов объекта, а в промежутке между ними никаких изменений состояния не происходит.

При построении формализованной схемы процесса должно выполняться следующее рекуррентное правило: событие, происходящее в момент времени t_i^M , может моделироваться только после того, как промоделированы все события, произошедшие в момент времени t_{i-1}^M . В противном случае результат моделирования может быть неверным.

Реализация этого правила может проводиться различными способами.

1. Повременное моделирование с детерминированным шагом («принцип Δt »). При повременном моделировании с детерминированным шагом алгоритм одновременно просматривает все элементы системы через достаточно малые промежутки времени (шаг моделирования) и анализирует все возможные взаимодействия элементов. Для этого определяется минимальный интервал времени, в течение которого не может измениться состояние ни одного из элементов системы; детализированная величина Δt принимается за шаг моделирования.

Способ моделирования с детерминированным шагом состоит из совокупности многократно повторяющихся действий:

а) на i -м шаге в момент t_i просматриваются все элементы объекта и определяется, какие из них изменяют свое состояние в этот момент;

б) моделируются все те изменения состояния, которые происходят в момент t_i ;

в) производится переход к $i+1$ -му шагу, который выполняется в момент $t_{i+1} = t_i + \Delta t$.

2. Повременное моделирование со случайным шагом (моделирование по «особым» состояниям). При рассмотрении большинства сложных систем можно обнаружить два типа состояний системы: 1) обычные (неособые) состояния, в которых система находится большую часть времени; 2) особые состояния, характерные для системы в некоторые моменты времени, совпадающие с моментами поступления в систему воздействий из окружения, выхода одной из характеристик системы на границу области существования и т.д. Например, станок работает – обычное состояние, станок сломан – особое состояние. Любое скачкообразное изменение состояния объекта может рассматриваться при моделировании как переход в новое «особое» состояние.

Повременное моделирование со случайным шагом (от события к событию) заключается в том, что моделирующий алгоритм осматривает модели элементов системы только в такие моменты времени, когда состояние исследуемой системы меняется. В те моменты времени, когда модель какого-либо элемента системы должна менять состояние, происходит осмотр модели именно этого элемента и с учетом взаимосвязей элементов корректируется состояние модели всей системы. Длительность шага – величина случайная. Этот способ отличается от «принципа Δt » тем, что включает процедуру определения момента времени, соответствующего ближайшему особому состоянию по известным характеристикам предыдущих состояний.

3. Позаявочный способ. При моделировании процессов обработки последовательно идущих заявок иногда удобно строить моделирующие алгоритмы позаявочным способом, при котором прослеживается прохождение каждой заявки (детали, носителя информации) от ее входа в систему и до выхода ее из системы. После этого алгоритм предусматривает переход к рассмотрению следующей заявки. Такого рода моделирующие алгоритмы весьма экономны и не требуют специальных мер для учета особых состояний системы. Однако этот способ может использоваться только в простых моделях в случаях последовательных заявок, не опережающих друг друга, так как в противном случае становится весьма затруднительным учет взаимодействия заявок, поступивших в систему.

3.5. Реализация имитационных моделей на ПК. Модели случайных входов

Создавая стохастическую имитационную модель, всегда приходится решать, следует ли в модели использовать имеющиеся эмпирические значения входных стохастических факторов непосредственно или целесообразно использовать их теоретико-вероятностные или частотные распределения. Непосредственное использование имеющихся статистических данных предполагает запись в память ПК и выборку их из памяти определенным образом, обеспечивающим «чистоту» статистического эксперимента. В общем случае применение теоретических частотных или вероятностных распределений с учетом требований к машинному времени и памяти более эффективно, чем использование табличных данных для получения значений случайных факторов, необходимых в работе с моделью.

В связи с этим на практике в большинстве случаев эмпирические данные обрабатываются для получения наиболее близкого к ним теоретического закона распределения данной случайной величины и дальнейшего воспроизведения этого закона в процессе моделирования.

Напомним необходимые сведения из курса теории вероятностей. Закон плотности распределения значений случайной величины y записывается в виде функции $f(y)$, что означает следующее: вероятность того, что случайная величина y примет значение из интервала $(y, y+dy)$, равна $f(y)dy$. Вероятность того, что случайная величина y примет любое допустимое значение, равна

$$\int_{-\infty}^{\infty} f(y)dy = 1. \quad (3.3)$$

Для введения в моделирующий алгоритм случайных факторов используются специальные модели имитации случайных величин и событий. Эти модели являются основополагающими в статистическом моделировании, так как практически все остальные действия сводятся к преобразованию полученных случайных чисел по соответствующим правилам. В качестве исходной последовательности для получения случайных чисел с любым законом распределения либо для моделирования случайных событий используются случайные числа, равномерно распределенные в интервале $(0;1)$. Наиболее распростра-

ненным способом получения таких чисел является программный. К настоящему времени создано большое количество разнообразных программ получения случайных чисел. В программное обеспечение современных ПК входят стандартные программы генерирования случайных чисел, равномерно распределенных в интервале (0;1). Например, в системе программирования Visual Basic имеется функция RND() для генерирования таких случайных чисел, а также функция RANDOMIZE() для смены последовательности случайных чисел. Более строго называть такие числа не случайными, а псевдослучайными, поскольку имеется периодичность (с очень большим периодом) появления таких чисел.

Разумеется, есть такие генераторы случайных чисел и в специальных системах имитационного моделирования, например в системе GPSS World, которая рассматривается в теме 5 и во второй части данного учебного пособия.

Преобразование случайных чисел с равномерным законом распределения в случайные числа с заданным (в общем случае произвольным) законом распределения базируется на теореме: если случайная величина y_i имеет плотность распределения $f(y)$, то случайная величина x_i :

$$x_i = \int_{-\infty}^{y_i} f(y) dy \quad (3.4)$$

является равномерно распределенной в интервале (0;1).

Решение уравнения (3.4) относительно y_i дает возможность получить искомое случайное число.

Точное решение интегрального уравнения (3.4) возможно только для ряда относительно простых законов распределения, например при равномерном распределении $f(y)$ в некотором интервале, а также в случае показательного распределения с параметром λ ($\lambda > 0$):

$$f(y) = \lambda e^{-\lambda y}. \quad (3.5)$$

Однако в большинстве практически важных случаев уравнение (3.4) относительно y_i не решается, поэтому развиты приближенные методы.

Что же касается моделей выходов, то они реализуют:

- накопление информации (обычно в отдельных файлах);

- обработку информации, т.е. приведение ее к форме, удобной для окончательного статистического анализа;
- анализ накопленной информации специальными методами, из которых важнейший – рассмотренный выше метод Монте-Карло.

3.6. Основные типы распределений случайных величин, применяемые в имитационном моделировании

Рассмотрим наиболее часто встречающиеся функции распределения случайных величин. Прежде всего заметим, что эти функции относятся либо к непрерывным, либо к дискретным случайным величинам (ниже случайные величины будем обозначать через x).

Равномерное распределение. Говорят, что непрерывная случайная величина x имеет равномерное распределение на отрезке от a до b , если ее плотность $f(x)$ на этом отрезке постоянна (рис. 3.2):

$$f(x) = \frac{1}{b-a} \text{ при } x \in (a,b) \text{ и } f(x) = 0 \text{ при } x \notin (a,b).$$

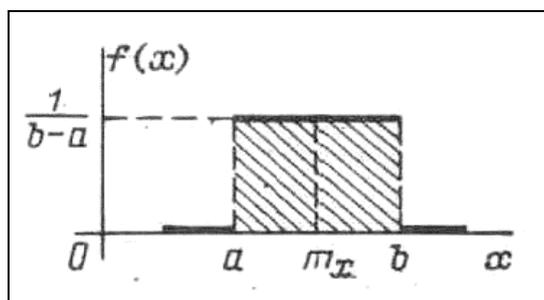


Рис. 3.2. График равномерного распределения

Среднее значение равномерно распределенной случайной величины равно:

$$m_x = \frac{a+b}{2}.$$

Нормальное распределение (иногда называемое законом Гаусса) играет исключительно важную роль в имитационном моделировании, поскольку часто встречается на практике. Говорят, что непрерывная случайная величина x распределена по нормальному закону с параметрами m , σ , если ее плотность распределения вероятности значений имеет вид (рис. 3.3):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}},$$

где m – среднее значение случайной величины; σ – среднее квадратическое отклонение случайной величины, равное корню квадратному из дисперсии D : $\sigma = \sqrt{D}$.

Максимальное значение $f(x)$ равно $1/\sigma\sqrt{2\pi}$ и достигается при $x = m$.

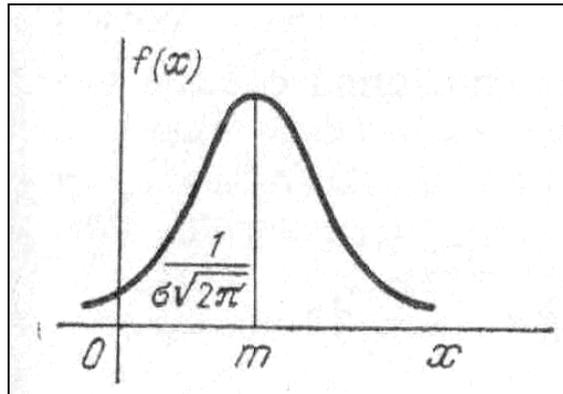


Рис. 3.3. График нормального распределения

Для получения случайных чисел, распределенных по нормальному закону, можно воспользоваться центральной теоремой теории вероятности, согласно которой сумма большого числа случайных чисел, имеющих равномерное распределение в интервале (0;1), дает распределение, асимптотически приближающееся к нормальному. Поэтому для получения нормального распределения следует суммировать случайные числа, равномерно распределенные в интервале (0;1), при этом среднее значение m и среднеквадратическое отклонение σ получаемого закона при достаточно большом числе случайных n будут определяться следующими выражениями:

$$m = n / 2; \quad \sigma = \sqrt{\frac{n}{6}}.$$

При $n = 8 - 12$ приближение к нормальному закону достаточно высокое для потребностей практики. Если необходимо смоделировать нормальный закон с параметрами математического ожидания m^* и среднеквадратического отклонения σ^* , то требуемые случайные числа x_k получаются по формуле:

$$x_k = \sqrt{3} \left(\frac{2}{\sqrt{n}} \sum_{i=1}^n x_i^* - \sqrt{n} \right) \sigma^* + m^*,$$

где $k = 1, 2, \dots$ – номер случайного числа с заданным нормальным законом распределения; x_i^* – равномерно распределенные на отрезке $(0; 1)$ случайные числа.

Распределение Пуассона дискретных случайных величин играет большую роль в практических задачах. Это распределение называют также вероятностным распределением *редких событий*.

Говорят, что случайная величина x имеет распределение Пуассона, если ее возможные значения $0, 1, 2, \dots, m$, а вероятность соответствующих значений выражается формулой:

$$P_m = \frac{a^m}{m!} e^{-a} \quad (m = 0, 1, 2, \dots).$$

Параметр a равен одновременно среднему значению и дисперсии случайной величины x . Графики вероятностного распределения Пуассона при различных значениях a представлены на рис. 3.4.

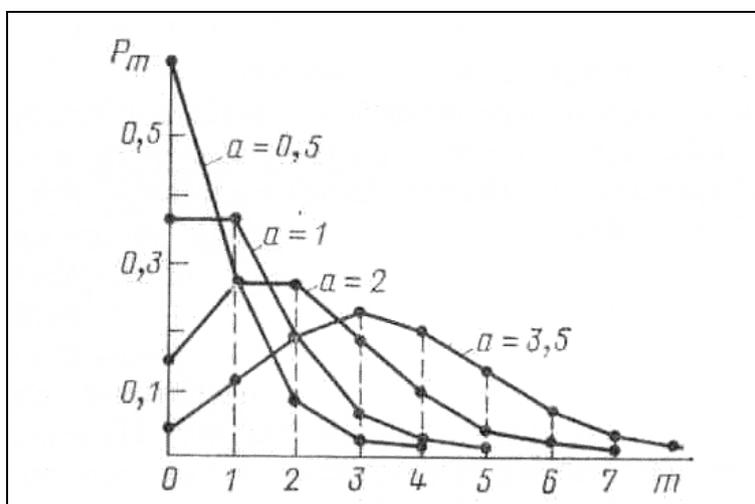


Рис. 3.4. График распределения Пуассона

В имитационном моделировании важное значение имеет так называемый *пуассоновский поток заявок* (к таким заявкам можно отнести, например, заявки на ремонт бытовой техники, телефонные звонки и т.п.). Пуассоновский поток описывается следующим образом: веро-

ятность поступления k заявок пуассоновского потока в течение интервала t составляет

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}.$$

В системах имитационного моделирования реализованы численные методы для генерирования случайных чисел, подчиняющихся любому заданному закону, в том числе закону Пуассона. Для получения случайного числа y_k с заданным законом распределения необходимо, получая случайные числа x_k , равномерно распределенные в интервале $(0;1)$, вычислять накопленные частоты попадания чисел y_k в интервалы D_{y_j} , на которые разбивается диапазон изменения y .

Экспоненциальное распределение. Это непрерывное распределение очень важно, поскольку ему подчиняются интервалы времени между соседними заявками пуассоновского потока.

Пусть x_j – равномерно распределенные на отрезке $(0;1)$ случайные числа. Тогда для случайных интервалов времен поступления пуассоновских заявок t_j имеем следующую формулу:

$$t_j = -\frac{1}{\lambda} \ln(x_j),$$

где $1/\lambda$ – средний интервал времени поступления заявок.

3.7. Моделирование случайных событий по экспериментальным данным

Часто наблюдающиеся на практике случайные события не подчиняются какому-либо теоретическому закону. Тогда моделирование базируется на использовании значений вероятности событий, найденных по данным наблюдения (эксперимента).

Например, пусть событие A_1 состоит в том, что вкладчик вносит в банк депозит сроком на один месяц, событие A_2 состоит в том, что вкладчик вносит в банк депозит сроком на три месяца, событие A_3 состоит в том, что вкладчик вносит в банк депозит сроком на шесть месяцев, событие A_4 состоит в том, что вкладчик вносит в банк депозит сроком на один год. Наблюдения в течение одного месяца показали, что срок, на который вносится депозит, является случайной величи-

ной, причем на срок один месяц внесено 63 депозита, на срок 3 месяца внесено 106 депозитов, на срок 6 месяцев внесено 141 депозит, на срок один год внесено 65 депозитов. Легко видеть, что вероятность события A_1 можно оценить так: $p_1 = \frac{63}{63 + 106 + 141 + 65} = 0,168$. Аналогичным образом нетрудно найти, что $p_2 = 0,282$, $p_3 = 0,376$, $p_4 = 0,173$.

Для моделирования данной полной группы событий A_1, A_2, A_3, A_4 , наступающих с вероятностями p_1, p_2, p_3, p_4 , причем $\sum_{j=1}^4 p_j = 1$, используется следующий алгоритм.

Генерируется очередное случайное число x_i , равномерно распределенное на отрезке $(0,1)$. Далее проверяем условие

$$f_{r-1} < x_i \leq f_r, \quad (3.6)$$

где $f_r = \sum_{j=1}^r p_j$, $1 \leq r \leq 4$.

Выполнению условия (3.6) соответствует свершение события A_r . В рассматриваемом примере условия свершения событий таковы:

$x_i \leq 0,168$ – совершается событие A_1 ,

$0,168 < x_i \leq 0,45$ – совершается событие A_2 ,

$0,45 < x_i \leq 0,826$ – совершается событие A_3 ,

$x_i > 0,826$ – совершается событие A_4 .

3.8. Нахождение вероятностных распределений по наблюдаемым данным

Любая современная система имитационного моделирования позволяет пользователю применять практически все известные законы распределения как непрерывных, так и дискретных случайных величин. Задача экономиста-исследователя заключается в том, чтобы по данным наблюдения экономических процессов найти наиболее близкий (т.е. соответствующий экспериментальным данным) закон распределения случайной величины.

Процедуру, позволяющую подобрать распределение, которое с достаточной степенью точности описывает наблюдаемые данные, назы-

вают *подгонкой* (английский термин *fitting*). Такая процедура реализована в наиболее известном пакете статистических программ STATISTICA 5.5, а также в других более поздних версиях этого пакета.

Итак, имея значения переменной y , мы проверяем гипотезу, согласно которой распределение y описывается вероятностным законом F , например законом Пуассона для дискретных случайных величин.

Одним из популярных и простых критериев согласия наблюдаемых данных с гипотезой является критерий χ^2 (хи-квадрат) Пирсона.

Изложим идею этого критерия для дискретной случайной величины (с.в.) y , принимающей значения y_1, y_2, \dots, y_k . Предположим, что произведено n независимых опытов, в каждом из которых с.в. y приняла определенное значение. На основе этих данных составлен статистический ряд распределения с.в. y :

y_1	y_2	...	y_i	...	y_k
p_1^*	p_2^*	...	p_i^*	...	p_k^*

где $p_i^* = n_i / n$ – частота события, состоящего в том, что y принимает значение, равное y_i ; n_i – число опытов, в которых появилось это событие, $i = 1, \dots, k$.

Мы выдвигаем гипотезу H , состоящую в том, что с.в. y имеет ряд распределения:

y_1	y_2	...	y_i	...	y_k
p_1	p_2	...	p_i	...	p_k

где p_1, \dots, p_k – значения вероятностей, вычисленные согласно некоторому теоретическому закону. Отклонения экспериментальных частот p_i^* от теоретических p_i мы объясняем случайными причинами.

Критерий Пирсона χ^2 (мера расхождения между экспериментальным и теоретическим распределениями) запишется в виде

$$\chi^2 = n \sum_{i=1}^k (p_i^* - p_i)^2 / p_i. \quad (3.7)$$

Распределение χ^2 зависит от параметра r («числа степеней свободы»), равного k минус единица минус число независимых условий («связей»), наложенных на частоты p_i^* . Обязательно накладывается условие $\sum_{i=1}^k p_i^* = 1$.

Для распределения χ^2 составлены таблицы, которые даются в любом учебнике по теории вероятностей и математической статистике. Пользуясь ими, можно для каждого рассчитанного значения χ^2 и числа степеней свободы r найти вероятность p ошибиться при отклонении гипотезы H .

Пример. Пусть $\chi^2 = 13,44$, $k = 12$. Тогда $r = k - 2$. Если обратиться к таблице, то $p = 0,20$. Это означает, что вероятность ошибиться при отклонении гипотезы H составляет 0,20. Если бы $\chi^2 = 16,92$, то вероятность ошибиться при отклонении гипотезы H составляла бы 0,05. Ясно, что с ростом χ^2 вероятность правдоподобности гипотезы H уменьшается.

Однако пакеты статистических программ (например, STATISTICA) проводят все необходимые вычисления и поправки на число степеней свободы автоматически, избавляя пользователя от необходимости утомительного поиска в таблицах.

Критерий согласия χ^2 можно применять и для непрерывных случайных величин, если, группируя статистический ряд, приближенно заменить непрерывную случайную величину y дискретной с возможными значениями $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_k$, где \tilde{y}_i – середина i -го интервала.

Вопросы для самоконтроля

1. Почему в имитационном моделировании необходимо применять метод статистических испытаний (метод Монте-Карло)?
2. Как вычисляется среднее время задержки запросов в очереди на обслуживание?
3. Как работает алгоритм повременного имитационного моделирования с детерминированным шагом?
4. В чем сущность алгоритма повременного имитационного моделирования по «особым» состояниям?
5. Как работает алгоритм позаявочного имитационного моделирования?

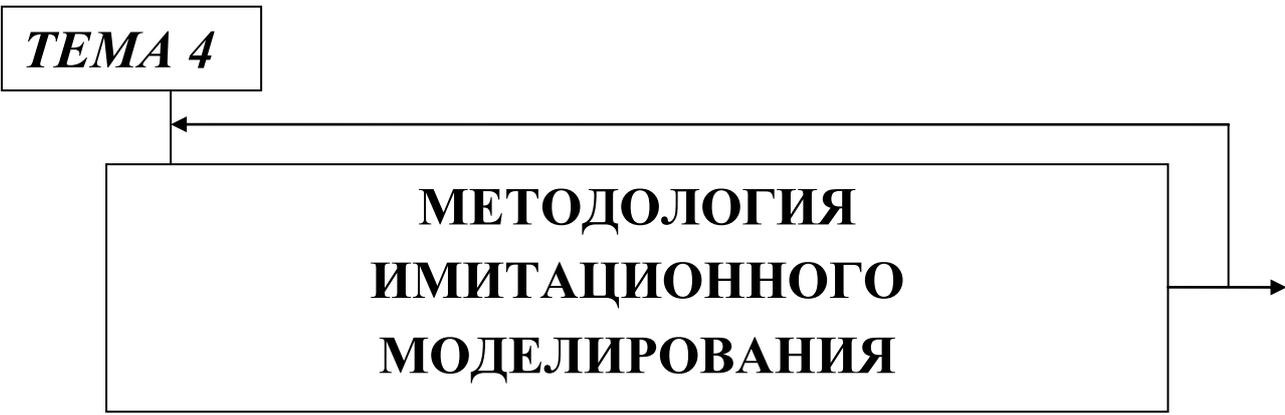
6. Почему числа, генерируемые системами ИМ, называют не случайными, а псевдослучайными?

7. Какие основные типы распределений случайных величин применяются в имитационном моделировании?

8. Как получить последовательность случайных чисел, распределенных на интервале (0;1) случайных чисел?

9. Опишите алгоритм моделирования случайных событий по экспериментальным данным.

10. В чем состоит смысл критерия Пирсона χ^2 ?



**МЕТОДОЛОГИЯ
ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ**

4.1. Имитационные проекты и планирование компьютерного эксперимента

Главная ценность имитационного моделирования состоит в том, что в его основу положена методология системного анализа. Она дает возможность исследовать проектируемую или анализируемую систему по технологии операционного исследования, включая такие взаимосвязанные этапы, как: содержательная постановка задачи; разработка концептуальной модели; разработка и программная реализация имитационной модели; оценка адекватности модели и точности результатов моделирования; планирование экспериментов; принятие решений. Благодаря этому имитационное моделирование можно применять как универсальный подход для принятия решений в условиях неопределенности и для учета в моделях трудно формализуемых факторов.

Имитационное исследование оформляется в виде документированного проекта, пояснительная записка которого состоит из следующих структурных элементов:

- титульный лист;
- реферат;
- содержание;
- постановка задачи;
- раздел «Анализ возможных методов решения поставленной задачи»;
- раздел «Разработка концептуальной модели»;
- раздел «Выбор программных средств моделирования»;

- раздел «Разработка структурной схемы имитационной модели и описание ее функционирования»;
- раздел «Оценка адекватности модели»;
- раздел «Организация экспериментов с моделью»;
- выводы и рекомендации относительно применения модели;
- перечень ссылок;
- приложения.

Реферат. Реферат предназначен для ознакомления с имитационным проектом. Он должен быть кратким и информативным.

Текст реферата передает полное библиографическое содержание проекта, который выполняют в соответствии с требованиями действующего стандарта относительно библиографического и издательского дела.

Реферат содержит:

- сведения об объекте исследования и количестве иллюстраций, таблиц, приложений и использованных источников (в соответствии с перечнем ссылок на них);
- текст реферата;
- перечень ключевых слов.

В тексте реферата необходимо отобразить приведенную в проекте информацию в такой последовательности:

- объект исследования;
- цель работы;
- методы исследования;
- результаты;
- основные конструктивные, технологические характеристики и показатели;
- значимость работы и выводы;
- прогнозы и предложения относительно развития объекта исследования или разработки.

Реферат рекомендуется выполнять в объеме не более 500 слов.

Ключевые слова или словосочетания, которые являются определяющими для раскрытия содержания (если такие необходимы), помещают после текста реферата в именительном падеже в строку через запятую.

Постановка задачи. Приводится содержательная постановка задачи, определяются цели исследования, внешние воздействия и ограничения, которые накладываются на систему.

Анализ возможных методов решения поставленной задачи.

В данном разделе (главе) анализируются методы решения поставленной задачи, указываются их преимущества и недостатки, дается четкое обоснование выбора метода решения, указываются источники, по которым проводится обзор методов решения. Приводятся конкретные причины, по которым задача не может быть решена аналитическими методами.

Разработка концептуальной модели. В данном разделе необходимо:

- определить цели моделирования;
- разработать структурную схему модели; описать входные, выходные переменные и параметры модели;
- представить функциональные зависимости, описывающие поведение переменных и параметров;
- описать ограничения на возможные изменения величин;
- выбрать степень детализации представления модели;
- сформулировать целевые функции (критерии эффективности) моделируемой системы.

Выбор программных средств моделирования. При предварительном выборе программных средств необходимо определить:

- существует ли хорошо написанное руководство или инструкция для пользователя;
- обеспечивается ли хорошая диагностика ошибок;
- знакомо ли средство программирования модели.

При кратком описании выбранного средства необходимо указать:

- имеющиеся средства генерации случайных чисел и переменных;
- возможности отладки программной реализации модели;
- организацию сбора статистических данных о работе модели;
- возможности отображения структуры моделируемой системы;
- возможности редактирования модели;
- наличие средств автоматизации создания программ.

Разработка структурной схемы имитационной модели и описание ее функционирования.

Описание имитационной модели. В данном подразделе разрабатывается алгоритм моделирования, приводится схема имитационной модели в терминах алгоритма моделирования или выбранного средства моделирования и описывается программная реализация модели. Приводится

таблица определений, содержательное значение всех используемых статических и динамических объектов с описанием их свойств.

Описание программной реализации имитационной модели. В данном подразделе дается описание каждого блока модели с комментариями к ним. Для оценки правильности программной реализации имитационной модели проводится пробный эксперимент (прогон модели с тестовыми данными) с целью проверки правильности функционирования программы. Проводятся данные по тестированию модели.

Оценка адекватности модели. В этом разделе выполняется предварительный расчет ожидаемых от модели результатов с помощью операционного анализа сетей массового обслуживания (СМО) или метода средних величин. Полученные результаты сравниваются с результатами пробного прогона модели. Обосновывается правильность построения модели путем обратных преобразований (программная модель преобразуется в алгоритм моделирования или логическую схему, а затем в концептуальную модель и постановку задачи). Осуществляется подбор тестовых данных для проверки функционирования модели во всем диапазоне исходных данных. Приводятся данные трассировки модели.

Организация экспериментов с моделью.

План экспериментов. Основная цель планирования – изучение поведения моделированной системы при наименьших затратах на экспериментирование. Для этого строится план экспериментов. Чаще всего используют такие эксперименты:

- сравнение средних значений и дисперсий разных альтернатив;
- определение важности учета или значимости влияния переменных и ограничений, которые накладываются на эти переменные;
- поиск оптимальных значений переменных на некотором множестве возможных значений.

Разрабатывается *план экспериментов* с моделью для достижения поставленной цели. При необходимости используют отсеивающий или оптимизирующий эксперименты. В случае оптимизации числового критерия формируют *гипотезы* о выборе наилучших вариантов структур моделируемой системы или режимов ее функционирования, определяют *диапазон значений параметров* (режимов функционирования) модели, в границах которых осуществляется поиск оптимального решения.

Оценка точности результатов моделирования. Для оценки точности стохастических моделей строятся доверительные интервалы

для получаемых выходных переменных. Если модель работает в переходном режиме, то используют метод повторений экспериментов и дисперсионный анализ. Для стационарных эргодических и регенерирующих процессов определяют длительности прогонов модели, при которых гарантирована точность полученных оценок.

В конце этого раздела указывают затраты компьютерного времени на моделирование, приводят соображения о возможных улучшениях в работе системы.

Анализ и оценка результатов. Приводятся результаты компьютерных экспериментов в виде графиков, таблиц, распечаток, а также даются качественные и количественные оценки результатов моделирования.

Поиск наилучших решений. За один прогон модели невозможно определить наилучшие показатели системы или выбрать ее структуру. Процедура поиска наилучших решений всегда оказывается итерационной и циклической. Если осуществляется поиск оптимальных значений на поверхности отклика, то используют оптимальное планирование экспериментов и численные методы оптимизации. Для выбора наилучшего решения из нескольких альтернатив обычно используют проверки гипотез и выявляют гипотезу победительницу.

Выводы и рекомендации по использованию модели. По полученным результатам формулируются выводы по проведенным исследованиям и определяются рекомендации по использованию модели. Описываются сценарии **принятия решений**.

Перечень ссылок. В списке литературы необходимо указывать только те источники, на которые есть ссылка в проекте.

Приложения. Приложения содержат тексты программ и другие вспомогательные материалы. Объем приложений не ограничивается.

