

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ**

**Т.А. НИГМАТУЛИН, А.Ю. СИДОРЕНКО, М.А. АВДОШИНА**

**АНАЛИЗ ГЕОСИСТЕМ  
В СРЕДАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ**

**Часть 1. Среда GPSS**

**УЧЕБНОЕ ПОСОБИЕ**

**Санкт-Петербург**

**2025**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ

Т.А.НИГМАТУЛИН, А.Ю. СИДОРЕНКО, М.А. АВДОШИНА

АНАЛИЗ ГЕОСИСТЕМ  
В СРЕДАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ  
Часть 1. Среда GPSS

Учебное пособие

Санкт-Петербург

2025

УДК 681.3.06

ББК 32.973.26-018.2

Н60

*Допущено*

*Учебно-методическим советом Российского государственного гидрометеорологического университета в качестве учебного пособия для студентов, обучающихся по специальностям «Прикладные информационные системы и технологии», «Геопространственные цифровые двойники», «Системы мониторинга окружающей среды»*

Анализ геоинформационных систем в средах имитационного моделирования. Часть 1.

Среда GPSS: учебное пособие / Т.А. Нигматулин, А.Ю. Сидоренко, М.А. Авдошина – СПб. «Свое издательство», 2025. – 192с.

#### **Рецензенты**

Присяжнюк С.П., заслуженный деятель науки Российской Федерации, доктор технических наук, профессор, генеральный директор ЗАО «Институт телекоммуникаций».

Учебное пособие содержит краткие сведения о математическом аппарате моделирования систем и методах статистической обработки результатов компьютерного эксперимента. Рассмотрены принципы построения имитационных моделей в среде моделирования общего назначения GPSS World. Методы применения инструментальных средств среды моделирования поясняются на многочисленных примерах. По каждому разделу предусмотрены задания для самостоятельной разработки имитационных моделей систем.

Учебное пособие рассчитано на студентов и аспирантов Российского государственного гидрометеорологического университета, а также на всех интересующихся вопросами моделирования систем.

СПб. ООО «Свое издательство», 191040, Санкт-Петербург, Пушкинская ул, д. 10 литера А, помещ. 1-н

© РГГМУ

# ОГЛАВЛЕНИЕ

	3
ВВЕДЕНИЕ	6
ГЛАВА 1. МОДЕЛИРОВАНИЕ КАК ИНСТРУМЕНТ АНАЛИЗА СИСТЕМ	8
1.1. Аналитические методы моделирования	8
1.2. Математические модели систем	9
1.3. Имитационные методы моделирования	13
1.4. Принципы имитационного моделирования	16
1.4. Проблемы применения имитационного моделирования	18
1.5. Статистическое моделирование	22
ГЛАВА 2. СЛУЧАЙНЫЕ ВЕЛИЧИНЫ И СЛУЧАЙНЫЕ ПРОЦЕССЫ	24
2.1. Дискретные случайные величины	24
2.2. Числовые характеристики дискретных случайных величин	25
2.3. Распределение Пуассона	26
2.4. Непрерывные случайные величины и их характеристики	28
2.5. Закон равномерного распределения вероятностей	29
2.6. Закон нормального распределения вероятностей	30
2.7. Закон экспоненциального распределения	32
ГЛАВА 3. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ	35
3.1. Задачи математической статистики	35
3.2. Генеральная и выборочная совокупность статистических данных	36
3.3. Полигон и гистограмма	38
3.4. Оценка параметров генеральной совокупности	42
ГЛАВА 4. ОСНОВЫ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ	45
4.1. Системы массового обслуживания	45
4.2. Марковские процессы	47
4.3. Одноканальные устройства с отказами	52
4.4. Многоканальные СМО с отказами	54
4.5. Одноканальные системы с ожиданием	57
4.6. Многоканальная система с ожиданием	60
ГЛАВА 5. СРЕДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD	64
5.1. Концепция и возможности объектно-ориентированной моделирующей системы	64
5.2. Управление модельным временем	66
5.3. Объекты языка GPSS	67
5.4. Синтаксис элементов языка	72
5.5. Структура программы на языке GPSS	73
5.6. Организация поступления транзактов в модель	74
5.7. Параметры транзактов. Изменение приоритета	78
5.8. Удаление транзактов из модели и завершение моделирования	80

ГЛАВА 6. ЧИСЛОВЫЕ ДАННЫЕ И ОБЪЕКТЫ ВЫЧИСЛИТЕЛЬНОЙ КАТЕГОРИИ	85
6.1. Стандартные числовые атрибуты	85
6.2. Параметры транзактов. Изменение приоритета	86
6.3. Арифметические константы и переменные	89
6.4. Логические (булевы) переменные	92
6.5. Ячейки и матрицы ячеек	94
6.6. Функции	98
ГЛАВА 7. МОДЕЛИРОВАНИЕ СИСТЕМ	105
7.1. Задержка транзактов в блоках ADVANCE	105
7.2. Операции занятия и освобождения устройств	106
7.3. Операции блокирования и разблокирования одноканальных устройств	110
7.4. Моделирование многоканальных устройств	112
7.5. Операции блокирования и разблокирования многоканальных устройств	114
7.6. Статистика очередей	117
ГЛАВА 8. МАРШРУТИЗАЦИЯ ТРАНЗАКТОВ И ПОСТРОЕНИЕ ЛОГИЧЕСКОЙ СХЕМЫ РАБОТЫ МОДЕЛИ	120
8.1. Блок прямого перемещения	120
8.2. Блок проверки числовых значений	121
8.3. Блок пересылки транзактов	124
8.4. Организация циклов	128
ГЛАВА 9. СПИСКИ ПОЛЬЗОВАТЕЛЯ	131
9.1. Внесение транзакта в список пользователя	131
9.2. Вывод транзактов из списков пользователя	133
ГЛАВА 10. ПРОВЕРКА СОСТОЯНИЯ УСТРОЙСТВ И СИНХРОНИЗАЦИЯ ТРАНЗАКТОВ	140
10.1. Проверка ОКУ	140
10.2. Проверка МКУ	141
10.3. Синхронизация транзактов	143
ЗАКЛЮЧЕНИЕ	149
СПИСОК ЛИТЕРАТУРЫ	150
Приложение 1. Числовые атрибуты GPSS World	152
Приложение 2. Алфавитный указатель основных блоков GPSS World	157
Приложение 3. Предметный указатель основных блоков GPSS World	159
Приложение 