4.2. Организация компьютерного эксперимента

С точки зрения представления поведения моделируемой системы имитационные модели относятся к классу описательных. Если в модели учитываются случайные факторы, то в процессе имитации обычно осуществляется большое число прогонов модели как с разными входными данными, так и с разными значениями последовательностей случайных чисел. Для детерминированной модели (без учета случайностей) достаточно одного прогона модели для каждой комбинации входных данных, однако в жизни такие модели встречаются крайне редко.

В результате экспериментирования с моделью получают большое количество выходных данных, которые должны быть структурированы и интерпретированы так, чтобы их можно было использовать для принятия решений по результатам моделирования. Для правильной интерпретации полученных от модели выходных данных необходимо организовать эксперименты с моделью.

Организация эксперимента – это разработка плана проведения экспериментов, который дает возможность за минимальное число прогонов модели и при минимальной стоимости работ сделать статистически значимые выводы или найти наилучшее решение. При организации эксперимента обычно определяют:

- входные данные для каждого эксперимента;
- количество прогонов имитационной модели;
- длительность одного прогона модели;
- длительность переходного процесса моделирования, после которого необходимо собирать выходные данные;
- стратегию сбора данных для каждого прогона модели;
- методы оценки точности выходных данных с построением доверительных интервалов;
- чувствительность модели к входным данным, различным видам распределений, сценариям поведения моделируемой системы;
- условия эксперимента и сценарии;
- условия генерации потоков случайных чисел внутри систем и моделирования и для вероятностных входных данных;
- стратегию достижения цели эксперимента (например, сравнение альтернативных вариантов или оптимизация целевой функции).

Конечная цель проведения экспериментов – это получение достаточной статистической информации для принятия решений по результатам моделирования. Как правило, моделирование проводится с целью нахождения некоторых экстремальных значений характеристик моделируемой системы (оптимизирующий эксперимент) или для выявления важных факторов, влияющих на моделируемую систему (отсеивающий эксперимент). Оба эти эксперимента используют факторные планы и аппроксимируют поверхность отклика полиномами разного порядка; для поиска экстремальных значений применяются численные методы оптимизации. Для этих экспериментов необходима некоторая функциональная зависимость значений выходной переменной (отклика) от входных переменных или факторов, которая, как пра-

вило, отражает критерий эффективности моделируемой системы. Таким образом, поиск наилучшего решения выражается численной характеристикой этого критерия, а для нахождения экстремальных значений необходимо исследовать поверхности отклика (проводить эксперименты) в разных точках. От выбора начальной точки в факторном пространстве во многом зависит эффективность экспериментов.

Другой вид экспериментов, проводимых с моделью, – это структурная оптимизация, под которой будем понимать поиск наилучшей структуры моделируемой системы. В этом случае эксперименты проводятся с разными моделями, а не с одной, как в предыдущем случае, причем модели могут отличаться структурой, параметрами и принятыми алгоритмами поведения. Для таких экспериментов нет единого числового критерия оптимизации, что затрудняет использование классических методов. Однако количество рассматриваемых вариантов, как правило, невелико, поэтому для структурной оптимизации можно использовать метод выдвижения гипотез с перебором вариантов. Оптимизация каждого варианта моделируемой системы обычно осуществляется с помощью поиска узких мест и их устранения, т.е. балансировки моделируемой системы. Узкие места определяют пропускную способность всей системы. Поиск наилучшего решения осуществляется путем сравнения рассмотренных вариантов.

4.3. Проблемы организации имитационных экспериментов

Перечислим основные проблемы, возникающие при экспериментировании с имитационными моделями.

Задание начальных условий эксперимента. Обычно эксперимент начинают из состояния «пусто и свободно», т.е. когда в модели нет транзактов. Если рассматривать достаточно длительный период моделирования, то можно наблюдать так называемый период «разогрева» модели или *переходной период*, после которого модель может перейти в стационарный режим работы. Учет данных переходного периода для выходных переменных модели будет вносить смещение в статические оценки. Чтобы уменьшить влияние данных переходного процесса на конечные результаты, можно поступить следующим образом:

- запустить модель с модальных (наиболее вероятных) значений установившегося режима;
- запустить модель со средних значений установившегося режима.

Эти способы обычно обеспечивают уменьшение длительности переходного процесса модели. При этом они дают эффект только в том случае, если загрузка обслуживающих устройств в модели не велика. При стремлении коэффициентов загрузки устройств к единице на выходе модели может наблюдаться стационарный процесс, в котором нельзя четко выделить данные переходного режима.

При оценивании выходной величины рекомендуется не учитывать данные переходного процесса, так как они могут давать существенное смещение искомых оценок. Это достигается путем удаления данных переходного процесса (с помощью команды **RESET**). Лучший способ определения установившегося процесса – это использование графиков для наблюдения за изменением переходного процесса во времени.

Команда **RESET** оставляет транзакты в модели, обнуляет статистику и освобождает устройство обслуживания. Стандартный числовой атрибут **C1** дает значение модельного времени с момента выдачи последней команды **RESET**, а **СЧА AC1** – абсолютное модельное время с начала моделирования.

Правило останова определяет длительность имитационного прогона. От продолжительности прогона зависит точность результатов моделирования.

Состояние модели в момент прекращения прогона. Часто при моделировании возникает процесс: «Что делать с оставшимися компонентами (транзактами) модели окончания ее работы?». Учет оставшихся компонентов может привести к смещению оценки в большую сторону. Например, при моделировании работы некоторого цеха использовалось правило: что наиболее короткие работы запускаются раньше. На момент окончания в модели останутся незавершенные работы с длительными временами выполнения. Если их не учитывать, то оценка средней длительности работ в цеху будет занижена.

Определение длительности прогона модели при наличии в модели процессов с различными скоростями протекания. Оценку точности результатов моделирования обычно выполняют для самого медленного процесса в модели. В этом случае оценки для быстрых процессов будут заведомо намного лучше, чем для медленного процесса, т.е. доверительные интервалы для них будут меньше. При разработке имитационной модели обычно выбирают степень детализации модели так, чтобы скорости протекающих в ней процессов не различались более чем на два порядка. В случаях моделирования редких событий

(медленные процессы), например отказов оборудования, необходимо укрупнять состояния для быстрых процессов. Для этого обычно используют аналитико-имитационные модели.

4.4. Имитация работы объекта экономики в трех измерениях. Имитация основных процессов

Имитация работы объекта экономики обычно производится в трех измерениях: материальные, денежные и информационные потоки. Поясним эту идею на примере фабрики по производству велосипедов.

На фабрику приходит заказ на производство велосипеда (партии велосипедов). Такой заказ можно представить в виде транзакта. Далее для имитации работы фабрики необходимо «запустить» имитацию трех прогонов: материальных, денежных и информационных. С этой целью исходный транзакт («транзакт-родитель») порождает несколько транзактов-потомков. Один транзакт-потомок инициирует изготовление паспорта изделия, инструкции по эксплуатации, гарантийного обязательства (можно отнести эти процессы к информационным). Другой транзакт-потомок инициирует оформление и контроль платежных документов, начисление НДС (это финансовый поток). Несколько транзактов-потомков инициируют процесс материального производства: поступление со склада заготовок и комплектующих изделий, изготовление узлов и деталей велосипеда.

В конце программы происходит синхронизация и объединение транзактов-потомков: производится сборка велосипеда, вручение покупателю платежных документов, документации на велосипед.

Фактически любая имитационная модель решает задачу моделирования системы массового обслуживания (СМО). Основные признаки, которые позволяют определить некоторую систему как СМО, следующие:

- входящий поток требований или заявок, которые поступают на обслуживание;
- дисциплина постановки в очередь и выбор из нее;
- правило, по которому осуществляется обслуживание;
- выходящий поток требований;
- режимы работы.

Входящий поток задается моментами времени поступления требований (заявок). Закон поступления может быть детерминированным

либо вероятностным. В последнем случае входящий поток требований необходимо описывать распределением вероятностей интервалов времени между соседними требованиями.

Дисциплины постановки в очередь и выбора из нее задают порядок постановки требований в очередь в случаях, когда заняты устройства обслуживания и порядок выхода из очереди, по которому поддерживается дисциплина требований (тразактов) в очереди:

- «раньше поступил – раньше обслужился» (FIFO);
- «последним поступил – первым обслужился» (LIFO).

Во многих производственно-технических системах очередь имеет ограниченное количество мест обслуживания – так называемый *буфер*.

Правила обслуживания задают длительность обслуживания (также детерминированная либо случайная величина), количество требований, которые могут обслуживаться одновременно, дисциплину обслуживания.

Обслуживание осуществляется *устройствами*. Различают *одноканальные* и *многоканальные* устройства. Под многоканальными устройствами понимают объединение параллельно работающих идентичных устройств.

Дисциплины обслуживания определяют условия прекращения обслуживания требований, правила выбора для обслуживания следующего требования. В последнем случае выбор для обслуживания очередного требования производится на основе заданных приоритетов.

Выходящий поток – это поток требований (как обслуженных, так и необслуженных), которые покидают систему. В большинстве случаев выходящий поток является входящим для следующего этапа моделирования. Если же мы имеем дело с последним этапом, то обработка параметров выходящего потока позволяет провести расчеты выходных характеристик системы.

В системах дискретно-событийного имитационного моделирования можно задавать самые различные ***режимы работы*** СМО. Могут быть предусмотрены режимы отказа устройства обслуживания (имитация выхода из строя устройств обслуживания), *блокирования обслуживания, замедление обслуживания*.

Определим основные понятия и термины, используемые в дискретно-событийном имитационном моделировании.

Система – множество объектов (например, станков, операторов, конвейеров), которые взаимодействуют одновременно для достижения одной или большего количества целей.

Модель – абстрактное представление системы, обычно содержит структурные, логические или математические отношения, которые описывают систему в терминалах состояний, объектов и их свойств, множеств, процессов, событий, действий и задержек.

Состояние системы – множество переменных, которые содержат всю информацию, необходимую для описания свойств системы в любое время.

Объект – любой элемент или компонент в системе, который должен быть представлен в модели в явном виде (например, обслуживающее устройство, клиент, машина).

Свойство или *атрибут* – свойства данного объекта (например, приоритет ожидающего клиента).

Список – множество связанных объектов, упорядоченное некоторым логическим способом (например, все клиенты, находящиеся в настоящее время в очереди ожидания, упорядочены по принципу «раньше прибыл – раньше обслужился» или по приоритету).

Событие – мгновенно возникающее изменение состояния системы (например, станок окончил обработку детали, или на освободившийся станок поступила для обработки деталь из очереди, или станок сломался, обработка детали прервана).

Уведомление о событии – запись события, которое произойдет в потоке событий или в некотором будущем времени наряду с любыми связанными данными, необходимыми для обработки события (запись всегда включает тип события и время события). Может применяться для прогнозирования процессов в системе.

Список событий – список намеченных будущих событий, упорядоченных по времени возникновения, известный также как список будущих событий (СБС).

Действие – продолжительность времени указанного промежутка (например, время обслуживания или время между поступлениями заявок), для которого известно, когда оно начинается и заканчивается (хотя может быть определено в терминалах статистического распределения).

Задержка – продолжительность времени неопределенного промежутка (например, задержка клиента в очереди по правилу «последний пришел, первый обслужился», так как начало обслуживания зависит от будущих поступлений).

Модельное время – неотрицательная возрастающая величина, отражающая течение времени в имитационной модели.

Часы – переменная, отражающая протекание времени моделирования, называется в примерах.

Транзакт – некоторое требование на обслуживание, которое может перемещаться по модели, вставать в очереди к устройствам.

Генератор – объект модели, вносящий транзакты в модель по некоторому определенному автором закону – детерминированному или статистическому.

Уничтожение – удаление требования из системы. Обычно применяется после полного обслуживания требования во всех необходимых устройствах либо после того, как по каким-то причинам установлена невозможность обслуживания.

Семейство транзактов – группа транзактов, полученных из одного транзакта операцией расщепления.

Вопросы для самоконтроля

1. Как оформляется имитационное исследование в виде документированного проекта?
2. В чем состоит разработка концептуальной модели?
3. По каким критериям выбираются программные средства для имитационного моделирования?
4. Как проводится оценка точности результатов моделирования?
5. Как учесть переходный период, предшествующий переходу модели в стационарный режим работы?
6. Имитация работы объекта экономики обычно производится в трех измерениях: материальные, денежные и информационные потоки. Поясните эту идею на примере.
7. Поясните на конкретном примере имитацию работы объекта экономики в трех измерениях: материальные, денежные и информационные потоки.
8. Какие дисциплины постановки в очередь и выбора из нее вам известны?

**ПАКЕТЫ ПРОГРАММ
ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ
ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ**

**5.1. Сравнительный анализ
популярных пакетов программ
имитационного моделирования:
GPSS World, AnyLogic, Pilgrim**

Наиболее распространены следующие пакеты программ:

- GPSS World компании Minuteman Software (США);
- AnyLogic Санкт-Петербургского института информатики РАН;
- Pilgrim разработки российских (из МЭСИ) и голландских ученых.

GPSS World развивается уже почти 50 лет. С 2000 г. появилась версия этой системы под ОС Windows.

В мире издано огромное количество литературы по GPSS World. Решено много типовых задач и в области экономики. Поэтому можно уверенно утверждать, что это самая популярная система имитационного моделирования. Важно, что имеется бесплатная версия GPSS World, которую можно использовать в учебном процессе вузов.

Система имитационного моделирования AnyLogic более современна и имеет значительно большие возможности. Однако стоимость ее существенна даже в случае применения в образовательных целях. К тому же авторы уделяют мало внимания изданию учебно-методической литературы.

Такие же недостатки присущи и системе имитационного моделирования Pilgrim.

В связи с этим учебный процесс в СГСЭУ ориентирован на применение GPSS World.

Дискретно-событийный метод имитационного моделирования в настоящее время реализуется некоторыми авторами с помощью различных универсальных объектно-ориентированных инструментальных средств.

Здесь нелишне вспомнить, что на основе объектно-ориентированного подхода в 90-е гг. прошлого века построены практически все алгоритмические языки высокого уровня, применяемые в настоящее время. «Старые» процедурно-ориентированные языки (Basic – в Visual Basic, Pascal – в Delphi, C – в Visual C++).

Специализированные системы программирования, предназначенные специально для имитационного моделирования, о которых говорилось выше, также относятся к объектно-ориентированным.

В основе объектно-ориентированных языков лежит понятие объекта, имеющего присущие ему свойства и методы, которые эти свойства изменяют.

Свойства (или данные) определенного экземпляра объекта можно *изменять* только его собственными методами. Другие объекты могут только просматривать данные этого экземпляра. Такая возможность называется *инкапсуляцией*.

Другим важным свойством объектно-ориентированного подхода является *наследование*. Оно означает, что если новый тип объекта (именуемого потомком) определяется на основе уже существующего типа объекта (предка), то объект-потомок наследует все характеристики объекта-предка. Однако некоторые характеристики потомка можно изменять или добавлять к ним новые.

Полиморфизм подразумевает, что различные типы объектов, объединенные общим происхождением, могут иметь методы с одинаковыми названиями, но при активизации поведение разных объектов будет различаться.

Для объектно-ориентированного подхода выделяют еще одно характерное свойство – *иерархию*. Известные ученые А.М. Лоу и В.Д. Кельтон иллюстрируют это свойство таким примером: в производственной системе цеха могут рассматриваться как объекты (первый уровень иерархии); эти цеха содержат такие объекты, как рабочие станки и автопогрузчики (второй уровень иерархии); данные автопогрузчика включают характеристики его скорости и грузоподъемности; методом автопогрузчика может быть порядок, используемый им для выбора следующей работы.

Многими авторами отмечаются такие преимущества объектно-ориентированного подхода:

1) обеспечивает возможность повторного использования объектов и удобный способ их изменения;

2) помогает «справиться» со сложными системами благодаря делению их на различные объекты;

3) упрощает изменение моделей, поскольку изменения в объекте-предке отображаются в объектах-потомках.

Многие производители утверждают, что их программное обеспечение имитационного моделирования является объектно-ориентированным, но иногда такие программные средства не обеспечивают наследование, полиморфизм или инкапсуляцию. Однако далеко не все пользователи систем (инструментальных средств) имитационного моделирования обладают опытом и квалификацией, позволяющими в полной мере воспользоваться этими свойствами объектно-ориентированного подхода. Поэтому эти недостатки систем имитационного моделирования зачастую «проходят» мимо внимания пользователей средней квалификации.

На основе метода имитационного моделирования построены программы для обучения экономистов, например БИЗНЕС-КОРПОРАЦИЯ для предприятий реального сектора экономики. Пользователь может менять различные параметры внешней и внутренней среды. Однако такая программа, очевидно, не вполне подходит для специальности информационно-компьютерного профиля, поскольку не дает возможности вмешиваться в программу, программировать новые модули, придающие системе новые функциональные свойства.

5.2. Характеристика системы имитационного моделирования GPSS World

GPSS World (GPSSW – General Purpose System Simulation World – Мировая общецелевая система моделирования) – это система моделирования, предназначенная для создания имитационных моделей в самых различных областях науки и техники. Многими исследователями система GPSS World успешно применяется для решения экономических и управленческих задач.

В основу системы GPSS World положен язык имитационного моделирования GPSS.

Достоинство языка GPSS заключается в следующем:

- он прост в изучении и использовании;
- наиболее важные классы объектов (транзакция (транзакты), каналы, накопители, логические переключатели и др.) и их свойства широко используются в реальных вычислительных сетях, производственных и коммерческих системах;
- диапазон использования языка достаточно широк;
- язык постоянно совершенствуется;
- расширение создаваемых моделей легко осуществимо;
- доступно широкое использование анимации;
- пользователи способны легко понять внутреннюю логику и алгоритм GPSS;
- интерфейс прост и удобен;
- при построении моделей языка позволяет оперировать непосредственно понятиями имитируемой системы.

Первая система GPSS разработана в 1984 г. и ориентирована на операционную систему DOS. Современная версия (которая и получила название GPSS World) системы имитационного моделирования, реализованная под операционной системой Windows компании Microsoft, появилась в 2000 году.

В GPSS World введены дополнительные возможности, назовем основные из них:

- 1) по всем классам объектов и переменных реализованы динамические графические окна, в которые предоставляется в реальном времени промежуточная выходная статистика;
- 2) гибкий процедурный язык PLUS может быть использован для построения моделей и в процедурах проведения эксперимента;
- 3) введены средства поддержки факторного анализа, традиционного дисперсионного (ANOVA) и регрессионного анализа, оптимизация на основе методологии оптимального планирования эксперимента;
- 4) стали доступны элементы непрерывного моделирования;
- 5) возможно автоматизированное конструирование моделей бизнес-процессов.

С помощью этой системы можно эффективно моделировать как производственные, так и непроизводственные процессы: функционирование торговых увеселительных заведений, портов, уличных движений, проведение военных действий, работу редакций, учреждений и сети Интернет, различных систем массового обслуживания и т.д.

Система имеет большой выбор команд для управления процессом моделирования, которые можно использовать как в интерактивном режиме, а также включать в модель. В системе GPSSW реализована процедура визуализации процесса функционирования модели с применением методов мультипликации.

Система GPSSW имеет новый высокоскоростной транслятор, работающий в сотни раз быстрее его предшественников.

В настоящем лабораторном практикуме рассматривается так называемая «студенческая версия» в системе GPSS World. Ее можно бесплатно скачать с сайта фирмы Minuteman Software по адресу: www.minutemansoftware/download. Эта версия так же эффективна, как и коммерческая. Единственное ограничение связано с числом операторов программы, которое не должно превышать 150.

В процесс инсталляции системы GPSSW по умолчанию устанавливается в каталог C:\Program Files\Minuteman Software\GPSS World Student Version.

Для системы GPSSW требуется IBM – совместимый компьютер с ОС Windows 95, 98 и выше. Целесообразно использовать процессоры типа Pentium II и выше. Требуется по меньшей мере 32 Мбайт оперативной памяти и 10 Мбайт свободного пространства на жестком диске (винчестере).

5.3. Главное меню GPSS World

Запуск системы GPSSW можно выполнить несколькими способами, например так:

1) щелкните мышью по кнопке **Пуск** ОС Windows. Появится всплывающее меню;

2) щелкните по пункту **Программы** во всплывающем меню. Появится следующее всплывающее меню;

3) щелкните по пункту **GPSS World Student Version**. Откроется главное окно системы GPSSW (рис. 5.1).

В первой строке (строке заголовка) главного окна указано название окна – GPSS World. Во второй строке располагаются пункты главного меню, в третьей – стандартная панель инструментов. Нижняя строка главного окна – строка состояния системы, в которой дается краткое описание выделенной команды.

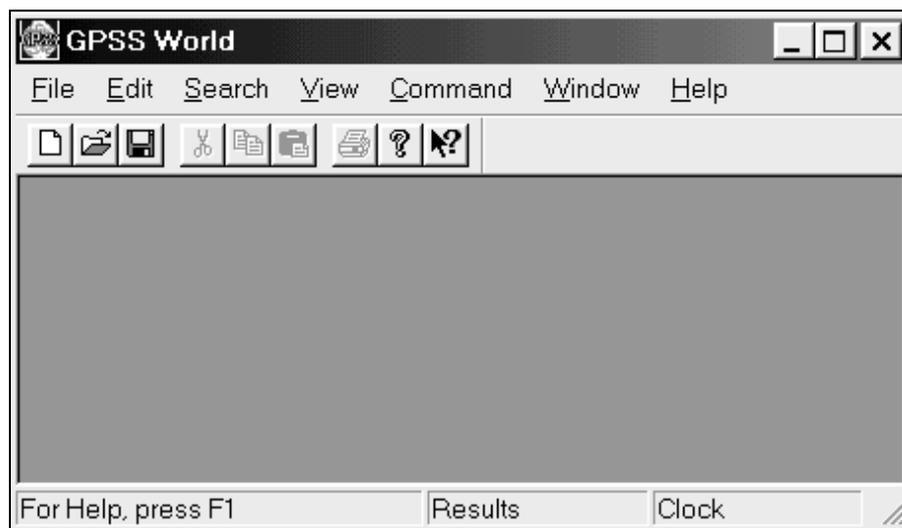


Рис. 5.1. Главное меню системы GPSS World

Система GPSSW имеет иерархическую систему меню, состоящую из главного меню, систем выпадающих и всплывающих меню (подменю).

Пункт **File** главного меню служит для работы с файлами документов (рис. 5.2).

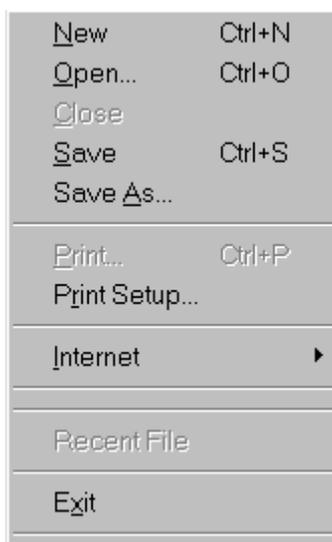


Рис. 5.2. Выпадающее меню пункта **File** главного меню

Файлы имитационных моделей в системе GPSSW записываются в окне **Model** (Модель) и сохраняются с расширением «.gps», которое указывается сразу после имени файла.

Текстовые файлы системы GPSSW записываются в окне **Text File** (Текстовый файл) и сохраняются с расширением «.txt». Они имеют текстовый формат, их можно прочитать и отредактировать с помощью любого текстового редактора.

Результаты моделирования можно сохранить в окне **REPORT** (Отчет). Такие файлы имеют расширение «**.gpr**».

Меню пункта **File** имеет типичную для Windows-приложений структуру.

Основные принципы построения имитационной модели и операторы GPSS World рассмотрим на примере простейшей задачи о моделировании работы магазина.

5.4. Простейший пример применения GPSS World: моделирование работы магазина

Постановка задачи. Предположим, что нам необходимо промоделировать работу небольшого магазина, который имеет один кассовый аппарат и одного продавца. Нам известны следующие параметры функционирования магазина:

1) поток покупателей (в терминах имитационного моделирования – требований), приходящих в магазин, равномерный; это означает, что функция плотности распределения соответствующей случайной величины имеет прямоугольную форму (рис. 3.2);

2) интервал времени прибытия покупателей колеблется в пределах от 8,7 до 10,3 мин включительно, или $9,5 \pm 0,8$ мин;

3) время пребывания покупателей у кассового аппарата составляет $2,3 \pm 0,7$ мин; после этого покупатели подходят к продавцу для получения товара;

4) время, потраченное на обслуживание покупателей продавцом, составляет $10 \pm 1,4$ минут.

Требуется определить параметры функционирования магазина:

- коэффициент загрузки кассира;
- коэффициент загрузки продавца;
- максимальное, среднее и текущее число покупателей в каждой очереди;
- среднее время обслуживания в каждом канале обслуживания;
- среднее время нахождения покупателя в каждой очереди.

Выявление основных особенностей модели. Для моделирования работы магазина необходимо сформировать входной поток покупателей (требований) и временной интервал моделирования работы магазина. Но перед этим следует выбрать единицу измерения времени.

Для моделирования работы магазина можно взять в качестве единицы измерения минуту.

Создание имитационной модели процесса. Загрузите систему GPSS World (**Пуск, Программы, GPSS World Student Version**). Откроется окно, показанное на рис. 5.1. Для подготовки к вводу текста программы выполните следующие действия:

- 1) щелкните по пункту меню **File** главного меню системы;
- 2) щелкните по пункту **New** выпадающего меню. Появится диалоговое окно для создания новой программы с заголовком **New**;
- 3) выделите пункт **Model** и щелкните по кнопке **OK**. Появится окно модели, в котором будем вводить программу. На рис. 5.3 показаны две первые строки программы.

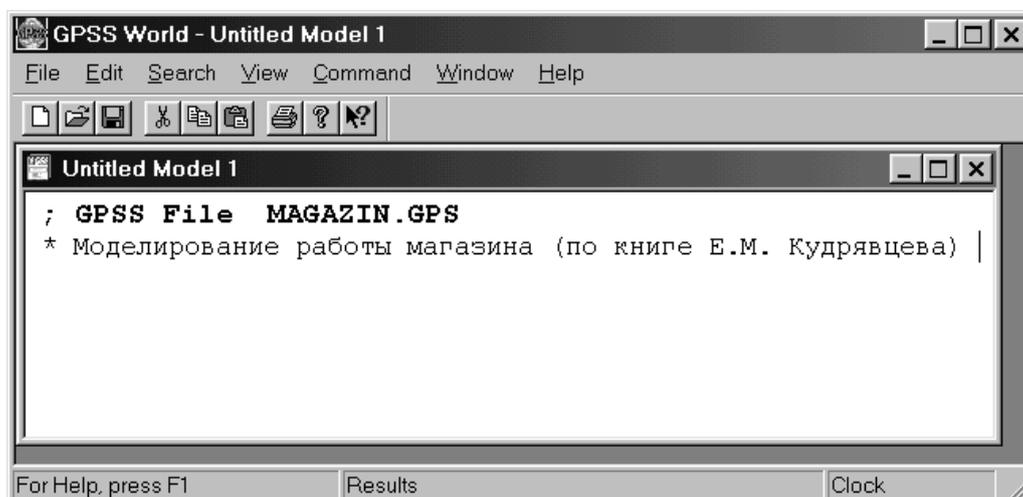


Рис. 5.3. Окно имитационной модели

Необходимо сохранить программу в личной папке в виде файла с расширением «.gps». Для этого подать команду **File, Save As**. Далее следует дать имя файлу (например, **Magazin.gps**) и указать к нему путь, т.е. выбрать диск и папку.

Создание имитационной модели начнем с построения заголовка модели, который может быть представлен, например, в таком виде:

```
; GPSSW File MAGAZIN.GPS  
* Моделирование работы магазина (по книге Е.М. Кудрявцева)
```

Если строка программы начинается со знаков "*" или ";", то она воспринимается не как команда, а как комментарий.

Как видно из первой строки, файл имитационной модели будем сохранять под именем **MAGAZIN.GPS**.

Затем с меткой **t_prod** вставляем команду **QTABLE** создания гистограммы очереди к продавцу:

```
t_prod QTABLE Ocher_prod,0,2,32
```

Подробно этот оператор рассмотрим в последующих разделах.

Моделирование потока покупателей будем выполнять с помощью оператора **GENERATE** (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE 9.5,0.8
```

В поле первого операнда указывается средний интервал времени между прибытием в магазин двух идущих один за другим покупателей (требований, транзактов). В нашем примере он составляет 9,5 минуты.

В поле второго операнда дано отклонение времени прихода покупателей от среднего. В нашем примере это отклонение составляет 0,8 минуты.

Покупатель, пришедший в магазин, сначала встает в очередь к кассиру, если она есть. Это можно промоделировать оператором **QUEUE** (Очередь), который только в совокупности с соответствующим оператором **DEPART** (Выйти) собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE Ocher_kassa
```

В поле операнда дается символьное или числовое имя очереди. Таких очередей в сложных системах может быть очень много. В нашей задаче дадим очереди имя **Ocher_kassa** (Очередь в кассу).

Покупатель может выйти из очереди только тогда, когда освободится кассир (канал обслуживания). Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это можно представить так:

```
SEIZE KASSIR
```

В поле операнда дается символьное или числовое имя канала обслуживания. Таких каналов обслуживания в системе может быть очень много. В нашей задаче каналу дано имя **Kassir** (Кассир). Здесь также имя должно отражать суть описываемого элемента системы.

Выход покупателя из очереди в кассу фиксируется оператором **DEPART** с соответствующим названием очереди. В нашем примере это будет выглядеть так:

DEPART Ocher_KASSA

Далее должно быть промоделировано время пребывания покупателя, непосредственно обслуживаемого кассиром. Это время в нашем примере составляет $2,3 \pm 0,7$ минуты. Для моделирования этого процесса используется оператор **ADVANCE** (Задержать), который в нашей задаче будет выглядеть так:

ADVANCE 2.3,0.7

После обслуживания кассиром покупатель отправляется к продавцу за получением оплаченного товара. Однако перед этим системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора **RELEASE**, который в нашей задаче записывается так:

RELEASE Kassir

Следует подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После обслуживания в кассе покупатель направляется к продавцу – следующему каналу обслуживания. Процесс моделирования этой цепи аналогичен только что описанному. И в нашем примере он может быть представлен, например, в таком виде:

QUEUE Ocher_prod
SEIZE Prodavec
DEPART Ocher_prod
ADVANCE 10,1.4
RELEASE Prodavec

После обслуживания продавцом (каналом обслуживания) покупатель (требование) покидает систему. Это действие может быть представлено оператором **TERMINATE** (Завершить):

TERMINATE 1

В поле операнда стоит число 1. Это означает, что систему обслуживания «магазин» покупатель покидают по одному. Завершающим оператором в нашей задаче является управляющая команда **START** (Начать), позволяющая начать моделирование:

START 100

В поле операнда стоит число 100, показывающее, с каким числом покупателей будет моделироваться система работы магазина.

Представление имитационной модели. Для представления имитационной модели (если раньше эти действия не были проделаны) выполните следующие действия:

1) щелкните по пункту **File** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **New** выпадающего меню. Появится диалоговое окно **Новый документ**;

3) выделите пункт **Model** и щелкните по кнопке **ОК**. Появится окно модели, в которое введите данную программу. Это будет выглядеть так, как показано в табл. 5.1 (в левой колонке помещены метки операторов, в центральной колонке – операторы, в правой колонке помещены комментарии к командам).

Во введенной программе в самом начале добавлен оператор **QTABLE** с меткой **t_prod** для сбора информации и построения соответствующей гистограммы функционирования очереди под именем **Ocher_prod** (табл. 5.1).

Текст программы следует сохранить в личной библиотеке на сервере командой **File, Save As**. В появившемся диалоговом окне следует указать путь к сохраняемому файлу (раздел на жестком диске, папку), а также имя файла. Расширение «**.gps**» присваивается файлу автоматически. Присвоим файлу имя **Magazin.gps**.

Если текст программы был ранее сохранен, то его можно загрузить командой **File, Open**, далее указать путь к файлу.

Окно имитационной модели «Магазин» (текст программы)

GPSSW File		MAGAZIN.GPS
* Моделирование работы магазина		
t_prod	QTABLE Ocher_prod,0,2,32	; Определить таблицу для вывода ;результатов по очереди
	GENERATE 9.5, 0.8	; Генерировать нового покупателя
	QUEUE OCHER_KASSA	; Увеличить длину очереди к кассиру
	SEIZE KASSIR	; Занять канал обслуживания (кассу)
	DEPART OCHER_KASSA	; Уменьшить содержимое очереди – ;покупатель начал обслуживаться ;кассиром
	ADVANCE 2.3,0.7	; Покупатель обслуживается в кассе
	RELEASE Kassir	; Кассир закончил обслуживание ;покупателя
	QUEUE Ocher_prod	; Увеличить длину очереди к продавцу
	SEIZE Prodavec	; Занять канал обслуживания (прилавок ;продавца)
	DEPART Ocher_prod	; Уменьшить содержимое очереди – ;покупатель начал обслуживаться ;продавцом
	ADVANCE 10,1.4	; Покупатель обслуживается продавцом
	RELEASE PRODAVEC	; Продавец закончил обслуживание ;покупателя
	TERMINATE 1	; Покупатель покидает магазин
	* START 100	; Запуск программы на выполнение для ;случая 100 покупателей, команда ;временно заблокирована символом "*"

Подготовка к моделированию системы. Перед началом моделирования можно установить вывод тех параметров моделирования, которые нужны пользователю. Для этого:

1) щелкните по пункту **Edit** (Правка) главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **Settings** (Установки) выпадающего меню, закладка **Reports** (Отчеты). Появится диалоговое окно **SETTINGS** для данной модели, в нем можно установить нужные выходные данные, которые отмечаются флажком (галочкой). Для нашего примера это может выглядеть так, как представлено на рис. 5.4.

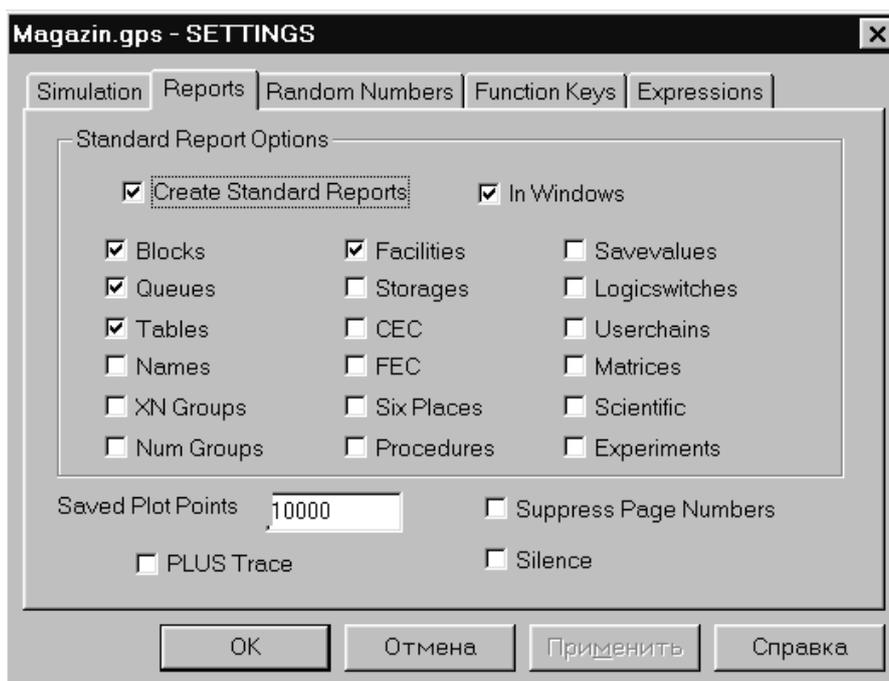


Рис. 5.4. Окно **SETTINGS** с установками для имитационной модели магазина (Выбрана закладка **Reports** для установки параметров отчета.)

Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация о следующих объектах:

- **Blocks** (Блоки);
- **QUEUES** (Очереди);
- **Tables** (Таблицы/гистограммы);
- **Facilities** (Каналы обслуживания).

Моделирование системы. После создания имитационную (выполняемую) модель необходимо оттранслировать и запустить на выполнение. Для этого:

1) щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **Create Simulation** (Создать выполняемую модель) выпадающего меню.

Если управляющая команда **START** есть в модели, то исходная имитационная модель после трансляции, если в ней нет ошибок, начнет выполняться. Будет выполняться то число прогонов, которое указано в поле операнда команды **START**. Затем появится окно **JOURNAL** (рис. 5.5).

Если управляющей команды **START** в модели нет, то исходная имитационная модель будет транслироваться, и если в ней нет оши-

бок, то будет получена оттранслированная модель, т.е. представленная в машинных кодах, готовая к выполнению моделирования, так называемая выполняемая модель (окно рис. 5.5 появится и в этом случае). Фактически этой команды в нашей программе нет – она заблокирована символом *.



Рис. 5.5. Окно **JOURNAL** с сообщением о результатах трансляции модели

Если же в окне рис. 5.5 появятся сообщения об ошибках в программе, то их необходимо устранить, после чего снова оттранслировать.

Перед началом моделирования, а точнее после появления окна **JOURNAL**, можно настроить графики вывода некоторых параметров функционирования системы. Для этого:

1) щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;

3) щелкните по пункту **Plot Window** (Окно графика) всплывающего меню. Появится диалоговое окно **Edit Plot Window** (Окно редактирования графика), которое необходимо соответствующим образом заполнить (рис. 5.6).

Правила образования выражений, используемых в диалоговом окне (рис. 5.6), рассмотрим позже.

Значение параметра **Time Range** (отрезок времени – часть всего интервала времени, который отображается на графике) принято считать равным 100 минут.

Параметры **Min Value** и **Max Value** определяют минимальное и максимальное значения диапазона по оси ординат графика, т.е. диапазона значений, в котором изменяется длина очереди. Очевидно, что **Min Value** необходимо принимать равным нулю. Что же касается величины **Max Value**, то ее надо подбирать исходя из требования наглядного графического представления.

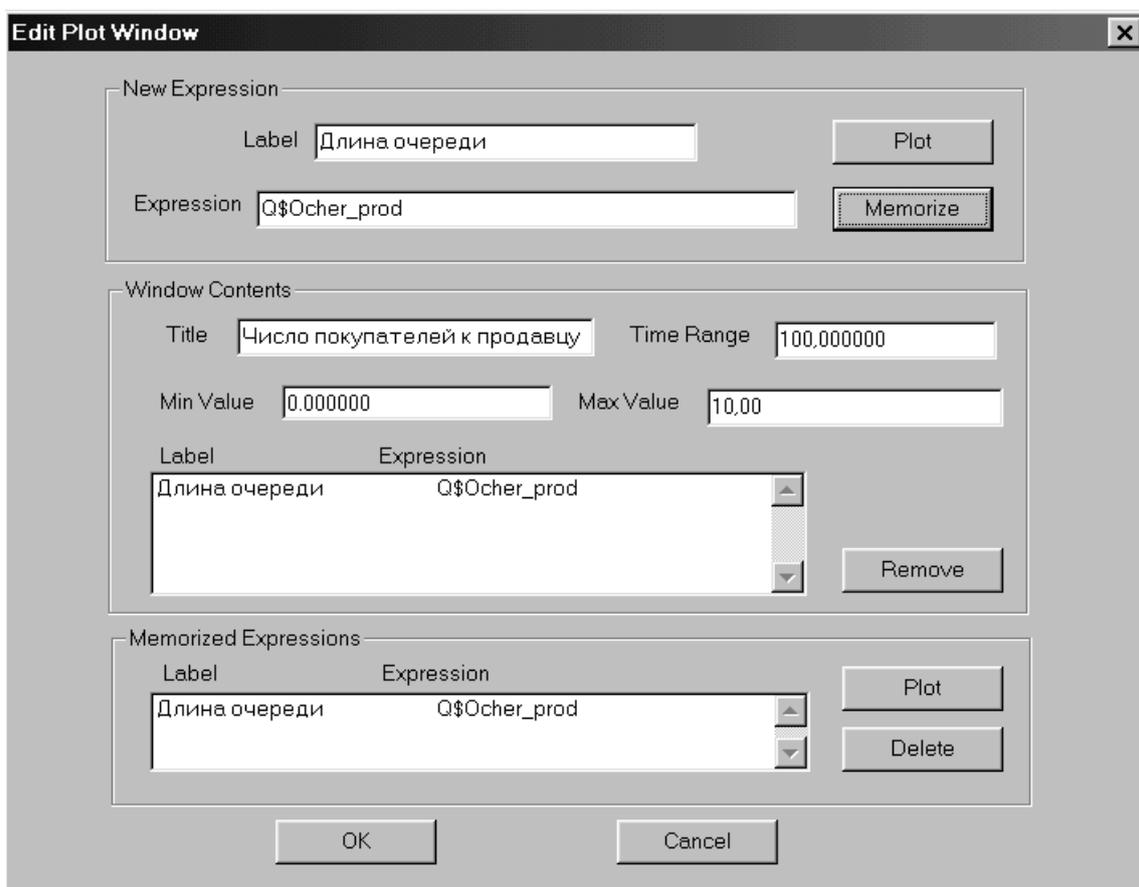


Рис. 5.6. Диалоговое окно **Edit Plot Window** для имитационной модели магазина

Графическое представление результатов моделирования. Допустим, мы хотим на всем периоде моделирования видеть график того, как меняется длина очереди к продавцу. Для нашей задачи окно **Edit Plot Window** может быть заполнено так, как показано на рис. 5.6. В этом окне в качестве разделителя целой и дробной части чисел нужно использовать тот символ, который был введен при установке операционной системы **Windows**, например запятую (но в тексте программы разделителем является точка).

После заполнения диалогового окна **Edit Plot Window** щелкните по кнопкам **Plot** (График), **Memorize** (Запомнить), а затем – по кнопке **OK**. Появится «заготовка графика», т.е. незаполненное окно графика.

После этого:

- 1) щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;
- 2) щелкните по пункту **START**. Появится диалоговое окно **Start Command**;

3) введите в диалоговом окне **Start Command** число посетителей магазина, например 100, и щелкните по кнопке ОК. Появится окно **REPORT** с результатами моделирования. На заднем плане будет размещаться график;

4) щелкните по графику, расположенному на заднем плане, – он выйдет на первый план (фрагмент графика представлен на рис. 5.7);

5) используя горизонтальную и вертикальную полосы прокрутки, вы можете просмотреть построенный график.

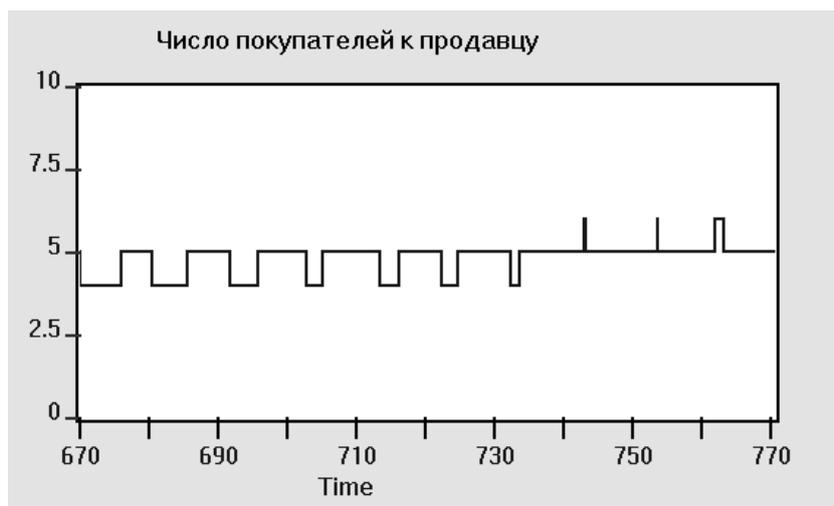


Рис. 5.7. Фрагмент графика зависимости длины очереди к продавцу от времени для имитационной модели магазина

Убедитесь, что длина очереди к продавцу увеличивается с ростом времени моделирования. Очередь возникает при времени моделирования $T_{\text{мод}}$ порядка 100 (в очереди – один покупатель), при $T_{\text{мод}}$ порядка 700 очередь возрастает до 5 покупателей, при $T_{\text{мод}}$ порядка 1000 – до 7.

Посмотрим, образуется ли очередь к кассиру. С этой целью в окне **Edit Plot Window** (рис. 5.6) в поле **Expression** введем имя **Q\$Ocher_kassa**. Результат моделирования, показанный на рис. 5.8, говорит о том, что кассир полностью справляется с обслуживанием потока покупателей.

При выводе графика на передний план окно **REPORT** с результатами моделирования переместится на задний план. Для просмотра окна **REPORT** щелкните по нему мышью. Оно вновь перейдет на передний план и будет выглядеть так, как показано на рис. 5.9.

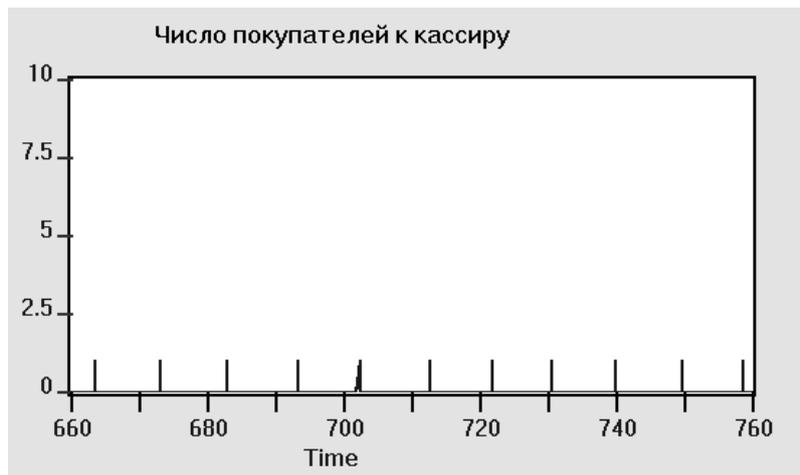


Рис. 5.8. Фрагмент графика зависимости длины очереди к кассиру от времени для имитационной модели магазина

Magazin.8.1 - REPORT

GPSS World Simulation Report - Magazin.8.1
Sunday, December 26, 2004 22:17:33

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES					
0.000	1016.324	12	2	0					
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY				
1		GENERATE	107	0	0				
2		QUEUE	107	0	0				
3		SEIZE	107	0	0				
4		DEPART	107	0	0				
5		ADVANCE	107	0	0				
6		RELEASE	107	0	0				
7		QUEUE	107	6	0				
8		SEIZE	101	1	0				
9		DEPART	100	0	0				
10		ADVANCE	100	0	0				
11		RELEASE	100	0	0				
12		TERMINATE	100	0	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASSIR	107	0.244	2.318	1	0	0	0	0	0
PRODAVEC	101	0.987	9.929	1	101	0	0	0	6
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY	
OCHER_KASSA	1	0	107	107	0.000	0.000	0.000	0	
OCHER_PROD	7	7	107	2	3.607	34.260	34.913	0	
TABLE	MEAN	STD.DEV.	RANGE	RETRY FREQUENCY	CUM.%				

Рис. 5.9. Окно **REPORT** с фрагментом результатов моделирования для имитационной модели магазина

В верхней строке окна **REPORT (Отчет)** указываются:

- **START TIME** (Начальное время) – 0.000;
- **END TIME** (Время окончания) – 1016.324;
- **BLOCKS** (Число блоков в модели, т.е. число операторов) – 12;
- **FACILITIES** (Число каналов обслуживания) – 2;
- **STORAGES** (Число накопителей) – 0.

Ниже перечисляются блоки модели и количество входов в них требований (покупателей). При этом каждый блок имеет свой порядковый номер.

Еще ниже указываются результаты моделирования каналов обслуживания под назначенными нами именами **KASSIR** и **PRODAVEC** соответственно:

- **ENTRIES** (Число входов) – 107, 101 ;
- **UTIL.** (Коэффициент использования) – 0.244; 0.987;
- **AVE. TIME** (Среднее время обслуживания) – 2.318; 9.929;
- **AVAIL.** (Доступность) – 1, 1;
- **OWNER** (Возможное число входов) – 0, 101;
- **PEND** – 0, 0;
- **INTER** – 0,0;
- **RETRY** (Повтор) – 0, 0;
- **DELAY** (Отказано) – 0, 6.

Заметим, что разница между числом входов в каналы **KASSIR** и **PRODAVEC** объясняется возникшей во втором канале очередью.

Далее указываются результаты моделирования каждой очереди под присвоенными нами именами **OCHER_PROD** и **OCHER_KASSA** соответственно:

- **MAX** (Максимальное содержание) – 7 и 1;
- **CONT.** (Текущее содержание) – 7 и 0;
- **ENTRY** (Число входов) – 107 и 107;
- **ENTRY(0)** (Число нулевых входов) – 2 и 107;
- **AVE.CONT.** (Среднее число входов) – 3.607 и 0.000;
- **AVE.TIME** (Среднее время) – 34.260 и 0.000;
- **AVE.(-0)** – 34.913 и 0.000;
- **RETRY** – 0 и 0.

Еще ниже указываются результаты моделирования для построения по табличным данным гистограммы **T_PROD** функционирования очереди под именем **OCHER_PROD**:

- **MEAN** (Средняя) – 34.338;
- **STD.DEV.** (Среднее квадратическое отклонение) – 17.466;
- **RANGE** (Область);
- **RETRY** – 0;
- **FREQUENCY** (Частота);
- **CUM.%** (Суммарный процент).

Благодаря оператору **t_prod QTABLE Ocher_prod,0,2,32** можно вывести соответствующую гистограмму. Для этого:

1) щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **Simulation Window** (Окно моделирования) выпадающего меню. Появится всплывающее меню;

3) щелкните по пункту **Table Window** (Окно гистограммы) во всплывающем меню. Появится диалоговое окно **Open Table Window** (Открыть окно гистограммы). В раскрывающемся списке **Table** щелкните по нужной гистограмме с именем **T_PROD**.

4) щелкните по кнопке **ОК**. Появится соответствующая гистограмма (рис. 5.10).

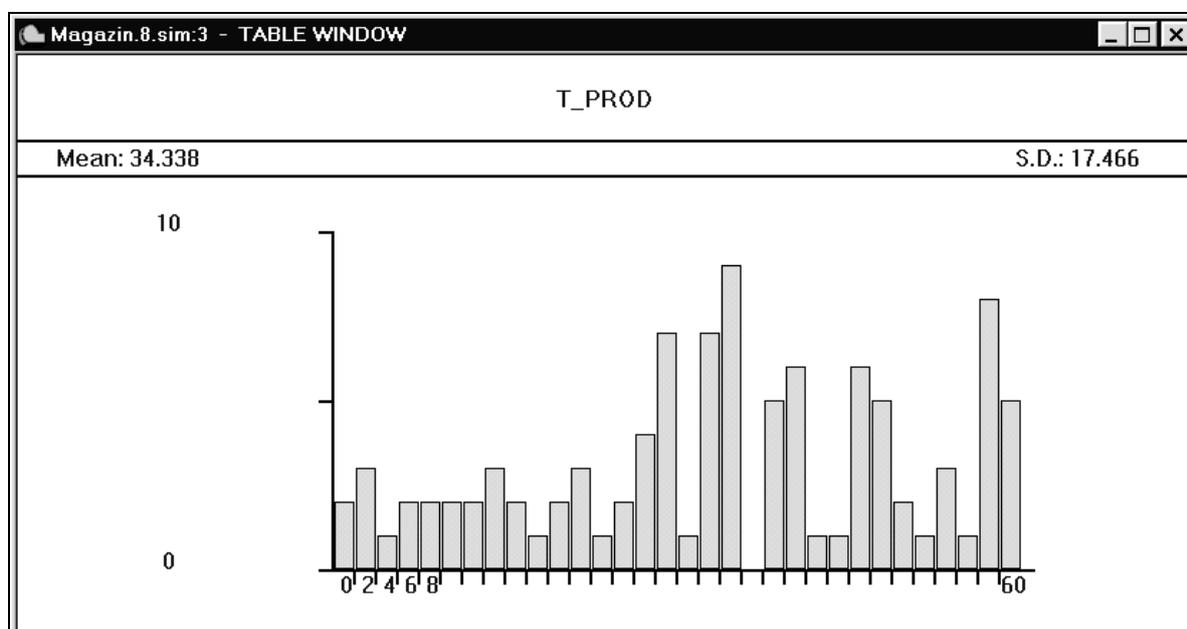


Рис. 5.10. Гистограмма очереди к продавцу

Эта гистограмма показывает распределение времени ожидания в очереди к продавцу. Среднее значение пребывания в очереди составляет 34.338 мин, среднеквадратичное отклонение – 17.446 минут.

Визуализация процесса функционирования системы. Система GPSSW обеспечивает возможность визуального наблюдения перемещения покупателей (активных требований) в процессе моделирования. Если в модели есть команда управления **START**, она должна быть заблокирована, т.е. переведена в комментарии. Для этого в позиции 1 поставьте звездочку. Для включения режима визуального

наблюдения перемещения покупателей в процессе моделирования выполните следующие действия:

1) щелкните по пункту **Window** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **Simulation Window** выпадающего меню. Появится всплывающее меню;

3) щелкните по пункту **Block Window** (Блочные элементы) всплывающего меню. Появится диалоговое окно **BLOCK ENTITIES** (рис. 5.11).

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	0	0	7	0
2 QUE	QUEUE	0	0	0	8	0
3 SEI	SEIZE	0	0	0	9	0
4 DEP	DEPART	0	0	0	10	0
5 ADV	ADVANCE	0	0	0	11	0
6 REL	RELEASE	0	0	0	12	0
7 QUE	QUEUE	0	0	0	13	0
8 SEI	SEIZE	0	0	0	14	0
9 DEP	DEPART	0	0	0	15	0
10 ADV	ADVANCE	0	0	0	16	0
11 REL	RELEASE	0	0	0	17	0
12 TER	TERMINATE	0	0	0	18	0

Рис. 5.11. Блок-схема модели магазина

Для визуализации перемещения активных требований (транзактов) в процессе моделирования:

1) щелкните по пункту **Command** главного меню системы. Появится выпадающее меню;

2) щелкните по пункту **START**. Появится диалоговое окно **Start Command**;

3) введите число покупателей, которые собираются посетить магазин, например 100;

4) щелкните по кнопке **OK**. Начнется процесс поступления и перемещения активных требований (покупателей) в магазине. Каждое перемещение требования по блокам системы фиксируется в правой части окна **BLOCK ENTITIES**;

5) щелкните по кнопке **Halt** (Остановить), расположенной на панели кнопок управления в верхней правой части окна **BLOCK ENTITIES**. Одно из состояний моделирования системы магазина представлено на рис. 5.12;

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-file
1 GEN	GENERATE	0	79	0	7	0
2 QUE	QUEUE	0	79	0	8	0
3 SEI	SEIZE	0	79	0	9	0
4 DEP	DEPART	0	79	0	10	0
5 ADV	ADVANCE	1	79	0	11	0
6 REL	RELEASE	0	78	0	12	0
7 QUE	QUEUE	5	78	0	13	0
8 SEI	SEIZE	0	73	0	14	0
9 DEP	DEPART	0	73	0	15	0
10 ADV	ADVANCE	1	73	0	16	0
11 REL	RELEASE	0	72	0	17	0
12 TER	TERMINATE	0	72	0	18	0

Рис. 5.12. Одно из состояний моделирования системы магазина в подробном представлении

б) щелкните по кнопке **Continue** (Продолжить) для продолжения моделирования или по кнопке **Step** (Шагнуть), чтобы промоделировать и посмотреть изменения в системе в течение одного шага. По кнопке **Step** можно щелкать многократно.

В первом столбце **Loc** окна (рис. 5.12) представлены имена блоков модели, а во втором столбце **Block Type** – имена соответствующих операторов. Прямоугольные значки слева от имен блоков означают наличие в данном блоке обслуживаемых покупателей.

В столбце **Current Count** показано, сколько покупателей в выбранный момент времени обслуживается данным блоком. В столбце **Entry Count** показано, сколько всего с самого начала процесса моделирования покупателей «вошли» в данный блок.

Как видно из рис. 5.12, всего к данному моменту времени в систему «вошли» 79 покупателей. В очереди к кассиру покупателей нет. Один покупатель обслуживается кассиром, пять покупателей стоят в очереди к продавцу, один покупатель обслуживается продавцом.

Против блока **TERMINATE** стоит число 72 – столько покупателей к данному моменту времени покинули магазин.

Рассмотренная в данном разделе задача иллюстрирует возможности системы GPSS World. Перейдем теперь к рассмотрению объектов и операторов языка для решения задач имитационного моделирования GPSS.

5.5. Объекты системы GPSS World

Язык для решения задач имитационного моделирования GPSS – объектно-ориентированный. Основными элементами этого языка являются *транзакты* и *блоки*, которые отображают соответственно динамические и статические объекты моделируемой системы.

Предназначение объектов системы различно. Выбор объектов конкретной модели зависит от характеристик моделируемой системы. Каждый объект имеет некоторое число свойств, названных в GPSS стандартными числовыми атрибутами (СЧА).

Блоки и транзакты. Каждая GPSS-модель обязательно должна содержать такие объекты, как блоки и транзакты.

В GPSS концепция передачи управления от блока к блоку имеет специфические особенности. Последовательность блоков GPSS модели показывает направления, в которых перемещаются элементы. Каждый такой элемент называется *транзактом*. Транзакты – это динамические элементы GPSS-модели.

Блоки языка GPSS представляют собой подпрограммы, содержащие набор параметров (операндов) для обращения к ним. Как и во всех языках моделирования, в GPSS существует внутренний механизм передачи управления, который реализуется в модельном времени, что дает возможность отобразить динамические процессы в реальных системах. Передача управления от блока к блоку в GPSS-программах реализуется с помощью движения транзактов в *модельном времени*. Обращение к подпрограммам блоков происходит через движение транзактов.

Содержательное значение транзактов определяет разработчик модели. Именно он устанавливает аналогию между транзактами и реальными динамическими элементами моделируемой системы. В табл. 5.2 приведены примеры аналогий между транзактами и элементами реальных систем.

С точки зрения программы *транзакт* – это структура данных, которая содержит такие поля: имя или номер транзакта; время появления транзакта; текущее модельное время; номер блока, в котором находится транзакт; номер блока, куда он продвигается; момент времени начала продвижения; приоритет транзакта; параметры транзакта.

Аналогия между транзактами и элементами реальных систем

Система	Элементы систем, которые моделируются транзактами
Магазин	Покупатель
Автомобильное шоссе	Автомобиль
Бригада слесарей-сборщиков, поочередно использующих некоторое оборудование	Слесарь-сборщик
Склад	Заявка

В языке GPSS все транзакты нумеруются по мере их появления в модели. Параметры транзактов отображают свойства моделируемого динамического объекта. Например, если моделируется работа магазина, то параметром транзакта (покупателя) может быть, например, сумма денег, которую он может потратить.

Таким образом, при начале моделирования в GPSS-модели не существует ни одного транзакта. В процессе моделирования транзакты входят в модель в определенные моменты времени, соответствующие логике функционирования моделируемой системы. Таким же образом транзакты покидают модель в зависимости от специфики моделирования. В общем случае в модели существует несколько транзактов, но в каждый момент времени движется только один из них. Поскольку модельное время имеет достаточно малый шаг дискретизации, то последнее положение вносит незначительную погрешность в расчеты.

Если транзакт начал свое движение, он передвигается от блока к блоку по пути, указанному блок-схемой (логикой работы модели). В тот момент, когда транзакт входит в блок, вызывается соответствующая этому блоку подпрограмма. Далее транзакт пытается войти в следующий блок. Его перемещение продолжается до тех пор, пока не выполнится одно из таких возможных условий:

1) транзакт входит в блок, функцией которого является задержка транзакта на определенное время;

2) транзакт входит в блок, функцией которого является удаление транзакта из модели;

3) в соответствии с логикой модели транзакт пытается войти в следующий блок, но блок не принимает этот транзакт. В этом случае транзакт остается в том блоке, в котором в данное время находится, но позже будет повторять попытки войти в следующий блок. Когда усло-

вия в модели изменятся, одна из таких попыток может быть успешной. После этого транзакт продолжит свое перемещение по модели.

Если выполняется одно из указанных условий, транзакт остается на месте, и в модели делается попытка перемещения другого транзакта.

Объекты типа «ресурсы». Аналогами обслуживающих устройств реальных систем в GPSS являются объекты типа «ресурсы». К объектам этого типа относятся *устройства, многоканальные устройства и логические ключи*. Как и в каждом объектно-ориентированном языке, в GPSS любой объект имеет *свойства и методы*, которые изменяют эти свойства. В GPSS свойства объектов называют *стандартными числовыми атрибутами (СЧА)*.

Устройство (одноканальное устройство, прибор) представляет собой ресурс, который в любой момент времени может быть занят только одним транзактом. Интерпретатор автоматически вычисляет такие его СЧА, как общее время занятости устройства, число транзактов, которые занимали устройство, коэффициент использования устройства, среднее время занятости устройства одним транзактом и т.п.

Многоканальные устройства (МКУ), состоящие из нескольких параллельных одинаковых устройств, представляют собой объекты типа «ресурсы» для параллельной обработки. Они могут быть использованы несколькими транзактами одновременно. Пользователь определяет емкость каждого МКУ, который используется в модели, а интерпретатор ведет учет числа устройств, занятых в каждый момент времени. Интерпретатор также автоматически подсчитывает такие СЧА: число транзактов, которые вошли в МКУ; среднее число каналов, занятых одним транзактом; среднее время нахождения транзакта в устройстве и ряд других.

Некоторые события в системе могут заблокировать или изменить движение транзактов. Например, кассир магазина, уходя на обед, ставит табличку «К другому кассиру», и все следующие клиенты на протяжении обеда обращаются в другую кассу. Для моделирования этих ситуаций введены логические ключи. Транзакт может устанавливать эти ключи в положение «Включено» или «Выключено». Через некоторое время состояние ключа может быть использовано другими транзактами для выбора одного из двух возможных путей движения или ожидания момента изменения состояния ключа. Состояние ключа может быть изменено любым транзактом.

Переменные. *Арифметические переменные* позволяют вычислять арифметические выражения, которые состоят из операций над СЧА объектов. В выражениях могут быть использованы функции (библиотечные или пользовательские). *Булевы переменные* позволяют пользователю одновременно проверять несколько условий, исходя из состояния объектов или значений СЧА.

Функции. Используя *функции*, пользователь может задавать непрерывную или дискретную функциональную зависимость между аргументом функции и ее значением. Функции в GPSS задаются табличным способом с помощью операторов описания функций.

Ячейки и матрицы сохраняемых величин. Ячейки сохраняемых величин и матрицы используются для хранения некоторой пользовательской числовой информации, запись в эти объекты выполняют транзакты. Записанную в этих объектах информацию может считать любой транзакт. Таким образом, эти объекты являются глобальными и доступны из любой части модели.

Очереди. В любой системе движение потока транзактов может быть задержано из-за недоступности ресурсов (например, необходимые устройства или МКУ уже заняты). В этом случае задержанные транзакты становятся в *очередь* – еще один тип объектов GPSS. Учет этих очередей составляет одну из основных функций интерпретатора.

Пользователь может специально определить точки модели, в которые необходимо собирать статистику об очередях, т.е. установить *регистраторы очереди*. Тогда интерпретатор будет автоматически собирать статистику об очередях (длину очереди, среднее время нахождения в очереди и т.п.). Вся эта информация является СЧА и доступна пользователю в процессе моделирования.

Интерпретатором автоматически поддерживается дисциплина обслуживания очереди FIFO («первым пришел – первым обслужился»), и пользователь может получить стандартную статистическую информацию только об этих очередях. Если у пользователя возникает необходимость организовать очередь из транзактов с другой дисциплиной обслуживания (например, LIFO – «последним пришел – первым обслужился»), то для этого используются *списки пользователей*. Эти списки также помогают осуществлять синхронизацию движения разных транзактов по модели.

Таблицы. Объект «*таблица*» предназначен для сбора статистики о случайных величинах, заданных пользователем. Таблица состоит

из частотных классов, в которые заносится число попаданий конкретной величины (некоторого СЧА). Для каждой таблицы вычисляется математическое ожидание и среднеквадратическое отклонение некоторого СЧА.

ЧАСЫ модельного времени. Разные события реальных систем происходят в течение некоторого периода времени. Например, покупатели приходят в магазин, когда подходит их очередь, они попадают на обслуживание. Когда покупки сделаны, покупатели покидают магазин. Если все эти события представить в модели, то их возникновение должно происходить на фоне модельного времени. Интерпретатор автоматически обслуживает ЧАСЫ модельного времени.

В момент начала моделирования интерпретатор планирует появление первого транзакта. После этого ЧАСЫ модельного времени устанавливаются на значение времени, которое соответствует моменту появления первого транзакта в модели. Этот транзакт (и другие, если они приходят в этот же момент времени) входит в модель. Далее он передвигается через все возможные блоки модели, которые ему встречаются. События, которые возникают вследствие перемещений транзакта через блоки, планируются на дальнейшие моменты времени. Интерпретатор GPSS продвигает дальше значения ЧАСОВ к тому значению времени, на которое запланировано следующее ближайшее событие. Если во второй, отмеченный ЧАСАМИ момент времени нет транзактов, которые нужно перемещать, ЧАСЫ снова продвигаются вперед и т.д. Именно так, от события к событию, и происходит смена модельного времени.

Отметим следующие особенности ЧАСОВ GPSS:

1. Единица модельного времени определяется разработчиком. Значение принятой единицы модельного времени выражают в неявном виде в форме переменных данных модели. Так, если все данные выражены в минутах, то единицей времени будет минута, т.е. масштаб времени в модели будет такой: одна единица модельного времени равна одной минуте реального времени. Если все данные выражены в миллисекундах, то единицей модельного времени будет миллисекунда. Разработчик может задавать такую единицу времени, которая ему удобна, для того, чтобы правильно отобразить события реальной системы в модели. Единица модельного времени всегда принимает целочисленные значения.

2. Система GPSS является интерпретатором «следующего события». Иначе говоря, после того как модель полностью скорректирована в

данный момент дискретно изменяющегося времени, ЧАСЫ перемещаются к следующему моменту времени, на который запланировано следующее событие. Таким образом, ЧАСЫ модельного времени продвигаются от одного события к другому.

5.6. Типы операторов

Операторы GPSS делятся на три типа:

- 1) блоки;
- 2) операторы описания данных;
- 3) команды GPSS.

Общие сведения о формате операторов GPSS. В GPSS для ссылки на числа, блоки и объекты используются *имена* (идентификаторы). Имя представляет собой алфавитно-цифровую последовательность длиной до 250 символов, которая начинается с буквы. Допускается использование символов только латинского алфавита, цифр и знака подчеркивания.

Формат GPSS-блоков такой:

[< Метка >] < Операция > < Операнды > <; Комментарии >

Метка (имя блока). Содержимым поля является имя – последовательность символов, начинающаяся с буквы. В некоторых операторах это поле является обязательным.

Операция. Операциями блоков являются глаголы, которые описывают основные функциональные назначения блоков. Каждый из блоков характеризуется своим собственным, предписанным ему глаголом.

Операнды. Блоки могут иметь операнды. Операнды блоков задают информацию, специфичную для действия данного блока. Число операндов блока зависит от типа блока. В блоках не может использоваться больше семи операндов. Операнды в общем случае обозначаются символами: A, B, C, D, E, F, G. Значения операндов определяются типом блока. Одни операнды некоторых блоков должны быть определены всегда, а другие могут задаваться или не задаваться (т.е. являются необязательными). Операнды следуют один за другим и отделяются запятыми или одним пробелом. Если операнд опущен, то вместо него ставится запятая. Между операндами не должно быть более одного пробела, так как это будет означать, что операнды закончились и интерпретатор прекращает чтение строки.

Комментарии. Необязательное поле. Комментарии отделяются от поля операндов символом «;». Допускается запись комментария с начала строки. В этом случае в первой позиции строки ставится символ «;» или «*». Допускается использование символов кириллицы.

Строка описания блока может содержать до 250 символов. При описании форматов квадратные скобки [] указывают на необязательность поля.

Именами и метками не могут быть названия или начальные символы названий блоков, операторов, команд и СЧА. Во избежание конфликтов с ключевыми словами рекомендуется в именах использовать символ подчеркивания.

5.7. Основные блоки GPSS World

Внесение транзактов в модель. Блок GENERATE. Блок **GENERATE** (**ГЕНЕРИРОВАТЬ**) – это блок, через который транзакты входят в модель. Не существует ограничений на количество разных блоков **GENERATE** в одной модели.

Интервал времени между последовательными появлениями транзактов из блока **GENERATE** называют *интервалом поступления*. Когда транзакт входит в модель через блок **GENERATE**, интерпретатор планирует время поступления следующего транзакта путем розыгрыша случайного числа с соответствующим распределением интервалов поступления на время, равное текущему значению **ЧАСОВ** плюс разыгранное значение. При достижении этого значения модельного времени следующий транзакт вводится в модель через блок **GENERATE** и т.д.

Разработчик должен задать функцию распределения интервалов поступления транзактов в блоке **GENERATE**.

Все возможные виды случайных распределений интервалов поступления транзактов в GPSS делятся на равномерное распределение и другие виды распределений. Пока мы будем рассматривать самое простое из всех случайных распределений – равномерное распределение. Использование других видов распределений требует задания функций, которые будут описаны в других разделах.

Формат блока:

GENERATE [A],[B],[C],[D],[E]

Описания операндов блока **GENERATE** приведены в табл. 5.3.

Таблица 5.3

Параметры блока **GENERATE**

Операнд	Значение	Значение по умолчанию
A	Средний интервал времени (число, СЧА)	0
B	Половина поля допуска равномерно распределенного интервала (число, СЧА)	0
C	Смещение интервалов	Смещение отсутствует
D	Ограничитель транзактов	∞
E	Уровень приоритета транзакта. Возможные значения 0 - 127	0

Значение операндов:

A – среднее значение интервала поступления;

B – величина разброса возможных значений относительно среднего значения (если операнд **B** не задается, то интервал времени поступления – детерминированная величина);

C – момент времени, в который в блоке **GENERATE** должен появиться первый транзакт (после этого первого прихода все остальные приходы транзактов возникают в соответствии с распределением, заданным операндами **A** и **B**);

D – ограничитель общего числа транзактов, которое может войти в модель через данный блок **GENERATE** на протяжении времени моделирования (если это число достигнуто, данный блок **GENERATE** перестает быть активным);

E – уровень или класс приоритета каждого из транзактов, которые вводятся в модель через данный блок **GENERATE** (всего существует 128 разных уровней, которые задаются с помощью чисел от 0 до 127; чем больше число, тем выше приоритет).

Отметим, что транзакты не могут входить в блок **GENERATE**, так как он сам их генерирует.

Пример 5.1. Задание равномерного закона распределения:

GENERATE 6,4

Операнды: **A** = 6, **B** = 4. Интервал времени поступления является случайным числом со средним значением 6 и полем допуска 8, т.е. он может приобретать значения в интервале от 2 до 10.

Пример 5.2. Задание детерминированного значения интервалов поступления:

GENERATE 10

Операнды: **A** = 10, **B** = 0 (по умолчанию). Транзакты входят в модель каждые 10 единиц модельного времени.

Пример 5.3. Генерирование одного транзакта:

GENERATE „,1

Операнды: **A=B=C=0** (по умолчанию), **D** = 1. В нулевой момент в модель входит один транзакт.

Блоки **GENERATE** являются основными средствами создания транзактов и ввода их в модель. Кроме блока **GENERATE** для ввода транзактов в модель используется также блок **SPLIT**, который создает заданное число копий транзактов, вошедших в блок.

Удаление транзактов из модели. Блок TERMINATE. Транзакты удаляются из модели, попадая в блок **TERMINATE** (**ЗАВЕРШИТЬ**).

В модели может быть любое количество блоков **TERMINATE**.
Формат блока:

TERMINATE [A]

Операнд **A** является величиной уменьшения специального счетчика, который называется *счетчиком завершения*. Этот операнд задает величину, которая вычитается из счетчика каждый раз, когда транзакт входит в блок **TERMINATE**. По умолчанию **A** = 0. Вход транзакта в блок **TERMINATE** с нулевым значением операнда **A** не вызывает уменьшения счетчика завершения.

Счетчик завершения – это ячейка в памяти ПК, которая хранит целое положительное число. Начальное значение этого счетчика устанавливается в начале моделирования. Оно равняется значению операнда **A** команды **START** (**НАЧАТЬ**). В процессе моделирования транзакты попадают в блок **TERMINATE** и, таким образом, уменьшают значение счетчика на величину операнда **A**. Моделирование заканчивается, когда значение счетчика становится равным нулю или отрицательному числу.

В модели может быть много блоков **TERMINATE**, но счетчик завершения один, с начальным значением, указанным в команде **START**.

Не следует путать ограничитель транзактов в блоке **GENERATE** и счетчик завершения. Ограничитель задает число транзактов, которые войдут в модель, а счетчик – число транзактов, которые выйдут из модели. По окончании моделирования транзакты могут оставаться в модели.

Команда START. Интерпретатор начинает моделирование по команде **START**. Ее формат:

START A,[B],[C],[D]

Управление продолжительностью процесса моделирования. В языке GPSS продолжительностью процесса моделирования можно управлять двумя способами:

- 1) завершать моделирование после того, как модель покинет заданное число транзактов определенного типа;
- 2) завершать моделирование по истечении заданного интервала времени.

Первый способ:

1. В команде **START** операнду **A** присваивается значение заданного числа транзактов.

2. Во всех блоках **TERMINATE**, через которые транзакты заданного типа покидают модель, операнду **A** присваивается значение «1» или другое, отличное от нуля (соответственно содержательному значению транзактов).

3. Во все других блоках **TERMINATE** используется значение операнда **A** по умолчанию ($A = 0$). Значение счетчика завершения не будет зависеть от этих блоков.

Первый способ позволяет закончить моделирование, когда через модель пройдет заданное количество транзактов, например 1000:

GENERATE 40,5

...

TERMINATE 1

START 1000

Примечание. Во многих случаях удобно разделить во времени процессы трансляции и выполнения программы. В этом случае сле-

дует заблокировать команду **START** (ввести слева символ "*"). При запуске программы на выполнение в соответствующем диалоговом окне ввести требуемое значение, например **START 1000**.

Второй способ:

Пусть разработчик выбрал за единицу модельного времени 1 мин и хочет смоделировать поведение системы на протяжении 8 часов (480 минут). Это можно сделать таким образом:

1. Ввести в модель таймер-сегмент, состоящий из двух блоков:

```
GENERATE 480  
TERMINATE 1
```

2. Во всех других блоках **TERMINATE** в модели использовать значение операнда **A** по умолчанию (**A=0**). Это означает, что прекращение моделирования, определяемое счетчиком завершения, не будет зависеть от других блоков **TERMINATE**.

3. В команде **START** операнд **A** должен равняться единице.

Таким образом, в процессе моделирования завершение движения транзактов в других блоках **TERMINATE** не влияет на счетчик завершения. В момент времени 480 транзакт выйдет из блока **GENERATE** и сразу же перейдет в блок **TERMINATE**. Счетчик завершения уменьшится на единицу, и интерпретатор закончит моделирование.

5.8. Элементы, отображающие одноканальные обслуживающие устройства

Рассмотрим элементы, которые используются для представления обслуживания. Аналогами обслуживающих элементов могут быть люди, механизмы, линии связи и другие объекты реальных систем. В GPSS такие объекты моделируются с помощью *устройств*.

Устройство характеризуется двумя основными свойствами:

1. Каждое устройство в любой момент времени может обслуживать только один транзакт. Если в процессе обслуживания появляется новый транзакт, то он должен:

- а) либо подождать своей очереди,
- б) либо направиться в другое место,
- в) либо, если вновь пришедший транзакт имеет больший приоритет, устройство прерывает текущее обслуживание и начинает обслуживать новый транзакт.

2. Когда транзакт поступает в устройство, он должен пробыть там необходимое для обслуживания время.

Всем устройствам следует задавать имена. Они могут быть или числовыми (числа должны быть положительными целыми), или символьными. Во время трансляции символьным именам сам транслятор присваивает числовые значения.

Для того чтобы использовать одноканальное обслуживающее устройство (прибор), транзакту необходимо выполнить четыре шага.

Первый шаг. Ждать своей очереди, если это необходимо. Ожидание длится в течение некоторого интервала времени.

Второй шаг. Когда подходит очередь, занять устройство. Событие «занятие устройства» происходит в некоторый момент модельного времени.

Третий шаг. Устройство находится в состоянии занятости до тех пор, пока не закончится обслуживание. Для обслуживания необходим некоторый интервал времени.

Четвертый шаг. Когда обслуживание закончится, освободить устройство. Событие «освобождение устройства» происходит в некоторый момент модельного времени.

Эта последовательность шагов выполняется GPSS при моделировании использования устройства. Второй и четвертый шаги реализуются блоками **SEIZE (ЗАНЯТЬ)** и **RELEASE (ОСВОБОДИТЬ)**.

Формат блока:

SEIZE A ,

где **A** – имя занимаемого устройства.

Этот блок имеет следующие свойства:

1. Если в текущий момент времени устройство используется, то транзакт не может войти в блок и должен ожидать своей очереди;

2. Если устройство свободно, транзакт может войти в блок. Вход транзакта в блок вызывает выполнение подпрограммы обработки этого блока. Состояние устройства изменяется со СВОБОДНОЕ на ЗАНЯТОЕ.

Предварительного объявления устройства в модели не требуется, так как тот факт, что блок **SEIZE** используется, свидетельствует о существовании данного устройства.

Предназначением блока **RELEASE** является изменение состояния ранее занятого устройства с **ЗАНЯТОГО** на **СВОБОДНОЕ**. Блок **RELEASE** никогда не запрещает вход транзакта. Формат блока:

RELEASE A ,

где **A** – имя занимаемого устройства.

В то время как транзакты находятся в модели временно, устройства, используемые в модели, существуют в ней в течение всего периода моделирования.

Статистическая информация о работе устройства при моделировании собирается автоматически.

Если в модели используются объекты типа «устройство», то в файле стандартной статистики будет представлена информация об использованных устройствах.

Статистику работы устройств в процессе моделирования можно наблюдать в окне **Facilities Window**:

FACILITY – номер устройства,

ENTRIES – количество входов,

UTIL – коэффициент использования,

AVE.TIME – среднее время пребывания транзакта в устройстве,

AVAIL. – состояние готовности,

OWNER – номер последнего транзакта, занявшего устройство,

DELAY – количество транзактов, ожидающих занятия устройства.

После блока **SIEZE** может сразу же следовать другой блок **SIEZE**, если транзакт должен одновременно занять два или более устройств (например, рабочего и инструмент).

Транзакт не может освободить устройство, которое он не занимал.

Блок ADVANCE. Перевод с английского языка названия блока **ADVANCE (ЗАДЕРЖАТЬ)** – продвигать, а не задерживать. Этот блок действительно продвигает **ЧАСЫ** модельного времени на некоторое значение, но фактически он осуществляет задержку продвижения транзакта в течение некоторого интервала времени. Обычно этот интервал задается случайной величиной.

В **GPSS** возможны следующие варианты распределения времени обслуживания:

1) детерминированное (постоянное);

2) равномерное распределение;

3) другие распределения.

Пока ограничимся рассмотрением равномерного распределения случайных величин. Применение более сложных видов распределений требует использования дополнительных функций.

Формат блока:

ADVANCE A[,B]

Таблица 5.4

Параметры оператора ADVANCE

Операнд	Значение	Значение по умолчанию
A	Среднее время задержки на обслуживание	0
B	Половина допуска поля равномерно распределенного времени задержки	0

Блок никогда не препятствует входу транзакта. Любое число транзактов может находиться в этом блоке одновременно. Когда транзакт попадает в такой блок, выполняется соответствующая подпрограмма и вычисляется время пребывания в нем транзакта. Вновь прибывший транзакт никак не влияет на уже находящийся в блоке транзакт.

Если время пребывания в блоке равно нулю, то вместо задержки в блоке **ADVANCE** интерпретатор сразу же пытается переместить этот транзакт в следующий блок.

Пример 5.4. Использование блока **ADVANCE**:

ADVANCE 30,5

Время задержки транзакта в этом блоке – случайная величина, равномерно распределенная на интервале [25, 35].

Пример 5.5. Типичный случай использования последовательности **SEIZE – ADVANCE – RELEASE**:

SEIZE PRIB

ADVANCE 16,4

RELEASE PRIB

Транзакт, двигаясь по этой цепочке блоков, займет устройство с именем **PRIB**, задержится там на 16 ± 4 единицы времени и затем покинет его. После того как транзакт войдет в блок **RELEASE** и соот-

ветствующая этому блоку подпрограмма закончится, интерпретатор попытается переместить транзакт в следующий блок модели и следующий транзакт может уже использовать устройство **PRIB**.

Блоки **ADVANCE** можно располагать в любых местах программы, а не только между блоками **SEIZE** и **RELEASE**.

Сбор статистики об ожидании. Блоки *QUEUE, DEPART*. Эти блоки обеспечивают в GPSS возможность автоматического сбора статистических данных, описывающих вынужденное ожидание, которое может происходить время от времени в различных точках модели.

Система моделирования GPSS обеспечивает возможность сбора статистики с помощью такого средства, как *регистратор очереди*.

При использовании регистратора очереди в тех точках модели, где число ресурсов ограничено, интерпретатор автоматически начинает собирать различную информацию об ожидании с помощью СЧА, а именно:

- 1) число входов транзактов в очередь;
- 2) количество транзактов, которые фактически присоединились к очереди и сразу ее покинули, т.е. имели время ожидания, равное нулю;
- 3) максимальная длина очереди;
- 4) среднее число ожидавших транзактов;
- 5) среднее время ожидания тех транзактов, которым пришлось ждать.

В модели может быть несколько регистраторов очередей, различающихся именами. Правила присвоения имен те же, что и для устройств. Разработчик вносит регистратор очереди в модель с помощью пары взаимодополняющих блоков:

**QUEUE A[,B]
DEPART A[,B]**

Таблица 5.5

Параметры операторов QUEUE и DEPART

Операнд	Значение	Результат по умолчанию
A	Имя очереди, в которую необходимо стать транзакту или которую надо покинуть (числовое или символическое имя, СЧА)	Ошибка
B	Число единиц, на которое увеличивается (уменьшается) длина очереди (число, СЧА)	1

При входе транзакта в блок **QUEUE (СТАТЬ В ОЧЕРЕДЬ)** выполняются четыре действия:

- 1) *счетчик входов* для данной очереди увеличивается на **В**;
- 2) *длина очереди (счетчик текущего содержимого)* для данной очереди увеличивается на **В**;
- 3) значение текущей *длины очереди* хранится в стандартном числовом атрибуте **Q\$<имя очереди>**;
- 4) транзакт присоединяется к очереди с запоминанием ее имени и значения текущего модельного времени.

Транзакт перестает быть элементом очереди только после того, как он переходит в блок **DEPART (ПОКИНУТЬ ОЧЕРЕДЬ)**, соответствующий очереди.

Когда это происходит, интерпретатор выполняет такие операции:

- 1) длина очереди уменьшается на **В**;
- 2) ликвидируется «привязка» транзакта к очереди.

Если в модели используются объекты типа «очередь», то в файле стандартной статистики будет представлена информация об этих объектах. В конце моделирования интерпретатор автоматически выдает статистические данные:

QUEUE – имя очереди;

MAX – максимальная длина очереди;

CONT. – текущая длина очереди;

ENTRY – общее количество входов транзактов в очередь;

ENTRY(0) – количество «нулевых» входов, т.е. входов, соответствующих транзактам с нулевым пребыванием в очереди;

AVE.CONT – средняя длина очереди;

AVE.TIME – среднее время пребывания транзактов в очереди.

Пример 5.6.

Пусть необходимо собрать статистику об ожидании в очереди при обслуживании устройством **PRIB**, тогда в сегмент модели будут введены блоки **QUEUE** и **DEPART**:

```
QUEUE QPRIB  
SEIZE PRIB  
DEPART QPRIB  
ADVANCE 16,4  
RELEASE PRIB
```

В этом примере все транзакты, попадающие в устройство, должны пройти через пару **QUEUE – DEPART** даже тогда, когда устройство свободно и его можно сразу же занять.

Отметим основные свойства блоков **QUEUE** и **DEPART**.

1. Когда транзакт входит в блок **QUEUE**, то ищется очередь с именем, определенным операндом **A**. При необходимости очередь создается.

2. Блок **QUEUE** не поддерживает список членов очереди, он только добавляет единицы к длине очереди.

3. Использование регистратора очереди необязательно. С его помощью интерпретатор собирает лишь статистику об ожидании. Если же регистратор не используется, то статистика не собирается, но везде, где должна возникать очередь, она возникает. Ожидание является следствием состояния устройства, а не следствием использования регистратора. Если в планы не входит обработка статистических данных об очередях, то лучше не собирать статистику – это сэкономит время, расходуемое на моделирование.

4. Один и тот же транзакт может одновременно увеличить длину нескольких очередей.

5. При выходе транзакта из очереди через блок **DEPART** транзакту не обязательно уменьшать длину очереди на ту же величину, на которую он увеличил ее при входе в блок **QUEUE**. Но в итоге число входов в очередь должно равняться числу выходов из нее.

5.9. Переход транзакта в блок, отличный от последующего

Блок TRANSFER. В GPSS блок **TRANSFER (ПЕРЕДАТЬ)** может быть использован в девяти разных режимах. Рассмотрим три основных.

1. Блок TRANSFER в режиме безусловной передачи. Его формат:

TRANSFER ,B

Таблица 5.6

Параметры оператора TRANSFER

Операнд	Значение	Результат по умолчанию
A	Не используется	–
B	Позиция блока, в которую должен перейти транзакт	Ошибка

Позиция блока – это номер или метка блока. Так как операнд **A** не используется, то перед операндом **B** должна стоять запятая. В режиме безусловной передачи блок **TRANSFER** не может отказывать транзакту во входе.

2. Статистический режим. В этом режиме осуществляется передача транзакта в один из двух блоков случайным образом. Формат блока:

TRANSFER A,[B],C

Таблица 5.7

Параметры оператора **TRANSFER** в статистическом режиме

Операнд	Значение	Результат по умолчанию
A	Вероятность передачи транзакта в блок C , задаваемая в долях тысячи	Ошибка
B	Позиция блока, в которую должен перейти транзакт (с вероятностью 1 - A)	Следующий по порядку блок
C	Позиция блока, в которую должен перейти транзакт (с вероятностью A)	Ошибка

При задании вероятности (операнд **A**) используется не более трех цифр. Первый символ записи частоты «.» (десятичная точка), если используется действительное число, которое должно быть в пределах от 0 до 1,0 (например, 0,235). Если операнд – целое число, то вероятность интерпретируется в долях тысячи.

Пример 5.7.

```

TRANSFER .333,LPRIB1,LPRIB2
...
LPRIB1 SEIZE PR1
...
LPRIB2 QUEUE QPR2
...

```

С частотой 0,667 транзакт переходит в блок с меткой **LPRIB1**, с частотой 0,333 – в блок с меткой **LPRIB2**.

Пример 5.8.

```
TRANSFER 4,,LPRIB2
SEIZE PR1
...
LPRIB2 QUEUE QPR2
...
```

С частотой 0,6 транзакт переходит в блок **SEIZE PR1**, с частотой 0,4 – в блок с меткой **LPRIB2**.

3. *Режим BOTH.* Если в операнде **A** стоит зарезервированное слово **BOTH**, то блок **TRANSFER** работает в режиме **BOTH**.

В этом режиме входящий транзакт сначала пытается перейти к блоку, указанному в операнде **B**. Если это сделать не удастся, транзакт пытается перейти в блок, указанный в операнде **C**. Если транзакт не сможет перейти ни к тому, ни к другому блоку, то он остается в блоке **TRANSFER** и при каждом просмотре списка текущих событий будет повторять в том же порядке попытки перехода до тех пор, пока не сможет выйти из блока **TRANSFER**.

Пример 5.9.

```
TRANSFER BOTH,LL1,LL2
...
LL1 SEIZE PRI1
...
LL2 SEIZE PRI2
...
```

Транзакт сначала пытается перейти в блок с меткой **LL1**. Если устройство **PRI1** занято, транзакт пытается войти в блок с меткой **LL2**. Если транзакт не может войти в этот блок (устройство **PRI2** также занято), он остается в списке текущих событий и повторяет эти попытки при каждом просмотре списка до тех пор, пока не выйдет из блока **TRANSFER**.

Если бы меткой **LL1** был помечен блок **QUEUE**, а не блок **SEIZE**, то все транзакты были бы направлены по метке **LL1**, так как в отличие от блока **SEIZE** блок **QUEUE** всегда готов принять транзакты.

5.10. Моделирование многоканальных устройств

Устройство в GPSS используют для моделирования одиночного устройства обслуживания. Два или более обслуживающих устройства, работающих параллельно, могут моделироваться в GPSS двумя или более одноканальными устройствами. Обычно это необходимо, когда отдельные устройства являются разнородными, например имеют различную интенсивность обслуживания.

Однако очень часто параллельно работающие устройства являются одинаковыми, и GPSS предоставляет для их моделирования объект, называемый многоканальным устройством (МКУ).

Количество устройств, которое моделируется каждым из МКУ, определяется пользователем. В этом смысле употребляют термин «емкость МКУ». Эта емкость заранее должна быть определена пользователем.

Блоки ENTER (ВОЙТИ) и LEAVE (ВЫЙТИ). Использование МКУ аналогично использованию одиночного устройства. Элементом, который занимает и использует МКУ, является транзакт. При моделировании МКУ события происходят в следующем порядке:

- 1) транзакт ожидает своей очереди, если это необходимо;
- 2) транзакт занимает устройство;
- 3) устройство осуществляет обслуживание на протяжении некоторого интервала времени;
- 4) транзакт освобождает устройство.

Блоки **ENTER** и **LEAVE** моделируют события 2 и 4.

Формат блоков:

ENTER A[,B]

LEAVE A[,B]

Таблица 5.8

Параметры блоков **ENTER** и **LEAVE**

Операнд	Значение	Результат по умолчанию
A	Имя МКУ	Ошибка
B	Количество занимаемых одновременно устройств	1

Когда транзакт входит в блок **ENTER**, интерпретатор выполняет следующие действия:

- 1) увеличивает *счетчик входов* МКУ на значение операнда **B**;
- 2) увеличивает *текущее содержимое* МКУ на значение операнда **B**;
- 3) уменьшает *доступную емкость* МКУ на значение операнда **B**.

Когда транзакт входит в блок **LEAVE**, интерпретатор выполняет обратные действия:

- 1) уменьшает *текущее содержимое* МКУ на значение операнда **B**;
- 2) увеличивает *доступную емкость* МКУ на значение операнда **B**.

Операнду **B** можно присвоить значение, отличное от единицы. Например, пусть транзакт моделирует корабль, а МКУ – причалы в порту. В зависимости от размера корабль может занимать несколько причалов, т.е. $B > 1$.

Если в модели используются объекты типа МКУ, то в файле стандартной статистики об этих объектах будет представлена такая информация:

- поле **STORAGE** определяет имя или номер МКУ;
- поле **CAP.** определяет емкость МКУ, заданную оператором **STORAGE**;
- поле **REMAIN** определяет количество единиц свободной емкости МКУ в конце периода моделирования;
- поле **MIN** определяет минимальное количество используемой емкости МКУ за период моделирования;
- поле **MAX** определяет максимальное количество используемой емкости МКУ за период моделирования;
- поле **ENTRIES** определяет количество входов в МКУ за период моделирования;
- поле **AVL.** определяет состояние готовности МКУ в конце периода моделирования: 1 – МКУ готов, 0 – не готов;
- поле **AVE.C** определяет среднее значение занятой емкости за период моделирования;
- поле **UTIL.** определяет средний коэффициент использования всех устройств МКУ;
- поле **RETRY** определяет количество транзактов, ожидающих специальных условий, зависящих от состояния МКУ;
- поле **DELAY** определяет количество транзактов, ожидающих возможности входа в блок **ENTER**.

Статистику о работе МКУ можно наблюдать в окне **Storages Window**.

Определение емкости МКУ. Все используемые в модели МКУ должны быть заранее описаны, т.е. должно быть определено количество однотипных устройств, входящих в МКУ. Для этого применяется оператор **STORAGE** (**ХРАНИЛИЩЕ** или **ПАМЯТЬ**), определя-

ющий емкость МКУ. Название **STORAGE** становится понятным, если представить себе, что МКУ – это автоматизированный склад или многоэтажный гараж с определенным числом мест, которое и задает этот оператор. В таких случаях МКУ определяет не количество одинаковых устройств для обслуживания, а количество одинаковых мест для хранения.

Формат оператора задания емкости МКУ:

Таблица 5.9

Параметры оператора задания емкости МКУ

Поле	Информация в поле
Метка	Символическое имя МКУ
Операция	STORAGE
Операнд А	Емкость МКУ

Пример 5.10.

Пусть система состоит из восьми механиков и десяти подъемных кранов, тогда в модель могут быть введены такие МКУ:

```
MECHANICIAN STORAGE 8
LIFTING_GRAIN STORAGE 10
```

В этом примере **MECHANICIAN** – имя МКУ «механики», а **LIFTING_GRAIN** – имя МКУ «подъемные краны».

Существует возможность периодически переопределять емкость МКУ при необходимости выполнения нескольких прогонов за один этап моделирования. Это делается введением в программу между операторами **START** предыдущего прогона и оператором **START** последующего прогона нового определения емкостей.

5.11. Программа «БИЗНЕС-КОРПОРАЦИЯ»

С использованием имитационного моделирования разработано большое количество моделей экономических систем самой различной природы: производственных, финансовых, банковских. Такие программы призваны помочь экономисту при анализе динамических процессов провести решение самых разнообразных задач типа

«что будет с предприятием, если изменить те либо иные параметры задачи...».

Одна из наиболее известных программ такого типа – программа «БИЗНЕС-КОРПОРАЦИЯ» производства МГУ. Это программа динамического моделирования предприятия реального сектора экономики.

Моделирование производится с шагом один месяц, период моделирования – до 73 месяцев.

Задаются параметры внешней среды: кредитная ставка коммерческого банка, ставки налогов, уровень инфляции, рыночные цены, сложившиеся на данный период на продукцию предприятия, цены поставщиков сырья и комплектующих и ряд других.

В качестве параметров внутренней среды можно задать производительность оборудования, размеры дивидендов на одну акцию, долю прибыли, направляемую на инвестиции в производство, уровень заработной платы.

Таким образом, программа «БИЗНЕС-КОРПОРАЦИЯ» позволяет учесть взаимодействие производственной фирмы:

- с рынком;
- с коммерческими банками;
- с бюджетом;
- с поставщиками;
- с наемным трудом.

Наиболее важная практическая задача, которую позволяет решать эта программа, – расчет затрат и себестоимости продукции. Имеется возможность решать оптимизационные задачи, в частности задачу минимизации затрат. «Проигрывая» различные варианты развития предприятия, менеджер может выявить факторы, в наибольшей степени влияющие на себестоимость продукции. Очевидно, что такая информация обладает большой ценностью и может стать основой для принятия решений по оптимизации производства.

В игре имеется интегральная оценка эффективности действий пользователя, называемая рейтингом. Цель пользователя заключается в достижении максимального значения этого показателя к концу игрового курса. Как видно из рис. 5.13, рейтинг имеют составляющие, определяемые вкладом чистой прибыли, вкладом рентабельности капитала и вкладом рыночной цены акции.

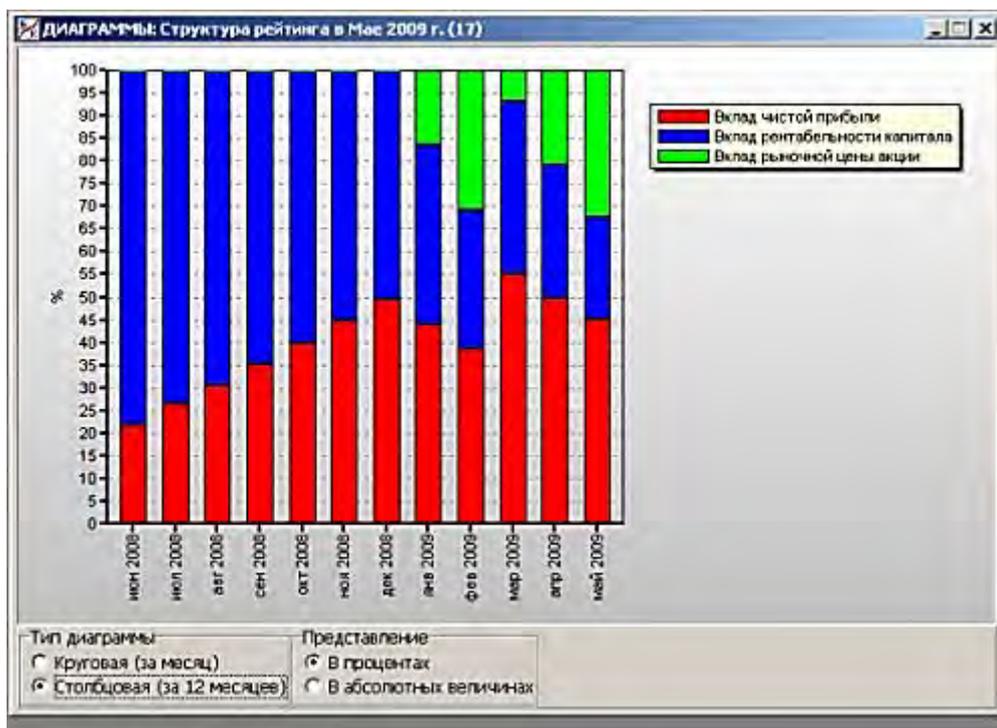


Рис. 5.13. Динамика рейтинга предприятия, рассчитываемого программой «БИЗНЕС-КОРПОРАЦИЯ»

Однако сам текст программы (исходные модули) скрыт от пользователя, который не может вносить в него необходимые корректировки. Именно поэтому такие программы, как «БИЗНЕС-КОРПОРАЦИЯ», не подходят для специальностей и направлений информационно-компьютерного профиля в качестве базовых при освоении курса «Имитационное моделирование экономических процессов». Однако знакомство с такими программами позволяет лучше понять возможности имитационного моделирования.

Большую известность получили работы по имитационному моделированию процессов в мировой экономике. Так, расчеты, проведенные рядом авторов, уже в 2005 г. позволили предсказать мировой финансово-экономический кризис, начавшийся во второй половине 2008 года. Однако, как это, увы, часто бывает, руководство крупнейших стран не обратило внимания на предсказания ученых. Меры по преодолению кризиса (такие как государственный контроль за деятельностью банков и корпораций) были приняты слишком поздно. В результате кризис может оказаться весьма длительным и тяжелым.

Вопросы для самоконтроля

1. В чем преимущества системы ИМ GPSS World перед другими аналогичными системами – AnyLogic и Pilgrim?
2. Перечислите основные возможности системы ИМ GPSS World.
3. Что включает главное меню системы ИМ GPSS World?
4. Как называется закладка для установки параметров отчета по результатам имитационного моделирования?
5. Как заполняется окно для вывода результатов моделирования в виде графиков?
6. Поясните информацию, выводимую в окне отчета **REPORT**: информация об устройствах (каналах) обслуживания, информация об очередях к устройствам.
7. Какие пункты меню выбираются для визуализации процесса функционирования системы?
8. Приведите примеры блоков и транзактов системы GPSS World.
9. Дайте определение многоканального устройства.
10. Перечислите особенности ЧАСОВ модельного времени.
11. Через какой блок транзакты входят в модель?
12. Что происходит с транзактами модели, которые попадают в блок **TERMINATE**?
13. Расскажите о способах управления продолжительностью процесса моделирования.
14. Назовите операторы занятия и освобождения устройства.
15. Для чего предназначен блок **ADVANCE**?
16. Какие блоки осуществляют сбор статистики об ожидании?
17. Приведите примеры режимов работы блока **TRANSFER**: статический, режим **BOTH**.
18. Какие блоки для моделирования многоканальных устройств (МКУ) используются в GPSS World?
19. Перечислите основные возможности программы «БИЗНЕС-КОРПОРАЦИЯ».



1. Модель парикмахерской

В парикмахерскую с одним креслом приходят клиенты двух типов. Клиенты первого типа желают только стричься. Распределение интервалов их прихода – 35 ± 10 минут. Клиенты второго типа желают постричься и побриться. Распределение интервалов их прихода – 60 ± 20 минут. Парикмахер обслуживает клиентов в порядке «первым пришел – первым обслужился». Время, затраченное на стрижку, составляет 18 ± 6 мин, а на бритье – 10 ± 2 минуты.

Постройте модель парикмахерской, обеспечив сбор данных об очереди клиентов. Моделирование провести для одной рабочей смены длительностью 480 минут.

Построение модели

Необходимо реализовать отличие в обслуживании клиентов, которые только стригутся, и клиентов, которые стригутся и бреются.

Такую систему можно промоделировать с помощью двух сегментов. Один из них моделирует обслуживание только стригущихся клиентов, а второй – стригущихся и бреющихся. В каждом из сегментов пара **QUEUE – DEPART** должна описывать одну и ту же очередь. Таким же образом пара блоков **SEIZE – RELEASE** должна описывать в каждом из двух сегментов одно и то же устройство и моделировать работу парикмахера.

Прежде чем писать программу, необходимо ответить на следующие вопросы: что является транзактами в каждом из сегментов данной модели? что является устройствами? какие очереди и к каким устройствам могут возникнуть? Ответы на эти вопросы содержатся в таблице.

Элементы программы и их интерпретация

Элементы GPSS	Интерпретация
<p style="text-align: center;">Транзакты</p> <p>В первом сегменте модели Во втором сегменте модели В третьем сегменте модели</p>	<p>Клиенты, которые только стригутся Клиенты, которые бреются и стригутся Таймер</p>
<p style="text-align: center;">Устройство</p> <p>BARBER</p>	<p>Парикмахер</p>
<p style="text-align: center;">Очередь</p> <p>BARBERQ</p>	<p>Очередь, используемая для сбора статистики об ожидании клиентов обоих типов</p>

Единицу модельного времени примем равной 1 минуте. Текст программы приведен ниже:

* * Модель парикмахерской	
* * Сегмент «Стрижка»	; Программа имитационного моделирования
GENERATE 35,10 QUEUE BARBERQ SEIZE BARBER DEPART BARBERQ ADVANCE 18,6 RELEASE BARBER TERMINATE 0	; Приход клиентов, которые только стригутся ; Присоединение к очереди ; Переход в кресло парикмахера ; Выход из очереди ; Стрижка у парикмахера ; Освобождение парикмахера ; Уход из парикмахерской
* * Сегмент «Стрижка и бритье»	
GENERATE 60,20 QUEUE BARBERQ SEIZE BARBER DEPART BARBERQ ADVANCE 10,2 ADVANCE 18,6 RELEASE BARBER TERMINATE 0	; Приход клиентов, которые стригутся и ; бреются ; Присоединение к очереди ; Переход в кресло парикмахера ; Выход из очереди ; Бритье у парикмахера ; Стрижка у парикмахера ; Освобождение парикмахера ; Уход из парикмахерской
* TIMER - Сегмент таймера	
GENERATE 480 TERMINATE 1 START 1	; Транзакт-таймер приходит в момент ; 480 ; Завершение прогона * В случае необходимости проведения на- * строек после трансляции команду за- * блокировать

Необходимо провести моделирование на ПК, получить отчет **REPORT** и проанализировать его.

Задания для самостоятельной работы

1. Выведите на экран график очереди к парикмахеру. Проведите расчет графика при следующих средних значениях интервала прихода клиентов, которые стригутся: 45, 25, 20, 15 минут. При каком значении среднего интервала прихода клиентов начинает образовываться очередь?
2. Постройте гистограмму времени пребывания в очереди клиентов, которые и стригутся, и бреются.

2. Задача о морских судах и буксирах

Морские судна двух типов прибывают в порт, где происходит их разгрузка. В порту есть два буксира, обеспечивающих ввод кораблей в порт и вывод из порта. К первому типу судов относятся корабли малого тоннажа, которые требуют использования одного буксира. Корабли второго типа имеют большие размеры, для их ввода и вывода из порта требуется два буксира. Из-за различия размеров двух типов кораблей необходимы и причалы различного размера. Кроме того, корабли имеют различное время погрузки-разгрузки. Исходные данные приведены в таблице:

Значение	Тип корабля	
	1	2
Интервал прибытия, мин	130 ± 30	390 ± 60
Время входа в порт, мин	30 ± 7	45 ± 12
Количество доступных причалов	6	3
Время погрузки-разгрузки, час	12 ± 2	18 ± 4
Время выхода из порта, мин	20 ± 5	35 ± 10

Постройте модель системы, в которой можно оценить время ожидания кораблями каждого типа входа в порт. (Время ожидания входа в порт включает время ожидания освобождения причала и буксира.) Корабль, ожидающий освобождения причала, не обслуживается буксиром до тех пор, пока не будет предоставлен нужный причал. Корабль второго типа не займет буксир до тех пор, пока ему не будут доступны два буксира.

Построение модели

Эту задачу необходимо решать с помощью операторов МКУ.

Единицу модельного времени примем равной 1 минуте. Текст программы приведен ниже. Обратите внимание на особенности применения операторов, описывающих МКУ. При окончании буксировки судна второго типа освобождаются сразу два буксира, поэтому в программе соответствующий оператор имеет вид **LEAVE BUKS,2**.

Ниже приводится текст программы:

* Модель для задачи о судах и буксирах		
PRCH1	STORAGE	6 ; 6 причалов для кораблей первого типа
PRCH2	STORAGE	3 ; 3 причала для кораблей второго типа
BUKS	STORAGE	2 ; 2 буксира
;	SHIPS OF TYPE 1	; Сегмент для судов первого типа
	GENERATE	130,30 ; Подход к порту
	QUEUE	TYPE1 ; Начало ожидания
	ENTER	PRCH1 ; Получение причала
	ENTER	BUKS ; Получение буксира
	DEPART	TYPE1 ; При успехе - конец ожидания
	ADVANCE	30,7 ; Буксирование до причала
	LEAVE	BUKS ; Освобождение буксира
	ADVANCE	720,120 ; Погрузка-разгрузка
	ENTER	BUKS ; Получение буксира
	LEAVE	PRCH1 ; Освобождение причала
	ADVANCE	20,5 ; Буксирование (отчаливание)
	LEAVE	BUKS ; Освобождение буксира
	TERMINATE	; Конец истории транзакта
;	SHIPS OF TYPE 2	; Сегмент для судов второго типа
	GENERATE	390,60 ; Подход к порту судна второго типа
	QUEUE	TYPE2 ; Начало ожидания
	ENTER	PRCH2 ; Получение причала
	ENTER	BUKS,2 ; Получение двух буксиров
	DEPART	TYPE2 ; При успехе - конец ожидания
	ADVANCE	45,12 ; Буксирование до причала
	LEAVE	BUKS,2 ; Освобождение двух буксиров
	ADVANCE	1080,240 ; Погрузка-разгрузка
	ENTER	BUKS,2 ; Получение двух буксиров
	LEAVE	PRCH2 ; Освобождение причала
	ADVANCE	35,10 ; Буксирование (отчаливание)
	LEAVE	BUKS,2 ; Освобождение двух буксиров
	TERMINATE	0 ; Конец истории транзакта
*	TIMER	; Сегмент таймера
	GENERATE	48000 ; Моделирование 48000 минут работы
	TERMINATE	1 ; Конец моделирования
	* START	1 ; Команда заблокирована с целью отделить процессы трансляции и выполнения программы
		; При запуске программы на выполнение в параметре START задавать 1

Среднее время ожидания кораблями каждого типа входа в порт получаем в конце моделирования из стандартной статистики об очередях: оно равно показателю **AVERAGE TIME** соответствующей очереди (Эти же значения дают СЧА **QT\$TYPE1** и **QT\$TYPE2**.)

Необходимо провести моделирование на ПК, получить отчет **REPORT** и проанализировать его.

Задания для самостоятельной работы

1. Проведя расчеты по программе, ответьте на вопросы:

а) каково среднее время ожидания кораблями каждого типа входа в порт?

б) как изменится среднее время ожидания кораблями каждого типа входа в порт, если количество буксиров увеличить с двух до трех?

2. Выведите на экран график очереди судов второго типа. Существенна ли возникающая очередь?

3. Предположим, что возросла интенсивность движения судов второго типа: интервал времени прибытия уменьшился с 390 до 240 минут. Можно ли ликвидировать возникшую очередь, если увеличить количество причалов для кораблей второго типа с трех до четырех? С трех до пяти?

3. Модель кладовой

На фабрике в кладовой работает один кладовщик. Он выдает запасные части механикам, обслуживающим станки и устанавливающим эти части на испорченных станках. Запасные части довольно дорогие, и, кроме того, их ассортимент слишком велик для того, чтобы каждый механик мог иметь все запасные части в своем ящике. Время, необходимое для удовлетворения запроса, зависит от типа запасной части. Запросы бывают двух категорий. Соответствующие данные приведены в таблице.

Категория запроса	Интервал времени прихода механиков, с	Время обслуживания, с
1	420±360	300±90
2	360±240	100±30

Порядок обслуживания механиков кладовщиком такой: запросы первой категории обслуживаются только в том случае, когда в оче-

реди нет ни одного запроса второй категории. Внутри одной категории дисциплина обслуживания – «первым пришел – первым обслужился». Необходимо создать модель работы кладовой, моделирование выполнять в течение восьмичасового рабочего дня.

Построение модели

Есть два различных типа заявок, поступающих на обслуживание к одному устройству. Различаются распределения интервалов приходов и времени обслуживания для этих типов заявок. Трудность заключается в том, что один из типов имеет преимущество в обслуживании. Построить модель можно, используя тот же подход, что и в задаче 1. Но при этом необходимо использовать средство задания заявкам различных приоритетов.

Итак, запросы первой категории будем моделировать одним сегментом, а запросы второй категории – другим. Разные относительные приоритеты задаются путем использования для операнда **E** блока **GENERATE** запросов второй категории большего значения (задаем приоритет 2), чем для запросов первой категории (задаем приоритет 1). Элементы модели представлены в таблице:

Элементы GPSS	Интерпретация
<p>Транзакты</p> <p>В первом сегменте модели Во втором сегменте модели В третьем сегменте модели</p>	<p>Механики, делающие запрос первой категории Механики, делающие запрос второй категории Таймер</p>
<p>Устройство</p> <p>STOCKMAN</p>	<p>Кладовщик</p>
<p>Очереди</p> <p>QSTCKM1 QSTCKM2</p>	<p>Очереди, используемые для сбора статистики об ожидании механиков различных категорий</p>

Единица модельного времени – 1 с. Соответствующий текст программы приведен ниже.

```

*   MECHANICS OF TYPE 1
GENERATE 420,360,,,1 ;Приход механиков 1-й категории
QUEUE   QSTCKM1      ;Присоединение к очереди 1-й кат.
SEIZE   STOCKMAN     ;Занятие кладовщика
DEPART  QSTCKM1      ;Выход из очереди
ADVANCE 300,90       ;Обслуживание кладовщиком
RELEASE STOCKMAN     ;Освобождение кладовщика
TERMINATE 0          ;Уход из кладовой

```

* MECHANICS OF TYPE 2		
GENERATE	360,240,,,2	;Приход механиков 2-й категории
QUEUE	QSTCKM2	;Присоединение к очереди 2-й кат.
SEIZE	STOCKMAN	;Занятие кладовщика
DEPART	QSTCKM2	;Выход из очереди
ADVANCE	100,30	;Обслуживание кладовщиком
RELEASE	STOCKMAN	;Освобождение кладовщика
TERMINATE	0	;Уход из кладовой
* TIMER – Сегмент таймера		
GENERATE	28880	; Приход транзакта-таймера
TERMINATE	1	; Завершение прогона
* START	1	; Команда заблокирована с целью от- делить процессы трансляции и выпол- нения программы. ; При запуске программы на выполне- ние в параметре START задавать 1

Ввести программу, получить отчет **REPORT**, проанализировать его.

Провести расчеты и получить отчеты при значениях интервала прихода механиков для первой категории запроса 360 ± 120 с, 300 ± 220 с. Как изменилось среднее время ожидания в очереди по запросам первой категории?

Задания для самостоятельной работы

Слесарю поступают для обработки два типа деталей. В деталях первого типа необходимо нарезать резьбу в двух отверстиях, а затем закрепить деталь на узле. В деталях второго типа необходимо нарезать резьбу в четырех отверстиях, а затем закрепить деталь на узле.

Распределение интервалов прихода деталей первого типа 40 ± 10 мин, время на нарезку резьбы в двух отверстиях составляет 30 ± 12 мин, время на крепление детали на узле составляет 5 ± 1 минута.

Распределение интервалов прихода деталей второго типа 70 ± 15 мин, время на нарезку резьбы в четырех отверстиях составляет 45 ± 15 мин, время на крепление детали на узле составляет 7 ± 1 минута.

Составьте модель, обеспечив сбор данных об очереди. Моделирование провести для одной рабочей смены длительностью 480 минут.

4. Задача об обжиге деталей в печи

Изготовление деталей определенного вида включает длительный процесс сборки, который заканчивается коротким периодом обжига в печи. Поскольку эксплуатация печи обходится очень дорого, не-

сколько сборщиков используют одну печь, в которой одновременно можно обжигать только одну деталь. Сборщик не может начать новую сборку, пока не вытащит из печи предыдущую деталь.

Таким образом, сборщик работает в таком режиме:

- 1) собирает следующую деталь;
- 2) ожидает возможности использования печи по принципу FIFO («первым пришел – первым обслужился»);
- 3) использует печь;
- 4) возвращается к п. 1.

Время, необходимое на выполнение различных операций, приведено в таблице.

Операция	Необходимое время, мин
Сборка	30±5
Обжиг	8±2

Необходимо построить GPSS – модель описанного процесса в течение 40 часов модельного времени. Предполагается, что в течение рабочего дня нет перерывов, а рабочими днями являются все дни (без выходных).

Построение модели

В данном случае есть два ограничивающих условия, влияющих на пропускную способность системы: одна печь и некоторое фиксированное количество сборщиков, работающих в системе.

Для моделирования печи необходимо использовать понятие «устройство», а также отождествлять сборщиков с транзактами. Тогда можно считать, что сборщики «циркулируют» в системе, периодически осуществляя сборку и обжиг. Аналогично и транзакты должны циркулировать в GPSS-модели системы. Как видно из постановки задачи, модель представляет собой замкнутую систему массового обслуживания (СМО) с одним устройством обслуживания.

В реальной системе сборщик, после того как вынимает из печи обожженную деталь, возвращается и начинает новый этап сборки. В модели транзакт, после того как завершает использование устройства, должен быть перемещен с помощью блока **TRANSFER** в блок следующей сборки. Для ограничения общего количества транзактов, циркулирующих в модели, необходимо использовать операнд **D** блока **GENERATE**.

В качестве единицы модельного времени принимаем 1 минуту. Элементы модели представлены в таблице:

Элементы GPSS	Интерпретация
Транзакты В первом сегменте модели Во втором сегменте модели	Сборщики Таймер
Устройство OVEN	Печь
Очередь OCHER_OVEN	Очередь сборщиков у печи

Текст программы приведен ниже.

*** Задача о сборщиках и обжиге в печи**

```

GENERATE      , , , 4           ; Определение количества сборщиков
BACK1 ADVANCE  30, 12          ; Сборка следующей детали
          QUEUE  OCHER_OVEN     ; Присоединение к очереди
          SEIZE  OVEN           ; Занятие печи
          DEPART OCHER_OVEN     ; Выход из очереди
          ADVANCE 8, 2          ; Использование печи
          RELEASE OVEN         ; Освобождение печи
          TRANSFER , BACK1      ; Переход к сборке следующей детали
* TIMER - Сегмент таймера
GENERATE      2400             ; Моделирование пяти дней работы
TERMINATE     1               ; Завершение работы
* START      1               ; Запуск программы на выполнение

```

Для вычисления прибыли при заданном числе сборщиков необходимо знать, сколько готовых деталей они сделают на протяжении моделируемого периода. Это значение дает количество выходов из печи, т.е. в процессе моделирования нас интересует именно эта величина.

Найти величину прибыли при числе сборщиков, равном 4.

Задания для самостоятельной работы

1. Определите оптимальное число сборщиков, т.е. такое количество, которое дает наибольшую прибыль при моделировании в течение 40 часов модельного времени.
2. В связи с понижением напряжения сети время обжига возросло и стало равным 12 ± 2 минуты. Построить графики очереди к печи обжига при числе сборщиков, равном а) 3, б) 4, в) 5.

5. Переменные GPSS World

Общая характеристика переменных. При построении модели системы иногда возникает необходимость задать сложные математические или логические отношения атрибутов системы. Для этой цели в программе используются *переменные*.

В GPSS имеется три типа переменных:

- 1) арифметические переменные;
- 2) арифметические переменные с «плавающей точкой»;
- 3) булевы переменные.

Значение арифметических переменных может использоваться как:

1) операнд блока; в этом случае значение арифметической переменной может представлять собой:

- номер объекта (устройства, МКУ, очереди и т.п.);
- номер параметра транзакта;
- значение стандартного числового атрибута;

2) операнд **A** функции;

3) операнд **A** таблицы;

4) операнд выражения другой переменной.

В *выражениях* арифметические переменные используют такие арифметические операции:

+ алгебраическое сложение;

– алгебраическое вычитание;

алгебраическое умножение;

/ алгебраическое деление (результатом операции является целая часть частного);

^ возведение в степень;

\ деление без остатка (перед делением у обоих операндов отбрасываются дробные части, результат операции – целая часть частного).

В выражениях может быть задано любое число приведенных операций в различных комбинациях. Знак результата вычисляется по обычным алгебраическим правилам. Допускаются отрицательные значения переменных. Выражения анализируются слева направо. Возведение в степень, умножение, деление выполняются раньше, чем сложение и вычитание.

Вычисленное значение переменной является ее стандартным числовым атрибутом.

Арифметические переменные. Арифметические переменные аналогичны арифметическим выражениям в алгоритмических языках. Переменная задается оператором **VARIABLE**, называемым оператором описания переменной, который содержит арифметическое выражение. Формат оператора описания переменной приведен в табл. 1.

Таблица 1

Формат оператора описания переменной

Поле	Информация, задаваемая в поле
Метка	Имя (числовое или символьное) переменной
Операция	VARIABLE
Операнд А	Выражение, которое используется для вычисления значения переменной

При обращении к переменной используется обозначение **V<номер переменной>** или **V\$<имя переменной>**, т.е. **V** – это СЧА переменной.

Пример 1

Оператор описания **VARIABLE** определяет арифметическую переменную **RSL**:

RSL VARIABLE QT\$WAITL+3-FN\$DSTRB#P7

При любом обращении к переменной **RSL** (употребляется обозначение **V\$RSL**) ее значение вычисляется как текущая длина очереди **WAITL** (**QT\$WAITL** – СЧА регистратора очереди) плюс константа 3 и минус произведение значения функции **DSTRB** на значение параметра 7 транзакта, обрабатываемого в данный момент. В приведенном выражении **FN** – СЧА для обращения к функции, а **P** – СЧА транзакта.

Перед выполнением любой арифметической операции определяется значение каждого элемента и выделяется его целая часть. Постоянные без знака считаются положительными числами.

В выражении арифметической переменной могут быть использованы любые СЧА, функции и другие арифметические переменные. Запрещается использование самой вычисляемой переменной, а также переменных со знаком, так как знаки в данном случае рассматриваются как арифметические операции.

Система моделирования GPSS допускает использование скобок в выражениях арифметических переменных (для группировки членов или для обозначения операции умножения).

Пример 2

ADD VARIABLE P10+25

При обращении к арифметической переменной **ADD** ее значение вычисляется как сумма значений десятого параметра транзакта, обрабатываемого в данный момент, и константы 25.

Арифметические переменные с плавающей точкой аналогичны рассмотренным арифметическим переменным, за исключением того, что все операции над операндами выражений переменных с плавающей точкой выполняются без преобразования операндов и промежуточных результатов в целые значения. Лишь окончательный результат вычисления преобразуется в целое число.

Формат операторов описания арифметических переменных с плавающей точкой идентичен рассмотренному выше формату операндов описания арифметических переменных за исключением того, что в поле операции записывается слово **FVARIABLE**. Правила написания операторов те же, что и для арифметических переменных. Арифметическая переменная и переменная с плавающей точкой не могут иметь одинаковые номера.

Стандартный числовой атрибут **V\$<имя переменной>** используется для обращения к значениям как арифметических переменных, так и переменных с плавающей точкой.

Булевы переменные. Булевы переменные позволяют принимать решения в зависимости от значений СЧА и состояния объектов GPSS, используя для этого только одно выражение.

Булевы переменные – это логические выражения, состоящие из различных СЧА и (или) других булевых переменных. В булевой переменной проверяется одно или несколько логических условий. Результатом проверки является единица (*истина*), если условия выполняются, и ноль (*ложь*) – в противном случае.

При описании булевых переменных используются три типа операторов: логические, булевы и операторы отношений. Можно задать булеву переменную оператором **BVARIABLE**.

Логические операторы связаны с такими ресурсами, как устройства, МКУ и логические ключи. Они используются для определения состояния данных объектов.

Например, логический оператор **FNVj** принимает значение 1, если устройство **j** не занято, в противном случае – 0. Логический оператор **Uj** принимает значение 1, если устройство **j** используется, в противном случае – 0. Всего имеется 14 операторов, которые представлены в табл. 2.

Таблица 2

Логические операторы GPSS World

Логические операторы	Значение оператора, отражающее состояние ресурса
FVj или Fj	Равно 1, если устройство j занято или обслуживает прерывание, в противном случае – 0
FNVj	Равно 1, если устройство j не занято и не обслуживает прерывание, в противном случае – 0
Ij	Равно 1, если устройство j обслуживает прерывание, в противном случае – 0
NIj	Равно 1, если устройство j не обслуживает прерывание, иначе – 0
NUj	Равно 1, если устройство j не используется, в противном случае – 0
Uj	Равно 1, если устройство j используется, в противном случае – 0
SFj	Равно 1, если многоканальное устройство j заполнено, иначе – 0
SNFj	Равно 1, если МКУ j не заполнено, иначе – 0
SEj	Равно 1, если МКУ j пусто, иначе – 0
SNEj	Равно 1, если МКУ j не пусто, иначе – 0
SVj	Равно 1, если МКУ j находится в состоянии использования, в противном случае – 0
SNVj	Равно 1, если МКУ j не используется, в противном случае – 0
LRj	Равно 1, если логический ключ j выключен, иначе – 0
LSj	Равно 1, если логический ключ j включен, иначе – 0

Задания для самостоятельной работы

- Поясните выражения:
 - REMBO VARIABLE p6#86**
 - ALPHA VARIABLE QTSERR + 12**
 - BETA FVARIABLE QTSERR#0.24**
- Какие логические операторы, отражающие состояние многоканальных устройств, вам известны?
- Какие логические операторы, отражающие состояние занятости устройства, вам известны?

6. Функции в GPSS

В GPSS рассматриваются пять типов функций:

- 1) дискретная числовая (**D**),
- 2) непрерывная числовая (**C**),
- 3) табличная числовая (**L**),
- 4) дискретная атрибутивная (**E**),
- 5) табличная атрибутивная (**M**).

Рассмотрим два первых типа функций:

Дискретная функция представляет собой кусочно-постоянную функцию, которая состоит из горизонтальных ступеней (рис. 1). **Непрерывная функция** представляет собой кусочно-непрерывную функцию. Непрерывная функция в GPSS состоит из соединенных между собой прямых отрезков и представляет собой ломаную линию (рис. 2). Чтобы задать дискретную функцию, необходимы координаты крайних правых точек горизонтальных отрезков. Для непрерывной функции следует задать координаты всех точек, которые являются концами отрезков.

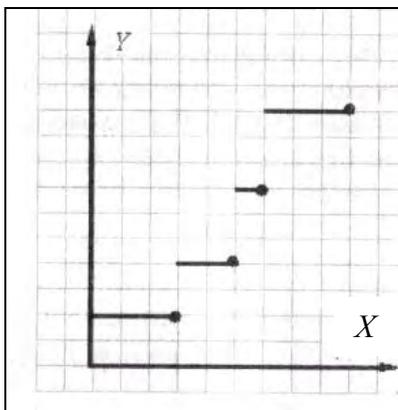


Рис. 1. Дискретная функция в GPSS

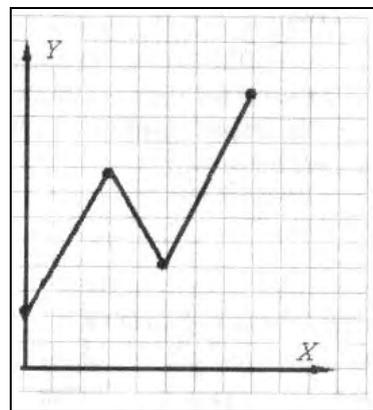


Рис. 2. Непрерывная функция в GPSS

Действия, необходимые для определения дискретной и непрерывной GPSS-функции:

1. Присвоить функции имя. Имя может быть числовым либо символьным.
2. Задать аргумент функции. Аргументом могут быть:
 - а) ссылка на генератор случайных чисел, используемый для розыгрыша в соответствии с распределением, заданным функцией;

- б) стандартный числовой атрибут;
- в) ссылка на любую другую функцию.

В случае а) аргумент задается в виде RN_j , j – целое число (номер генератора). В GPSS World количество генераторов случайных чисел неограничено, а выдаваемые ими значения лежат в интервале $[0; 0,999999]$.

3. Задать тип функции и число крайних точек функции.

4. Задать значения аргумента (переменной) и соответствующие значения функции (т.е. координаты крайних точек функции).

Три первых элемента информации указываются в *операторе определения функции*. Формат оператора представлен в таблице.

Поле	Информация, задаваемая в поле
Метка	Имя функции (числовое или символьное)
Операция	FUNCTION
Операнды	
A	RN_j ($j = 1, \dots, 7$) или СЧА
B	D_n либо C_n , где D определяет дискретную функцию, C определяет непрерывную функцию; n для дискретной функции – это число различных значений, получаемых функцией (количество горизонтальных отрезков, для непрерывной функции – это число, на единицу большее числа отрезков, составляющих функцию (количество точек)

За каждым оператором описания **FUNCTION** следуют операторы задания координат точек функции (значений аргументов X_i и соответствующих им значений функции Y_i) – *операторы описания координат функции*. Их формат:

$$X_1, Y_1/X_2, \dots /X_n, Y_n ,$$

где X_i, Y_i – координаты i -й точки функции (в случае моделирования случайной величины X_i является i -й суммарной (кумулятивной) частотой, Y_i – соответствующим значением случайной величины).

Значение функции является ее стандартным числовым атрибутом. Способ ссылки на этот атрибут зависит от того, как задано имя функции: в символьном или числовом виде. Если имя числовое, то к значению функции обращаемся через FN_j (где j – номер функции), если имя символьное – через $FNS\langle$ имя функции \rangle .

Аргументом функции может быть и значение какой-либо другой функции.

Каждая функция должна иметь, по крайней мере, две описанные точки.

Пример 1

Пусть необходимо смоделировать дискретную случайную переменную, заданную в таблице.

Значение случайной переменной	Относительная частота	Суммарная частота	Диапазон	Интервал
2	0,15	0,15	[0,0 – 0,15]	1
5	0,20	0,35	[0,15 – 0,35]	2
8	0,25	0,60	[0,35 – 0,60]	3
9	0,22	0,82	[0,60 – 0,82]	4
12	0,18	1,00	[0,82 – 1,0]	5

GPSS-функцию можно определить таким образом:

```
SERV FUNCTION RN4,D5  
.15,2/.35,5/.6,8/.82,9/1,12
```

В этом фрагменте программы **SERV** – имя функции, **RN3** – ссылка на генератор случайных чисел, **D5** означает, что функция дискретная, число значений равно 5. В следующей строке задаются значения функции и накопленные (кумулятивные) частоты появления этих значений.

Пример 2

Часто возникают ситуации, когда в процессе моделирования необходимо переходить в различные блоки программы в зависимости от логики работы модели.

Для вызова переключающей функции используется блок TRANSFER в режиме безусловного перехода.

Пример переключающей функции:

```
PEREKL FUNCTION RN4,D5  
0.2,LB1/0.4,LB2/0.6,LB3/0.8,LB4/1,LB5
```

	GENERATE	, „100
	TRANSFER	, FN\$PEREKL
LB1	QUEUE	STOR1
	TERMINATE	0
LB2	QUEUE	STOR2
	TERMINATE	0
LB3	QUEUE	STOR3
	TERMINATE	0
LB4	QUEUE	STOR4
	TERMINATE	0
LB5	QUEUE	STOR5
	TERMINATE	0
	GENERATE	1
	TERMINATE	1
	START	1

В этой программе **PEREKL** – имя функции, **D5** означает, что функция дискретная, число значений функции равно 5, во второй строке показаны значения этой функции (метки) и вероятности появления значений. Оператор **TRANSFER** вызывает значение функции по ее имени **FN\$PEREKL**. Далее происходит переход на метку – значение функции.

*Моделирование неравномерных случайных величин. Использование функций в блоках **GENERATE** и **ADVANCE**.* Пусть распределение интервалов поступления через определенный блок **GENERATE** или время задержки в некотором блоке **ADVANCE** *не является равномерным* (либо является равномерным с «плавающими во времени», т.е. нефиксированными значениями среднего и половины поля допуска). Для входов транзактов в модель через этот блок **GENERATE** и для задания закона времени задержки в соответствующем блоке **ADVANCE** необходимо использовать функции и (или) СЧА.

Использование функций, заданных в операндах блоков, зависит от контекста. От значения функции берется целая часть, за исключением тех случаев, когда это значение используется в качестве операнда **B** блоков **GENERATE** и **ADVANCE** или операнда **C** блока **ASSIGN**.

Пример 3

Пусть в моделируемой системе время обслуживания некоторым устройством распределено равномерно на интервале $A \pm 2$, где среднее время обслуживания A с вероятностью 0,4 принимает значение 5, а с вероятностью 0,6 – значение 7. Эту ситуацию можно смоделировать следующим образом.

Определим функцию **AVERAGET**:

**AVERAGET FUNCTION RN1,D2
.4,5/1,7**

Используем ее в блоке **ADVANCE**:

ADVANCE FNSAVERAGET,2

Выполнение подпрограммы блока **ADVANCE** включает расчет функции **AVERAGET**. Это, в свою очередь, требует обращения к генератору случайных чисел **RN1**. Пусть генератор выдал значение меньшее, чем 0,4. Тогда соответствующее значение функции

AVERAGET равно 5. Таким образом, время задержки текущего транзакта в устройстве будет равномерно распределено на интервале 5 ± 2 .

Моделирование пуассоновского потока. Рассмотрим табличный способ задания пуассоновского потока заявок. Пуассоновский входящий поток описывается таким образом: вероятность поступления k заявок пуассоновского потока в течение интервала t составляет

$$p_k(t) = \frac{e^{-\lambda t} (\lambda t)^k}{k!}, \quad (1)$$

где λ – интенсивность потока.

Пуассоновский поток заявок относится к случаю так называемых «редких событий», например к вероятности k телефонных звонков за единицу времени.

Интервалы времени между соседними заявками пуассоновского потока распределены по **экспоненциальному закону**. Можно получить ряд чисел, которые имеют экспоненциальное распределение, если ряд случайных чисел, равномерно распределенных на интервале $[0,1]$, преобразовать в соответствии с функцией, обратной к экспоненциальной функции распределения:

$$t_j = -\bar{T} \ln(r_j), \quad (2)$$

где t_j – j -й разыгранный интервал времени поступления; $\bar{T} = \frac{1}{\lambda}$ – средний интервал времени поступления; r_j – j -е число в последова-

тельности случайных чисел R с равномерным распределением на интервале $[0,1]$.

Разработчиками GPSS была осуществлена аппроксимация функции (2) с параметром $\lambda = 1$ с помощью 23 отрезков.

Эта функция **XPDIS** определяет экспоненциальное распределение с интенсивностью $\lambda = 1$:

XPDIS FUNCTION RN1,C24 ; exponential distribution function
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8

Пуассоновский входящий поток с интенсивностью λ , отличной от единицы, моделируется с помощью блока **GENERATE** таким образом:

1) в качестве операнда **A** используют среднее значение интервалов времени $T = \frac{1}{\lambda}$, где λ – интенсивность пуассоновского потока;

2) в качестве операнда **B** используют СЧА – значение функции **XPDIS**, операторы определения и описания которой приведены выше.

Пример 4

Пусть среднее значение интервалов поступления T в пуассоновском потоке требований равно 2 ч, а единица времени в модели равна 1 мин, тогда поступление заявок моделируется блоком:

GENERATE 120, FN\$XPDIS

Если необходимо моделировать задержку (обслуживание), распределенную по экспоненциальному закону со средним значением времени 345, то для этого используется блок:

ADVANCE 345, FN\$XPDIS

Свойство *ординарности* пуассоновского потока гласит: вероятность поступления двух или более заявок в течение малого временного интервала равна нулю.

Пусть пуассоновский поток моделируется блоком

GENERATE 5, FN\$XPDIS

Если в результате обращения к функции **XPDIS** полученное значение меньше, чем $1/5$, то целая часть произведения числа 5 и значения функции **XPDIS** равна нулю. Отсюда следует нарушение свойства ординарности. Во избежание этого рекомендуется, чтобы операнд **A** в блоке **GENERATE** был *больше 50*. Это легко достигается путем варьирования значения единицы модельного времени.

Моделирование нормального закона распределения. Функция стандартного нормального закона распределения с параметрами $m = 0$, $\sigma = 1$ задается в GPSS 24 отрезками следующим образом:

```
NOR FUNCTION RN1,C25
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2
.06681,-1.5/.11507,-1.2/.15866,-1/.21186,-.8/.27425,-.6
.34458,-.4/.42074,-.2/.5,0/.57926,.2/.65542,.4
.72575,.6/.78814,.8/.84134,1/.88493,1.2/.93319,1.5.97725,
2/.99379,2.5/.99865,3/.99997,4/1,5
```

Для того чтобы получить функцию нормального распределения случайной величины X с математическим ожиданием $m_x \neq 0$ и среднеквадратичным отклонением $\sigma_x \neq 1$, необходимо произвести вычисления по формуле:

$$X = m_x + \sigma_x Z,$$

где Z – случайная величина со стандартной нормальной функцией распределения.

Например, если случайная величина X имеет параметры $m_x \neq 60$ и $\sigma_x = 10$, то в GPSS эта случайная величина моделируется так:

```
NOR1 FVARIABLE 60+10#FN$NOR
```

Здесь **NOR1** – имя переменной, **FVARIABLE** означает, что это арифметическая переменная с плавающей точкой, **FN\$NOR** – обращение к функции **NOR**, которая определена выше.

Если необходимо осуществить задержку (обслуживание) по этому закону распределения, то используется блок

```
ADVANCE V$NOR1
```

Здесь **V\$NOR1** – арифметическая переменная. Напомним отличие арифметических переменных и переменных с плавающей точкой. Перед выполнением любой арифметической операции с арифметическими переменными определяется значение каждого элемента и выделяется его целая часть. Все операции над операндами выражений переменных с плавающей точкой выполняются без преобразования операндов и промежуточных значений в целые значения. Лишь окончательный результат вычисления преобразуется в целое число.

При использовании функции нормального распределения для блоков **GENERATE** и **ADVANCE** необходимо обеспечить неотрицательность значений интервалов поступления и задержки. Это можно сделать, если $m_x \geq 5\sigma_x$.

Задания для самостоятельной работы

1. Смоделируйте дискретную случайную переменную, заданную в таблице (предварительно заполнив пустые столбцы в этой таблице), с помощью дискретной функции **FEVR**.

Значение случайной переменной	Относительная частота	Суммарная частота	Диапазон	Интервал
16	0,08			1
20	0,24			2
32	0,30			3
120	0,10			4
1200	0,28			5

2. Пусть в моделируемой системе время обслуживания некоторым устройством распределено равномерно на интервале $A \pm 3$, где среднее время обслуживания A с вероятностью 0,25 принимает значение 5, с вероятностью 0,1 – значение 12, с вероятностью 0,25 – значение 15, с вероятностью 0,12 – значение 25, с вероятностью 0,33 – значение 100. Смоделируйте ситуацию, используя оператор **ADVANCE** и определив дискретную функцию.

3. Случайная величина X , распределенная по нормальному закону, имеет параметры $m_x = 250$ и $y_x = 30$. Запишите функцию, моделирующую эту случайную величину.

7. Стандартные числовые атрибуты, параметры транзактов, оператор цикла. Блоки ASSIGN, PRIORITY

В процессе моделирования интерпретатор автоматически регистрирует и корректирует информацию, касающуюся различных элементов, используемых в модели. Большая часть информации доступна только интерпретатору и используется для сбора статистической информации о работе модели. Однако к некоторым атрибутам (свойствам) объектов может обращаться и программист, управляя процессом моделирования в зависимости от их значений.

Рассмотрим несколько примеров зависимости функционирования элементов модели от системных атрибутов, т.е. СЧА.

1. Интенсивность работы некоторого устройства зависит от длины очереди. Для определения времени обслуживания при каждом поступлении транзакта на обслуживание необходимо знать значение такого системного атрибута, как длина очереди.

2. Интенсивность обслуживания некоторого устройства зависит от общей продолжительности его функционирования (проявление усталости – интенсивность со временем уменьшается, разогрев устройства – интенсивность со временем увеличивается). Время обслуживания – функция, которая зависит от времени, прошедшего с начала работы.

Условно атрибуты можно поделить на две категории:

- 1) атрибуты системы;
- 2) атрибуты транзактов.

Атрибуты системы – это параметры, которые описывают состояние объектов модели. Такие количественные показатели, как «текущая длина очереди» или «коэффициент загрузки устройства», являются типичными системными атрибутами. Стандартный набор атрибутов, подобных указанным, автоматически поддерживается интерпретатором GPSS.

Транзакты также могут иметь некоторые числовые характеристики (например **уровень приоритета**). Кроме того, транзакт снабжается некоторым числом **параметров**.

В языке GPSS атрибуты (свойства) объектов – это СЧА. Каждый объект GPSS имеет свой набор СЧА. Доступ к СЧА осуществляется

при использовании специальных обозначений этих атрибутов. Имя СЧА состоит из двух частей:

- 1) *групповое имя*, состоящее из одной или двух букв, идентифицирует тип объекта и тип информации о нем;
- 2) имя конкретного члена группы.

Объекты могут идентифицироваться с помощью числовых и символьных имен. При символьной идентификации объекта ссылка на его стандартный числовой атрибут записывается как **СЧА\$<имя объекта>** (в приведенных ссылках под «СЧА» понимается групповое имя).

Системные числовые атрибуты (СЧА). В табл. 1 показаны СЧА устройств, в табл. 2 – СЧА МКУ, в табл. 3 – СЧА очередей.

Таблица 1

СЧА устройств

Обозначение	Значение
F\$имя	Показатель занятости устройства (0 – если не занято, 1 – если занято)
FC\$имя	Число занятий устройства
FR\$имя	Нагрузка устройства, выраженная в долях тысячи
FT\$имя	Целая часть значения среднего времени задержки транзакта в устройстве
FV\$имя	Флаг готовности устройства к использованию (1 – готово, 0 – в противном случае)

Таблица 2

СЧА МКУ

Обозначение	Значение
R\$имя	Емкость незаполненной части МКУ
S\$имя	Емкость заполненной части МКУ
SC\$имя	Счетчик числа входов в МКУ (при каждом выполнении блока ENTER значение счетчика увеличивается на значение операнда B этого блока)
SM\$имя	Максимально занятая емкость МКУ. Запоминает максимальное значение S\$имя
SR\$имя	Нагрузка МКУ, выраженная в долях тысячи
SV\$ имя	Флаг готовности МКУj к использованию (1 – готово, 0 – не готово)

СЧА очередей

Обозначение	Значение
Q\$имя	Текущее значение длины очереди (текущее содержимое)
QA\$имя	Целая часть среднего значения длины очереди
QC\$имя	Число входов в очередь. При каждом входе в блок QUEUE очереди значение QC\$имя увеличивается на значение операнда B , при каждом входе в блок DEPART очереди значение QC\$имя уменьшается на значение операнда B
QM\$имя	Максимальное значение длины очереди (максимальное значение Q\$имя)
QT\$имя	Целая часть среднего времени пребывания в очереди всех транзактов, которые входили в очередь (включая и те, которые не ждали, – нулевые входы)

Мы привели лишь некоторые наиболее важные СЧА.

СЧА блоков и системные СЧА. Блоки имеют два стандартных числовых атрибута (их подсчет ведется автоматически):

W\$<метка блока> – *счетчик текущего содержимого блока* с меткой блока;

N\$<метка блока> – *счетчик входов*, т.е. общее число транзактов, вошедших в блок с момента последнего действия операторов **RESET** и **CLEAR** (от начала работы модели, если не было операторов **RESET** и **CLEAR**).

Например, **W\$BL1** – это число транзактов, которые находятся в блоке с меткой **BL1**; **N\$QP** – число транзактов, вошедших в блок с меткой **QP**.

Важным системным СЧА являются: **C1** – текущее значение относительного модельного времени, **M1** – время пребывания в модели транзакта, обрабатываемого интерпретатором в данный момент.

Пример 1

Использование СЧА в блоках **ENTER** и **ADVANCE**:

ENTER HS,RSSH

Войдя в блок **ENTER**, транзакт занимает **RSSH** каналов МКУ с именем **HS**.

ADVANCE FC\$PRIB

Задержка в этом блоке равна числу занятий устройства **PRIB**.

Параметры транзактов. Параметры транзактов – это свойства транзакта, определяемые пользователем. Множество параметров транзакта – набор стандартных числовых атрибутов, которые принадлежат транзакту. Параметры транзакта являются локальными переменными, которые доступны только данному транзакту.

В процессе перемещения транзакта по модели его параметры могут задаваться и модифицироваться в соответствии с логикой работы модели.

Особенности параметров транзактов:

1. Доступ к параметрам транзактов осуществляется таким образом:

P\$<имя>,

где **P** – СЧА транзакта, определяющий его групповое имя, т.е. имя всех параметров транзакта;

2. Номера (имена) конкретных членов множества параметров задаются с помощью целых чисел 1, 2, ... или символьных имен. Например, **P\$COLOR** – параметр с именем **COLOR**;

3. При входе транзакта в модель **начальное значение** всех его параметров устанавливается в **ноль**;

4. Значения параметров транзактов и их изменение определяет пользователь.

Пример 2

**SEIZE P\$PRIB
ADVANCE P\$TSERV
RELEASE P\$PRIB**

Транзакт занимает устройство, номер которого задан параметром транзакта **PRIB**. Время обслуживания задается параметром **TSERV**.

Изменение значений параметров. Блок ASSIGN (НАЗНАЧИТЬ). При входе транзакта в этот блок значения параметров могут задаваться или изменяться.

Формат блока:

ASSIGN A[+,-],B[,C]

Смысл параметров поясняется в таблице.

Операнд	Значение	Результат по умолчанию
A	Номер или имя модифицируемого или задаваемого параметра	Ошибка
B	Величина, используемая для модификации (число или СЧА)	Ошибка
C	Имя функции	Не используется

Блок **ASSIGN** может быть использован как в режиме замещения значения параметра (начальное значение всех параметров транзактов равно 0), так и в режиме увеличения и уменьшения. В режиме увеличения предшествующее значение параметра увеличивается на значение, стоящее в операнде **B**. В режиме уменьшения оно уменьшается на величину, стоящую в операнде **B**. Режимы увеличения и уменьшения определяются введением соответственно знаков «плюс» и «минус» перед запятой, которая разделяет операнды **A** и **B**.

При использовании операнда **C** значение операнда **B** умножается на значение функции, указанной в операнде **C**. Параметр, заданный в операнде **A**, изменяется на величину полученного произведения (в режиме увеличения и уменьшения) или приобретает значение результата (в режиме замещения).

Пример 3

Блок **ASSIGN** в режиме присваивания:

ASSIGN MEST,36

Параметру транзакта с именем **MEST** присваивается значение 36.

ASSIGN 3,25

Параметру **P3** присваивается значение 25.

Блок **ASSIGN** в режимах накопления и уменьшения:

ASSIGN 4+,Q5

Параметр транзакта 4 увеличивается на значение, равное текущей длине очереди 5.

ASSIGN P2 – , 7

От значения параметра, номер которого задан параметром **P2**, вычитается 7.

Организация циклов. Блок LOOP (ЦИКЛ). С помощью параметров транзактов в программе можно организовать циклы. Для этого используется блок **LOOP**. Он управляет количеством повторных прохождений транзактом определенной последовательности блоков модели.

Формат блока:

LOOP A[,B]

Операнд	Значение	Результат по умолчанию
A	Параметр транзакта, используемый для организации цикла (переменная цикла). Он может быть именем, положительным целым числом, СЧА	Ошибка
B	Метка (имя блока) начального блока цикла	Ошибка

Когда транзакт входит в блок **LOOP**, параметр, указанный в операнде **A**, уменьшается на единицу, а затем проверяется его значение на равенство нулю. Если значение не равно нулю, то транзакт переходит в блок, указанный в операнде **B**. Если значение параметра равно нулю, транзакт переходит в следующий блок.

Переменная блока **LOOP** может только уменьшаться.

Пример 4

```

        ASSIGN  1,3
SIS     SEIZE  PC
        ...
        RELEASE PC
        LOOP   1,SIS
        ...
    
```

Цикл организован по первому параметру транзакта. Его начальное значение равно 3. После освобождения устройства проверяется значение первого параметра. Если оно не равно нулю, то транзакт возвращается к блоку, помеченному меткой **SIS**, т.е. занимает устройство с именем **PC**. Всего каждый транзакт будет занимать это устройство три раза.

Блок PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ) присваивает или изменяет приоритет транзакта, если он был задан блоком **GENERATE** (по умолчанию приоритет транзакта равен нулю). Его формат:

PRIORITY A ,

где **A** – новый приоритет.

Новое значение приоритета может быть меньше, больше или равно текущему значению приоритета транзакта. Приоритет влияет на порядок выбора транзакта для обслуживания устройствами.

Стандартный числовой атрибут этого блока – **PR**. Поскольку уровень приоритета транзакта может изменяться от 0 до 127, то **PR** будет выдавать значение в диапазоне 0 – 127.

Пример 5

DELAY FUNCTION PR, D3
1,4/2,7/3,10
ADVANCE FN\$DELAY

Задержка в блоке **ADVANCE** зависит от приоритета транзакта. Транзакт с наиболее низким приоритетом (1) задерживается на 4 единицы модельного времени. Транзакт с наиболее высоким приоритетом (3) задерживается на 10 единиц модельного времени.

Задания для самостоятельной работы

1. Задан оператор:

ENTER 3,R3

Сколько каналов займет транзакт войдя в устройство МКУ с именем 3?

2. Задан оператор:

ENTER HS,RSSH

Используя определение СЧА многоканальных устройств, ответьте на вопрос: сколько каналов займет транзакт, войдя в МКУ?

3. Поясните, как работает оператор цикла:

	ASSIGN	5,6
ALPHA	SEISE	BETA
	...	
	RELEASE	PC
	LOOP	1, ALPHA
	...	

4. На сколько единиц модельного времени задерживается в блоке **ADVANCE** транзакт с приоритетом **PR =5**:

DELAY1 FUNCTION PR, D5

1,4/2,7/3,10/4,15/5,26

ADVANCE FN\$DELAY1

8. Сохраняемые величины. Блок **TEST**

В GPSS пользователю предоставляется возможность определить «свои» глобальные переменные, начальные значения которых могут быть заданы перед моделированием и к которым можно обратиться из любого места модели в любой момент времени. Эти переменные называют *сохраняемыми величинами (ячейками)*. Совокупность логически связанных между собой ячеек образует *матрицу* (аналог массива).

В отличие от параметров транзакта, приоритета и отметки времени, которые теряются в момент выхода транзакта из модели, ячейки доступны на протяжении всего процесса моделирования. Значения сохраняемых величин не подсчитываются интерпретатором автоматически (как СЧА устройств, очередей, МКУ и т.п.), а задаются и изменяются программистом.

Сохраняемые величины могут принимать положительные и отрицательные значения. Стандартный числовой атрибут **X_j** либо **X\$<имя ячейки>**.

Например, **X2** – значение ячейки 2, **X\$DAY** – значение ячейки **DAY**.

С матрицами связан стандартный числовой атрибут **MX_{j(m,n)}** – значение, записанное в строке **m** и столбце **n** матрицы **j** или **MX\$<имя матрицы>(m,n)**, если матрица имеет символьное имя.

Перед использованием матрица определяется оператором описания **MATRIX**. Начальные значения ячеек и матриц можно задать с помощью оператора описания **INITIAL**.

Оператор INITIAL (ИНИЦИАЛИЗИРОВАТЬ). Если в процессе моделирования происходит обращение к сохраняемой величине, которая не была задана, то интерпретатор выдает ошибку в процессе выполнения программы. Поэтому перед началом моделирования все сохраняемые величины должны быть инициализированы с помощью оператора **INITIAL**.

Пример 1

INITIAL X\$TIMER,1000000

Ячейке с именем **TIMER** присваивается начальное значение **1000000**.

INITIAL X3,25

Ячейке с номером **3** присваивается начальное значение **25**.

INITIAL MX\$COST(1,3),22

Величина **22** записывается в строку **1**, столбец **3** матрицы **COST**.

Блок TEST. Сравнение СЧА может быть выполнено с помощью блока **TEST (ПРОВЕРИТЬ)**. Его формат:

TEST X A,B[,C]

Смысл операндов приводится в таблице.

Операнд	Значение	Результат по умолчанию
A	СЧА	Ошибка
B	СЧА	Ошибка
C	Имя блока, в который переходит транзакт при условии, что ответ на вопрос, подразумеваемый оператором отношения, отрицательный	При отсутствии операнда C проверку выполняют в режиме отказа
X	Вспомогательный оператор, который представляет собой оператор отношения, использующийся при проверке	
	Значение оператора отношения:	Вопрос оператора отношения:
	G	A больше B?
	GE	A больше или равно B?
	E	A равно B?
	NE	A не равно B?
	LE	A меньше или равно B?
	L	A меньше B?

Операнды **A** и **B** – имена СЧА, которые сравниваются. Вспомогательный оператор **X** указывает способ сравнения этих двух СЧА друг с другом.

Пример 2

Режим отказа

TEST LE Q1,Q2

Проверяющий транзакт будет задержан в предыдущем блоке до тех пор, пока длина первой очереди **Q1** не станет меньше или равна длине второй очереди **Q2**.

Пример 3

Режим условного перехода

TEST LE Q1,Q2,ZHVS

Проверяющий транзакт перейдет в следующий по порядку блок, если содержимое первой очереди меньше или равно содержимому второй очереди. Если это условие не выполняется, транзакт перейдет в блок с меткой **ZHVS**.

Пример 4

Определение функции, значения которой зависят от текущего содержимого блока с именем **PPP**. Вид зависимости задан в таблице.

Текущее содержимое блока с именем PPP	0	1, 2 или 3	4 или 5	6	7 и больше
Значение функции	1	4	2	4	5

W\$PPP, D5

FFF FUNCTION W\$PPP,D5

0,1/3,4/5,2/6,4/7,5

Как мы указывали выше, обращение к счетчику текущего содержимого блока производится с помощью СЧА **W\$<метка блока>**, в данном случае – **W\$PPP**. Параметр **D5** означает, что функция дискретна, а число значений равно 5.

Пример 5

Определение функции, значения которой были бы вдвое больше текущей длины очереди **ALPHA** для значений 0, 1, 2, 3, 4. Для остальных значений содержимого очереди значение функции должно быть равно 10.

Решим эту задачу с помощью дискретной функции, определяемой шестью значениями. СЧА очереди, определяющий ее длину, определяется как **Q\$<имя очереди>**

**LONG FUNCTION Q\$ALPHA,D6
0,0/1,2/2,4/3,6/4,8/5,10**

Пример 6

В системе массового обслуживания (СМО) с одним устройством и очередью поступает пуассоновский поток заявок с интенсивностью 12 приходивших за 1 час. Обслуживание имеет экспоненциальное распределение, но среднее время обслуживания зависит от числа заявок, которые находятся в очереди к устройству. Эта зависимость приведена в таблице. Промоделировать обслуживание 500 заявок.

Длина очереди	0	1 или 2	3, 4 или 5	6 и больше
Среднее время обслуживания, мин	5,5	5,0	4,5	4,0

Для учета длины очереди при определении интенсивности обслуживания в модель необходимо включить дискретную функцию **MEAN**, в которой текущая длина очереди является аргументом. Эта функция используется для определения среднего значения интенсивности обслуживания.

Единица модельного времени – 1 секунда. Основные объекты программы поясняются в таблице.

Элементы GPSS	Интерпретация
Транзакты	Заявки
Устройство SURVR	Обслуживающее устройство
Функции: MEAN	Функция, определяющая среднее время обслуживания в зависимости от длины очереди.
XPDIS	Функция розыгрыша случайных чисел в соответствии с экспоненциальным законом со средним значением 1.
Очередь WAIT	Регистратор очереди для сбора статистики о состоянии очереди перед устройством

**MEAN FUNCTION Q\$WAIT,D4 ; Функция, построенная по данным
0,330/2,300/5,270/6,240 * таблицы 3**

**XPDIS FUNCTION RN1,C24 ; exponential distribution function
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8**

**GENERATE 300,FN\$XPDIS
QUEUE WAIT
SEIZE SURVR
DEPART WAIT
ADVANCE FN\$MEAN,FN\$XPDIS
RELEASE SURVR
TERMINATE 1
START 500**

В определении функции **MEAN** операнд **A** – СЧА очереди **WAIT**, а именно – ее длина **Q\$WAIT**. Операнд **B** показывает, что функция дискретная, принимает четыре значения. Таким образом, функция **MEAN** определяет среднее время обслуживания в зависимости от длины очереди.

Пуассоновский входящий поток с интенсивностью λ , отличной от единицы, моделируется с помощью блока **GENERATE**, в котором в качестве операнда **A** используется среднее значение интервалов времени $T = 1/\lambda$. В данном случае $A = 1 \text{ ч}/12 = 5 \text{ мин} = 300 \text{ секунд}$.

В блоке **ADVANCE** операнд **A** (среднее время задержки на обслуживание) задается значениями функции **FN\$MEAN**, а операнд **B** – значением функции **FN\$XPDIS**.

Пример 7

Устройство с экспоненциальным временем обслуживания имеет свойство уменьшать интенсивность своей работы на протяжении восьмичасового рабочего дня. В течение первых двух часов ему требуется в среднем 12 мин для выполнения обслуживания. В течение следующих двух часов среднее время обслуживания составляет 15 минут. В течение пятого, шестого и седьмого часа – 17 мин, в течение восьмого часа – 20 минут. Предполагая, что единица времени в модели равна 0,1 мин, определите функцию, значения которой давали бы среднее время, необходимое устройству для выполнения обслуживания. Покажите, как эту функцию нужно использовать в блоке **ADVANCE**. (СЧА **C1** – текущее значение относительного времени работы модели.)

Фрагмент программы, решающий данную задачу:

LAMDA FUNCTION C1,D4
1200,120/2400,150/4200,170/4800,200

XPDIS FUNCTION RN1,C24 ; exponential distribution function
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8

...

ADVANCE FNSLAMBDA,FNSXPDIS

...

Задания для самостоятельной работы

1. Поясните действие следующих операторов:

INITIAL XSTIMER,100
TEST E MX\$DELTA(2,9) , XSTIMER

2. Поясните действие следующих операторов:

TEST G Q5, Q6, МЕТКА1

...

МЕТКА1 TEST E F\$PRICHAL,0,МЕТКА02

3. В систему массового обслуживания (СМО) с одним устройством и очередью поступает пуассоновский поток заявок с интенсивностью 6 приходов за 1 час. Обслуживание имеет экспоненциальное распределение, но среднее время обслуживания зависит от числа заявок, которые находятся в очереди к устройству: при длине очереди 0 оно равно 2, при длине очереди 1 или 2 – равно 7, при очереди 3 и больше – равно 10. Про моделировать обслуживание 1000 заявок. Получить отчет и проанализировать его.

9. Моделирование замкнутых систем

Пример 1

Одноканальная замкнутая система. На монтажной площадке функционирует система «кран – машины». Известно среднее время погрузки машины краном – 2,1 минуты. Возможное отклонение от

этого времени – 1 мин, среднее время, необходимое для транспортирования груза и возвращения машины, составляет 20 мин, и отклонение от него – 3 минуты. Число машин, обслуживаемых краном, может варьироваться в пределах 6 – 9. Требуется определить основные характеристики системы при различном числе машин, обслуживаемых каналом обслуживания – краном:

- коэффициент использования канала обслуживания;
- среднее время пребывания требования (машины) в канале;
- максимальное содержимое (длину) очереди, т.е. число машин ожидающих погрузки (наибольшее из возможных);
- среднее содержимое (длину) очереди, т.е. среднее число машин, ожидающих обслуживания (погрузки);
- общее число входов в очередь, т.е. общее число поступлений машин на обслуживание в течение смены;
- среднее время пребывания требования в очереди.

Поступление машин на обслуживание – равномерное с интервалом 17 – 23 минут. Время обслуживания машины также равномерное с интервалом 3 – 5 минут.

Решение. Поскольку в этой задаче число машин, обслуживаемых краном, изменяется, то целесообразно использовать переменную в виде сохраняемой **X\$MASH**. Первоначальное значение **X\$MASH** определяется оператором **INITIAL**:

INITIAL X\$MASH,6

Особенности моделирования данной системы заключаются в следующем:

1. Оператор **GENERATE** используется только для формирования числа машин, которые обслуживает кран. Этот режим использования оператора **GENERATE** предполагает, что поля **A**, **B**, **C** должны оставаться пустыми, т.е. ставятся соответственно три запятые, затем в поле операнда **D** указывается число машин, которые должен обслуживать кран.

2. Машины после доставки груза в пункт назначения снова возвращаются в систему для погрузки. Возвращение машины в систему происходит при вхождении ее в оператор **TRANSFER**, который используется в режиме безусловной передачи.

3. Время обращения машины моделируется оператором **ADVANCE**. К нему направляются машины от оператора **TRANSFER ,AVTO**.

4. Возвращение машин в систему выполняется до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования основано на использовании простой модели измерения времени, состоящей из трех операторов:

GENERATE 480
TERMINATE 1
START 1

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 минут.

Ниже приведен текст программы **KRAN.GPS**, а на рис. 1 – окно **REPORT** с результатами моделирования системы «кран – машины».

*	GPSSW File	KRAN.GPS
*	Моделирование системы "Кран - машины" (по Е.М. Кудрявцеву)	
	INITIAL X\$MASH, 6	; Ввод числа машин
	GENERATE , , , X\$MASH	; Моделирование потока машин
AVTO	ADVANCE 20, 4	; Моделирование времени приезда машины к каналу обслуживания
	QUEUE POGR	; Машина встает в очередь POGR на погрузку
	SEIZE KRAN	; Занять канал обслуживания (кран)
	DEPART POGR	; Уменьшить содержимое очереди - машина начала обслуживаться краном
	ADVANCE 4, 0.7	; Моделирование времени погрузки машины
	RELEASE KRAN	; Кран освобожден
	TRANSFER , AVTO	; После погрузки краном машина направляется к месту разгрузки (переход на метку AVTO)
	GENERATE 480	
	TERMINATE 1	; Моделирование времени работы системы
	START 1	

GPSS World Simulation Report - Kran.1.1										
Friday, October 07, 2005 20:48:30										
START TIME		END TIME		BLOCKS	FACILITIES	STORAGES				
0.000		480.000		10	1	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
KRAN	107	0.885	3.970	1	1	0	0	0	0	
QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)		RETRY		
POGR	4	0	107	22	0.490	2.197		2.765 0		
SAVEVALUE	RETRY		VALUE							
MASH	0		6.000							

Рис. 1. Окно **REPORT** с результатами моделирования системы «кран – машины»

Пример 2

Многоканальная замкнутая система. Необходимо промоделировать работу многоканальной системы массового обслуживания, замкнутой, с экспоненциальным законом поступления требований на обслуживание (лифты – пассажиры, экскаваторы – автосамосвалы и т.п.), для которой справедливы следующие условия:

- поступление одного требования в систему на обслуживание не зависит от поступления другого (отсутствие последействия);
- в систему одновременно никогда не поступает два или более требований (поток ординарный);
- вероятность поступления требований зависит только от продолжительности периода наблюдений (поток требований стационарный), а не от принятого начала отсчета времени.

Известно среднее время поступления требования на обслуживание, равное 10 мин, которое подчиняется экспоненциальному распределению вероятностей. В системе массового обслуживания имеется три канала обслуживания (например, три экскаватора, нагружающих автосамосвалы). Требуется смоделировать процесс функционирования системы и определить следующие основные ее характеристики:

- коэффициент использования каждого канала обслуживания;
- среднее время использования каждого канала обслуживания;
- число входов в каждый канал обслуживания;
- среднее содержимое накопителя;
- среднее время пребывания требования в накопителе;
- максимальное содержимое накопителя;
- коэффициент использования накопителя.

Процесс функционирования трехканальной замкнутой системы массового обслуживания (СМО) представлен на рис. 2.

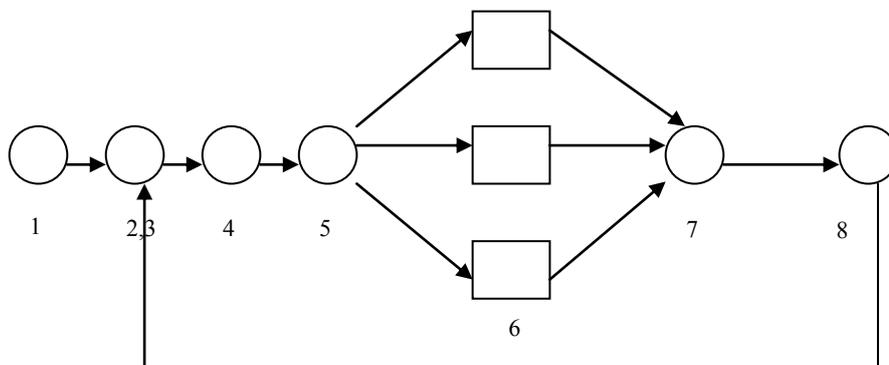


Рис. 2. Процесс функционирования трехканальной замкнутой СМО

Охарактеризуем каждое событие, возникшее в моделируемой системе:

1. Поступление требований в систему (**GENERATE**).
2. Вход требования в накопитель (**ENTER**).
3. Передача требования в один из свободных каналов обслуживания (**TRANSFER**).
4. Ожидание освобождения одного из каналов обслуживания (**SEIZE**).
5. Выход требования из накопителя (**LEAVE**).
6. Время обслуживания требования в канале обслуживания (**ADVANCE**).
7. Освобождение канала обслуживания (**RELEASE**).
8. Возвращение требования в систему (**TRANSFER**).

Программу работы СМО можно представить в виде трех секторов.

В первом секторе указывается вместимость СМО с помощью оператора **STORAGE**. Таким образом, емкость накопителя равна 3.

Во втором секторе будем моделировать поток требований в систему. Оператор **GENERATE** используем для формирования числа машин, которые обслуживает канал обслуживания. Этот режим использования оператора предполагает, что поля **A,B,C** остаются пустыми, т.е. ставятся соответственно три запятые, затем в поле операнда **D** указывается число машин: **GENERATE ,,10**.

Для задания экспоненциального распределения вероятностей можно использовать встроенную функцию **Exponential (A,B,C)**, где **A** – номер датчика случайных чисел, **B** – смещение экспоненты (нулевое), **C** – среднее значение, в данном случае – 10 минут.

Далее с помощью команды **ENTER NAK** требование входит в накопитель.

Поскольку СМО многоканальная, то необходимо использовать оператор **TRANSFER** для обеспечения возможности направления требования к незанятому каналу:

TRANSFER ALL,KAN1,KAN3,3

Сначала требование направляется к оператору, имеющему метку **KAN1**. Этим оператором является оператор **KAN1 SEIZE CAN1** (оператор ожидания освобождения одного из каналов обслуживания).

Если канал обслуживания с меткой **KAN1** занят, то требование направляется к следующему каналу, перешагивая через три оператора (3 – это число, заданное в поле операнда **D** в операторе **TRANSFER**).

Тогда следующим оператором будет: **SEIZE CAN1**. Если и он будет занят, то требование снова перешагнет через три оператора и т.д., пока не найдется незанятый канал обслуживания. В свободном канале обслуживания требование будет обслужено. Но предварительно требование должно запомнить канал, в который оно попало на обслуживание. Для этого используется оператор **ASSIGN** (присвоить). С его помощью в параметре «Требования» под номером 1 запоминается имя канала, в который требование пошло на обслуживание. В каждом канале имеется свой оператор **ASSIGN**. Например, для первого канала это присвоение будет выглядеть так: **ASSIGN 1,CAN1**.

Далее, после определения свободного канала и записи его имени с помощью оператора **TRANSFER ,COME** требование направляется на обслуживание. Однако перед началом обслуживания должно быть подано сообщение о том, что требование оставило накопитель под именем **NAK**, в котором оно находилось. Это выполняется с помощью оператора **COME LEAVE NAK**.

После выхода из накопителя требование поступает в канал на обслуживание. Это действие выполняется с помощью оператора **ADVANCE (Exponential (1, 0, 2))**.

После обслуживания требование выходит из канала обслуживания, и должен появиться сигнал о его освобождении. Это делается с помощью оператора **RELEASE P1**.

Параметр требования под номером **P1** содержит имя освобождаемого канала обслуживания. Далее требование входит в оператор **TRANSFER ,MASH** с безусловным переходом для возвращения в систему.

Таким образом, машины после обслуживания снова возвращаются в систему для обслуживания. Возвращение машин в систему выполняется до тех пор, пока время моделирования не превысит время моделирования системы. Определение времени моделирования системы выполняется в третьем секторе модели. Оно определяется с помощью простой модели измерения времени, состоящей из трех операторов:

GENERATE 480
TERMINATE 1
START 1

Этот сектор моделирует время работы системы в течение рабочей смены, равной 480 минут.

Текст программы приведен ниже.

; GPSSW File Trehkanalnaja_system.GPS		
* Моделирование трехканальной замкнутой *		
* СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ		
NAK	STORAGE	3
	GENERATE	, , , 10
MASH	ADVANCE	(Exponential(1, 0, 10))
	ENTER	NAK; NAK - имя накопителя
	TRANSFER	ALL, KAN1, KAN3, 3
KAN1	SEIZE	CAN1
	ASSIGN	1, CAN1
	TRANSFER	, COME
	SEIZE	CAN2
	ASSIGN	1, CAN2
	TRANSFER	, COME
KAN3	SEIZE	CAN3
	ASSIGN	1, CAN3
COME	LEAVE	NAK
	ADVANCE	(Exponential (1, 0, 2))
	RELEASE	P1
	TRANSFER	, MASH
	GENERATE	480
	TERMINATE	1
	START	1

Окно **SETTINGS** приведено на рис. 3, а фрагмент отчета показан на рис. 4.

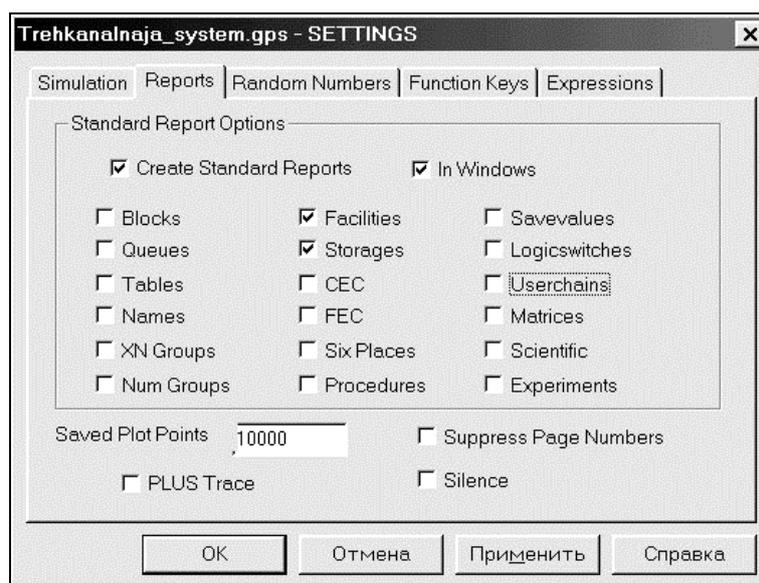


Рис. 3. Окно **SETTINGS** (вкладка **Reports**) с установками для имитационной модели многоканальной замкнутой СМО

GPSS World Simulation Report - Mnongokanalhaja_system										
Tuesday, October 11, 2005 11:01:35										
START TIME		END TIME		BLOCKS	FACILITIES	STORAGES				
0.000		480.000		18	3	1				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
CAN1	163	0.674	1.984	1	0	0	0	0	0	
CAN2	127	0.580	2.192	1	0	0	0	0	0	
CAN3	98	0.433	2.119	1	5	0	0	0	0	
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NAK	3	3	0	3	388	1	0.169	0.056	0	0

Рис. 4. Окно **REPORT** с результатами моделирования многоканальной замкнутой СМО

Как видно из отчета, коэффициент использования первого канала существенно больше коэффициентов использования второго и третьего каналов.

Задания для самостоятельной работы

1. В примере 2 (многоканальная замкнутая система) функцию экспоненциального распределения можно моделировать с помощью ломаных линий (функция **XPDIS**, как это показано в задании 8 данного практикума):

XPDIS FUNCTION RN1, C24
0,0/.1,.104/.2,.222 ...

Тогда обращение к этой функции будет выглядеть следующим образом:

ADVANCE 10, FN\$XPDIS

Введите эти операторы в программу и проведите расчеты. Выясните, насколько отличаются результаты от случая, когда использовалась встроенная функция **Exponential**.

2. Решите задачу из примера 2 (многоканальная замкнутая система) для случая двух каналов обслуживания, используя оператор **TRANSFER BOTH** (описан в разделе 5.9).

10. Модель производства стекла

Рассмотрим задачу о моделировании производства листового строительного стекла, являющуюся примером применения имитационного моделирования для анализа непрерывных производств.

Листы стекла движутся по конвейеру от стеклоплавильной печи к участку упаковки. Перед участком упаковки производится разбраковка листов стекла. Вероятность того, что лист идет в брак, равна 0,2, вероятность того, что лист первого сорта, равна 0,3, вероятность того, что лист второго сорта, равна 0,5.

На упаковочном участке работают три бригады. Первая бригада направляет бракованные листы в контейнер со стекольным боем, вторая и третья бригады упаковывают листы соответственно первого и второго сорта в бумагу и далее в деревянные ящики.

Необходимо составить программу, имитирующую работу данной системы, определить значения основных статистических показателей.

За единицу модельного времени выбираем 1 с; моделирование проводим за период одной рабочей смены (28800 с).

В качестве транзактов выбираем листы стекла, в качестве «обслуживающих устройств» – три бригады. В этой задаче мы имеем дело с транзактами трех типов. Направление движения транзакта к тому либо иному устройству зависит от его типа. Удобно в этих целях использовать так называемую переключающую функцию.

Дискретная функция **SORTST** позволяет с необходимой вероятностью выбирать метки **BRAK** (направление транзакта к устройству **BOY** – перевод листа стекла в брак), **FIRST** (упаковка листа в ящик для стекла первого сорта – устройство **UPAK1**), **SECOND** (упаковка листа в ящик для стекла второго сорта – устройство **UPAK2**). Имена очередей к соответствующим устройствам получаем добавлением буквы **Q** к имени устройства.

Приводим текст программы, реализующей описанный алгоритм:

```
*Моделирование производства листового строительного стекла
SORTST FUNCTION RN3,D3
0.20 ,BRAK/0.5 ,FIRST/1.0 ,SECOND
GENERATE 20,1 ; Может потом использовать нормальное
; распределение
TRANSFER ,FN$SORTST
```

```

BRAK      QUEUE BOYQ
          SEIZE BOY
          DEPART BOYQ
          ADVANCE 5,1
          RELEASE BOY
          TERMINATE 0
FIRST     QUEUE UPAK1Q
          SEIZE UPAK1
          DEPART UPAK1Q
          ADVANCE 10,1
          RELEASE UPAK1
          TERMINATE 0
SECOND    QUEUE UPAK2Q
          SEIZE UPAK2
          DEPART UPAK2Q
          ADVANCE 10,1
          RELEASE UPAK2
          TERMINATE 0
*****
          GENERATE 28800
          TERMINATE 1
          START 1

```

Задания для самостоятельной работы

1. Проведите моделирование в течение 8 часов, проанализируйте полученный отчет.

2. Разбраковка листов стекла осуществляется по трем сортам: вероятность того, что лист идет в брак, равна 0,2, вероятность того, что лист принадлежит к первому сорту, равна 0,6, вероятность того, что лист принадлежит ко второму сорту, равна 0,1, вероятность того, что лист принадлежит к третьему сорту, равна 0,1.

Составьте программу и проведите моделирование на восьмичасовой рабочий день, проанализируйте полученный отчет.

11. Модель производственной системы, выпускающей электронное оборудование

Рассмотрим задачу, в которой применены практически все изученные нами средства GPSS World.

Завод компании, производящей электронное оборудование, выпускает электронные часы. В упаковочном отделе часы укладываются автоматической упаковочной машиной в количестве, заказанном розничными продавцами. Размер заказа (случайная величина) определяется функцией, приведенной в табл. 1.

Частота поступления заказов различного размера

Частота	.10	.25	.30	.15	.12	.05	.03
Размер заказа	6	12	18	24	30	36	48

Время между поступлением заказов распределено экспоненциально со средним значением 15 минут. Время упаковки заказа составляет 120 с плюс 10 с на каждые часы, упакованные в заказе. Производственный цех выпускает электронные часы партиями по 60 штук через каждые 40 ± 5 минут.

Необходимо смоделировать 5 дней работы компании и получить отчет.

Текст соответствующей программы имеет следующий вид:

```

*Модель производственной системы, выпускающей электронное
*оборудование
*****
* 1-й сегмент - описание функций и МКУ
* Единица времени равна одному часу
* 1-й СЕГМЕНТ
Sizeorder FUNCTION RN1,D7 ; Размер заказа
.10,6/.35,12/.65,18/.80,24/.92,30/.97,36/1.0,48
Transit TABLE M1,.015,.015,20 ;Транзитное время
Number TABLE X1,100,100,20 ;Количество упакованных часов
; каждый день
Ptime VARIABLE .0028#P1+0.0334 ;Время упаковки
Amount EQU 1000 ;Начальное количество запасов
Stock STORAGE 4000 ;Склад вмещает 4000 часов
*****
* 2-й СЕГМЕНТ
GENERATE (Exponential(1,0,0.25)) ;Поступление заказа
ASSIGN 1,1,Sizeorder ;P1=размер заказа
TEST GE S$Stock,P1,Stockout ;Достаточно запасов?
LEAVE Stock,P1 ;Remove P1 from stock
QUEUE Packing
SEIZE Machine ;Машина занимается
DEPART Packing
ADVANCE V$Ptime ;Время упаковки
RELEASE Machine ;Машина освобождается
SAVEVALUE 1+,P1 ;Суммарное количество упакованных часов
TABULATE Transit ;Записывается транзитное время
TERMINATE
Stockout TERMINATE
*****
* 3-й СЕГМЕНТ
GENERATE 0.75,0.08334,1 ;Транзакт каждые 40+/-5 минут
ENTER Stock,60 ; Произведено 60 часов, склад
;пополняется
Stockad TERMINATE
*****

```

```

* 4-й СЕГМЕНТ
GENERATE      8          ; Транзакт каждый день
TABULATE      Number
SAVEVALUE     1,0        ; Записываем нуль в параметр 1
TERMINATE     1
*****
* 5-й СЕГМЕНТ - установка начального запаса на склад
GENERATE      ,,1,10     ; Транзакт, инициализирующий склад
ENTER         Stock,Amount ; Устанавливаем начальный запас
TERMINATE
*****

```

Программа состоит из пяти сегментов.

Пояснения к первому сегменту программы.

В первом сегменте определяются функции, переменные и памяти, т.е. сохраняемые ячейки.

M1 – СЧА, равный разнице между текущим значением таймера абсолютного времени и значением времени входа транзакта в модель. Сразу после входа транзакта в модель **M1=0**, через 10 единиц модельного времени **M1=10** и т.д.

Для накопления выборочных значений случайных величин и статистической обработки этих выборок используются GPSS-таблицы. Графическим аналогом GPSS-таблицы является гистограмма выборочных значений случайной величины, которую можно просмотреть в окне таблицы. Прежде чем использовать таблицу, ее нужно определить, а потом задать собираемые выборочные значения.

Оператор **TABLE** (таблица). Для определения таблицы надо задать имя таблицы, затем имя случайной переменной, значение которой будет табулироваться, далее число, являющееся первым граничным значением, затем ширину интервала, общую для всех интервалов таблицы, а также общее число интервалов таблицы, включая нижний и верхний.

Например, **Transit TABLE M1,.015,.015,20** означает, что имя таблицы – **Transit**, **M1** – имя случайной переменной. Первое граничное значение равно **.015**, ширина интервала равна **.015**, всего интервалов **20**.

X1 – содержимое ячейки 1. В конце расчета в этой ячейке будет количество часов, упакованных за каждый день. Таблица **Number** показывает гистограмму количества часов, упакованных за день.

Пояснения ко второму сегменту программы.

Транзакты в этом сегменте – заказы.

Оператором **GENERATE (Exponential(1,0,0.25))** имитируется поступление заказа.

ASSIGN 1,1,Sizeorder – первому параметру транзакта **P1** присваивается величина заказа **Sizeorder**. **P1** – параметр транзакта «заказ», имеющий смысл количества часов в заказе.

TEST GE S\$Stock,P1,Stockout

Этим оператором проверяется количество запасов на складе. **S\$имя** – емкость заполненной части МКУ, в данном случае – склада с именем **Stock**. Транзакт перейдет в следующий по порядку блок, т.е. на устройство упаковки **Packing**, если запас часов на складе **S\$Stock** будет больше или равен количеству часов в заказе **P1**. В противном случае переходим на метку **Stockout**.

Пояснения к третьему сегменту программы.

Транзакты в этом сегменте помещают на склад партии из 60 часов каждые 40 ± 5 минут.

Пояснения к четвертому сегменту программы.

Транзакты в этом сегменте появляются каждые 8 часов, т.е. один раз в день. Происходит табулирование суточных продаж и обнуление значения **P1** перед началом нового дня.

Пояснения к пятому сегменту программы.

Транзакт в этом сегменте инициализирует уровень запасов в 1000 единиц. В блоке **GENERATE „,1,10** означает, что транзакт один, приоритет транзакта задан равным 10, т.е. высоким. Таким образом, этот транзакт появляется первым.

Время измеряется в часах. Уровень запасов представляется величиной **S\$Stock**, текущим содержимым памяти **Stock**. Поэтому для операций со складом используются блоки **ENTER** (поместить на склад) и **LEAVE** (взять со склада). Невыполненные заказы определяются отсутствием изделий на складе. Склад может вместить 4000 изделий. Уровень запасов, представленный величиной **S\$Stock**, после начала процесса моделирования с помощью транзакта с высоким приоритетом устанавливается в последнем сегменте равным 1000 единиц. Этот транзакт и является самым первым транзактом, появляющимся в процессе моделирования.

Порядок выполнения программы. Чтобы запустить процесс моделирования и создать стандартный отчет, подайте команду **File/Open**, в диалоговом окне выберете файл **Manufact.gps**, в который был записан текст программы, нажать **Open**.

Затем выберете **Command/Create Simulation**. Если трансляция прошла успешно, то можно командой **Command/Start** запустить про-

грамму на выполнение. В диалоговом окне последней команды заменить 1 на 5 и нажать **ОК**. Команда относится к четвертому сегменту, поскольку именно в нем параметр при команде **TERMINATE** равен 1. Таким образом, моделирование продлится 5 дней, или 40 часов, при односменной работе.

После завершения процесса моделирования GPSS World выводит отчет в файл отчета.

Просмотрите отчет. Обратите внимание, что среднее значение количества заказов в отделе упаковки равно 0,115. Это указано в значении **Ave.Cont**, соответствующем очереди с именем **Packing**. Таким образом, фактически очередь в отдел упаковки не возникает, т.е. производство справляется с заданным потоком заказов.

Для просмотра гистограмм подать команды **Window/Simulation Window/Table Window**, далее выбрать имя таблицы, в которой представлены данные для гистограмм. Просмотрите гистограммы **Transit** и **Number**. Эти гистограммы показывают соответственно распределение транзитного времени заказов и количества часов, упакованных за день.

Рассмотрим графические окна. Выберите **Window/Simulation Window/Storages Window**. Это окно **Storages (Памяти)**. Уровень запасов на складе составляет 898 (столбец **Storage in Use**). Таким образом, текущий уровень запасов очень велик.

Выберите команду **Window/Simulation Window/Facilities Window**. Это окно показывает работу устройств. В этом окне видно, что коэффициент использования упаковочной машины был низким. Цвет значка говорит о том, что машина в данный момент используется.

Задания для самостоятельной работы

1. Выполните программу «Модель производственной системы, выпускающей электронное оборудование» (файл **Manufact.gps**). Проанализируйте полученные в отчете данные по очередям ко всем устройствам, имеющимся в модели. Ответьте на вопрос: достаточно ли производительности устройств для нормальной работы производства?

2. Введите в программу **Manufact.gps** новое распределение частоты поступления заказов различного размера (табл. 2).

Частота поступления заказов различного размера

Частота	.10	.25	.30	.15	.12	.05	.03
Размер заказа	6	12	25	30	45	50	70

Проведите расчеты, получите отчет. Справляется ли производство с новым увеличенным потоком заказов? В каких устройствах появляются очереди? Какие, по вашему мнению, следует ввести изменения в производство с тем, чтобы оно справлялось с новым увеличенным потоком заказов?

3. Выполните расчеты с использованием данных из табл. 1. Предположим, что производственный цех выпускает электронные часы партиями по 60 штук каждые 30 ± 5 минут. Будет ли в этом случае наблюдаться переполнение склада? (Указание: используйте средства GPSS World для построения графиков.)

4. Выполните расчеты с использованием данных из табл. 2. Предположим, что произведенный цех выпускает электронные часы партиями по 60 штук через каждые 30 ± 5 минут. Будет ли в этом случае наблюдаться переполнение склада? (Указание: используйте средства GPSS World для построения графиков.)

12. Метод статистических испытаний (метод Монте-Карло)

Вышеприведенные задачи имитационного моделирования, содержащие случайные факторы, решались путем однократного просчета. Такой подход годится лишь в качестве первого приближения к решению задачи.

Строгий подход к имитационному моделированию основывается на методе статистических испытаний (метод Монте-Карло). (Об этом методе мы уже рассказывали в разделе 3.1 данного учебного пособия.)

Суть метода состоит в том, что вместо описания случайных явлений аналитическими зависимостями проводится «розыгрыш» случайного явления с помощью некоторой программно реализуемой процедуры, которая дает случайный результат. С помощью такого «розыгрыша» получают одну реализацию случайного явления. Далее

такой «розыгрыш» проводится многократно, накапливается статистический материал. Этот материал обрабатывается теми же методами, что и материал, получаемый в результате реальных экспериментов. Метод Монте-Карло, как хорошо известно, показал свою эффективность при решении детерминированных задач. Поэтому переход к стохастическим задачам не вызывает серьезных принципиальных изменений в моделирующем алгоритме. Это важнейшее достоинство метода. Впрочем, у метода Монте-Карло, собственно, и нет каких-либо серьезных конкурентов.

Метод Монте-Карло реализуется путем многократного просчета значений выходных характеристик \bar{W} при постоянных значениях детерминированных факторов \bar{a} и различных значениях случайных факторов \bar{y} . Последние «разыгрываются» с помощью датчиков случайных чисел в соответствии с законами распределения. Таким образом, в моделирующем алгоритме образуется «цикл по \bar{y} ».

Изменения случайных факторов в GPSS World обеспечиваются сменой последовательностей случайных чисел, генерируемых программными датчиками случайных чисел.

Проиллюстрируем это на следующем примере (программа ANOVA1_1.GPS):

```
; GPSS World Sample File - ANOVA1_1.GPS
*****
* Моделирование парикмахерской
* Время в минутах
*****
GENERATE 5,1.7      ;Новый клиент
QUEUE Barber      ;Начало ожидания
SEIZE Barber      ;Захват или ожидание парикмахера
DEPART Barber     ;Конец ожидания
ADVANCE 5         ;Стрижка
RELEASE Barber    ;Освобождение парикмахера
TERMINATE 1       ;Уход клиента
*****
```

Программа ANOVA1_1.GPS имитирует работу парикмахерской с устройством обслуживания Barber. Случайность задается только в приходе новых клиентов (оператор GENERATE).

Введите программу, проведите расчеты, получите отчет. Этот отчет отражает только одну реализацию последовательности случайных чисел.

Для проведения трех прогонов (или трех реплик) имитации с разными последовательностями случайных чисел составим включаемый файл **ctl1.txt**, запишем его в ту же папку, что и файл **ANOVA1_1.GPS**. Этот текстовый файл выглядит следующим образом:

```
; Включаемый файл ctl1.txt  
RMULT 411 ; Новый генератор случайных чисел  
START 100 ; Первый прогон  
CLEAR Off ; Сброс статистики между прогонами  
RMULT 421 ; Новый генератор случайных чисел  
START 100 ; Второй прогон  
CLEAR Off ; Сброс статистики между прогонами  
RMULT 431 ; Новый генератор случайных чисел  
START 100 ; Третий прогон  
CLEAR Off ; Сброс статистики между прогонами  
; INCLUDE "ctl1.txt"
```

С помощью команды **RMULT <число>** можно запустить генератор случайных чисел с новыми начальными условиями. При этом **<число>** может быть в диапазоне от 1 до 9999. Оператор **CLEAR Off** полностью очищает значения всех переменных перед новым расчетом модели.

Как видим, теперь мы должны получить три прогона процесса имитации. Для этого загрузите программу **ANOVA1_1.GPS**. Откомпилируйте модель командами **Command, Create Simulation**. Получите журнал с результатами компиляции **JOURNAL**.

Теперь перейдем к расчетам. Подайте команду **Command, Custom**.

В открывшемся окне наберите (либо скопируйте из буфера обмена) **INCLUDE "ctl1.txt"**. Нажмите ОК. После этого процесс имитации выполнится трижды с разными случайными числами. Получаем три отчета.

Просмотрите эти отчеты. Вы заметите разницу в результатах. Например, в строке **QUEUE BARBER** среднее время ожидания в очереди (параметр **AVE.TIME**) составит 6.33; 2.74; 1.61. Как видим, реализации (прогоны) могут отличаться довольно заметно. Для того чтобы на основе применения метода Монте-Карло убедиться в статистической значимости результатов имитационного моделирования, необходимо определить доверительный интервал для выборочного среднего случайной величины. Считается, что если отношение этого интервала к выборочному среднему меньше 0,1, то число испытаний достаточно (см. задание для самостоятельной работы №2 к данному разделу).

Задания для самостоятельной работы

1. Введите в модель 15 прогонов, проведите расчеты, выпишите из отчетов в строке **QUEUE BARBER** значение параметра **AVE.TIME**, определите среднее значение этой величины.

2. Введите 15 значений параметра **AVE.TIME**, полученных в задании 1, в программу «Анализ данных» табличного процессора EXCEL. Расчет этого критерия производится автоматически (после загрузки EXCEL подать команды: «Сервис» – «Анализ данных» – «Описательная статистика»). Определите выборочное среднее значение величины **AVE.TIME**, среднее квадратичное отклонение, доверительный интервал выборочного среднего при «уровне надежности 95%». Является ли полученное выборочное среднее статистически значимым?

13. Применение дисперсионного анализа для оценки значимости результатов имитационного моделирования

Итак, мы убедились, что изменение только начальных значений генераторов случайных чисел в процессе моделирования может привести к значительной разнице в результате. Как при решении реальных проблем с помощью имитационной модели различать случайные и действительные результаты? С этой целью используется дисперсионный анализ. В системе GPSS World дисперсионный анализ выполняется специальной подпрограммой **ANOVA**.

Вернемся к задаче о парикмахерской (программа **ANOVA1_1.GPS**). Предположим, что время стрижки увеличено с 5 мин до 6,8 минут. Должно увеличиться и среднее время пребывания клиентов в очереди к парикмахеру. Расчеты подтверждают это. Но является ли этот вывод статистически значимым? Ведь в нашей задаче есть случайный фактор – промежуток времени между приходом в парикмахерскую клиентов.

Чтобы выяснить это, введем в программу **ANOVA1_1.GPS** в оператор **ADVANCE** вместо константы (была равна 5 мин) переменную **Cut_Time**. Проведем две серии расчетов: в первой серии этой переменной присвоим значение 6,8 мин, а во второй серии – 5 минут. Каждая серия состоит из трех имитаций с тремя разными последовательностями случайных чисел. Новую программу назовем **ANOVA_1.GPS**.

```

; GPSS World Sample File - ANOVA_1.GPS
*****
* Моделирование парикмахерской
* Время в минутах
* Время стрижки задается через переменную Cut_Time
*****
GENERATE 5,1.7 ;Новый клиент
QUEUE Barber ;Начало ожидания
SEIZE Barber ;Захват или ожидание парикмахера
DEPART Barber ;Конец ожидания
ADVANCE Cut_Time ;Стрижка
RELEASE Barber ;Освобождение парикмахера
TERMINATE 1 ;Уход клиента
; INCLUDE "CTLANOVA.TXT"
; SHOW ANOVA(Results,2,1)

```

Таким образом, будем исследовать значимость фактора – среднее время ожидания в очереди (AVE.TIME). Этот фактор имеет два уровня – 5 мин и 6,8 минуты. На каждом уровне сделаем три прогона процесса имитации (три реплики). Матрица результатов (назовем ее RESULTS) будет иметь размерность 2×3.

Составим включаемый файл CTLANOVA.TXT:

```

; Включаемый файл CTLANOVA.TXT
RESULTS MATRIX ,2,3 ; Матрица результатов - три реплики на двух
;уровнях
Cut_Time EQU 6.8 ; Присвоение значения времени стрижки
Treatment EQU 1 ; Присвоение номера уровня
RMULT 411 ; Новый генератор случайных чисел
START 100,NP ; Первый прогон процесса имитации
MSAVEVALUE RESULTS,1,1,QT$Barber ; Сохраняем результаты в
;матрице
CLEAR Off ; Сброс статистики между прогонами
RMULT 421 ; Новый генератор случайных чисел
START 100,NP ; Второй прогон процесса имитации
MSAVEVALUE RESULTS,1,2,QT$Barber ; Сохраняем результаты в
;матрице
CLEAR Off ; Сброс статистики между прогонами
RMULT 431 ; Новый генератор случайных чисел
START 100,NP ; Третий прогон процесса имитации
MSAVEVALUE RESULTS,1,3,QT$Barber ; Сохраняем результаты в
;матрице
CLEAR Off ; Сброс статистики между прогонами
Cut_Time EQU 5 ; Переходим ко 2-му уровню.
Treatment EQU 2 ; Присвоение номера уровня
RMULT 411
START 100,NP
MSAVEVALUE RESULTS,2,1,QT$Barber
CLEAR Off
RMULT 421
START 100,NP
MSAVEVALUE RESULTS,2,2,QT$Barber
CLEAR Off
RMULT 431
START 100,NP
MSAVEVALUE RESULTS,2,3,QT$Barber

```

В этом файле мы использовали стандартный числовой атрибут **QT\$Barber** – среднее время пребывания транзактов (клиентов) в очереди. Опция **NP** в блоке **START** используется, чтобы отключить создание стандартного отчета.

Загрузите программу **ANOVA_1.GPS**. Откомпилируйте ее командой **Command, Create Simulation**. Подайте команду **Command, Custom**. В открывшемся окне наберите (либо скопируйте из буфера обмена) **INCLUDE “CTLANOVA.TXT”**. Нажмите ОК. После этого процесс имитации выполнится шесть раз: на двух уровнях, на каждом – трижды с разными последовательностями случайных чисел. Однако мы подавили вывод отчетов, так что на экране этого не видно.

Теперь проведем дисперсионный анализ значимости результатов. Для этого подадим команду:

Command, Custom SHOW ANOVA(Results,2,1)

Найдите окно **Journal**. В нем вы увидите таблицу результатов дисперсионного анализа:

The screenshot shows the following data:

ANOVA						
Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)	
A	10844.768	1	10844.768	720.459	7.71	
Error	60.210	4	15.053			
Total	10904.978	5				

Treatment Level	Count	Mean	Minimum	Maximum	95% C.I. (SE)
A					
1	3	88.590	84.640	94.073	(82.372, 94.809)
2	3	3.562	1.609	6.337	(-2.656, 9.780)

3.8797650

Результаты дисперсионного анализа расшифрованы ниже:

Источник вариации	Сумма квадратов	Степени свободы	Средний квадрат	F	Критическое F (p=0.05)
А	10844.768	1	10844.768	720.459	7.71
Ошибка	60.210	4	15.053		
Всего	10904.978	5			

В колонке «Источник вариации» приведены источники: **A** – факторный источник, источник ошибок и суммарная вариация. Во второй колонке приведены соответствующие суммы квадратов, далее степень свободы, затем средние квадраты (суммы квадратов, деленные на степень свободы), значение фактора **F**, критическое значение фактора **F** на уровне 95% достоверности.

Объясним наши результаты. Основной вопрос: выше ли результат более быстрого парикмахера уровня случайных помех.

Посмотрим на фактор **A** в строке таблицы ANOVA окна **Journal** («Журнал»). Видно, что теоретически вычисленное критическое значение фактора **F** (**Critical Value of F**; этот фактор называют фактором Фишера) на уровне 95% достоверности равно 7,71. В нашем эксперименте значение получилось равным 720,459. Вычисленное нами значение намного выше: мы делаем заключение, что результат более быстрого парикмахера статистически значим.

Задания для самостоятельной работы

1. Проведите дисперсионный анализ (программа **ANOVA_1.GPS**, включаемый файл **CTLANOVA.TXT**) для случая, когда разброс интервалов прихода клиентов увеличен с 1.7 до 4.5, т.е. оператор генерации транзактов выглядит следующим образом: **GENERATE 5,4.5**. Каково значение фактора **F** в этом случае? Будет ли результат статистически значимым?

2. Проведите дисперсионный анализ (программа **ANOVA_1.GPS**, включаемый файл **CTLANOVA.TXT**) для случая, когда разброс интервалов прихода клиентов увеличен с 1.7 до 4.5, т.е. оператор генерации транзактов выглядит следующим образом: **GENERATE 5,4.5**. Кроме того, время стрижки 5 мин (параметр **Cut_Time**) следует увеличить с 5 мин до 6.3 минуты. Каково значение фактора **F** в этом случае? Будет ли результат статистически значимым?

3. Результат, полученный в предыдущем задании, может объясняться незначительным количеством прогонов имитационной модели. Измените включаемый файл **STLANOVA.TXT**, увеличив число прогонов до пяти. Повторите расчеты. Достигнута ли статистическая значимость результатов?

Список использованной литературы

1. *Емельянов А.А., Власова Е.А., Дума В.В.* Имитационное моделирование экономических процессов. М.: Маркет ДС, 2009.
2. *Кудрявцев Е.М.* GPSS World. Основы имитационного моделирования различных систем. М.: ДМК, 2004.
3. *Томашевский В., Жданова Е.* Имитационное моделирование в среде GPSS. М.: БЕСТСЕЛЛЕР, 2003.
4. *Рыжиков Ю.И.* Имитационное моделирование. Теория и технология. СПб.: Корона-принт, 2004.
5. *Бражник А.Н.* Имитационное моделирование: возможности GPSS World. СПб.: РЕНОМЕ, 2006.

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5
Тема 1. Основные понятия экономико-математического моделирования	8
1.1. Целевое назначение модели. Виды моделей	8
1.2. Критерии, ограничения, уравнения связи, решение модели	10
1.3. Принципы построения экономико-математических моделей	15
1.4. Процедура построения и реализации модели на ЭВМ	17
1.5. Цели построения экономико-математических моделей	20
Тема 2. Предпосылки создания имитационных моделей	22
2.1. Границы возможностей классических математических методов в экономике	22
2.2. Понятие имитационного моделирования. Особенности имитационных моделей	23
2.3. Классификация моделей имитационного моделирования	23
Тема 3. Математические основы имитационного моделирования	26
3.1. Метод Монте-Карло как математическая основа имитационного моделирования	26
3.2. Формула Поллачека-Хинчина	29
3.3. Формализованная схема процесса имитационного моделирования	30
3.4. Принципы построения имитационных моделирующих алгоритмов	32
3.5. Реализация имитационных моделей на ПК. Модели случайных входов	34
3.6. Основные типы распределений случайных величин, применяемые в имитационном моделировании	36
3.7. Моделирование случайных событий по экспериментальным данным	39
3.8. Нахождение вероятностных распределений по наблюдаемым данным	40
Тема 4. Методология имитационного моделирования	44
4.1. Имитационные проекты и планирование компьютерного эксперимента	44
4.2. Организация компьютерного эксперимента	48
4.3. Проблемы организации имитационных экспериментов	50
4.4. Имитация работы объекта экономики в трех измерениях. Имитация основных процессов	52

Тема 5. Пакеты программ имитационного моделирования экономических процессов	56
5.1. Сравнительный анализ популярных пакетов программ имитационного моделирования: GPSS World, AnyLogic, Pilgrim	56
5.2. Характеристика системы имитационного моделирования GPSS World	58
5.3. Главное меню GPSS World	60
5.4. Простейший пример применения GPSS World: моделирование работы магазина	62
5.5. Объекты системы GPSS World	77
5.6. Типы операторов	82
5.7. Основные блоки GPSS World	83
5.8. Элементы, отображающие одноканальные обслуживающие устройства	87
5.9. Переход транзакта в блок, отличный от последующего	93
5.10. Моделирование многоканальных устройств	96
5.11. Программа «БИЗНЕС-КОРПОРАЦИЯ»	98
Задачи по имитационному моделированию	102
1. Модель парикмахерской	102
2. Задача о морских судах и буксирах	104
3. Модель кладовой	106
4. Задача об обжиге деталей в печи	108
5. Переменные GPSS World	111
6. Функции GPSS	115
7. Стандартные числовые атрибуты, параметры транзактов, оператор цикла. Блоки ASSIGN, PRIORITY	123
8. Сохраняемые величины. Блок TEST	130
9. Моделирование замкнутых систем	135
10. Модель производства стекла	143
11. Модель производственной системы, выпускающей электронное оборудование	144
12. Методы статистических испытаний (метод Монте-Карло)	149
13. Применение дисперсионного анализа для оценки значимости результатов имитационного моделирования	152
Список использованной литературы	157

Учебное издание

Авторы-составители:

Бочаров Евгений Петрович

Алексенцева Ольга Николаевна

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ
ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ**

Учебное пособие

для студентов, обучающихся по направлению подготовки
080500.62 «Бизнес-информатика»

Редактор *Е.В. Смолякова*

Компьютерная вёрстка *Е.Н. Доронкиной*

Подписано в печать 25.11.2014 г. Формат 60×84 1/16.

Уч. -изд. л. 9,5. Усл. печ. л. 9,3.

Тираж . Заказ .

410003, г. Саратов, ул. Радищева, 89. ССЭИ РЭУ им. Г.В. Плеханова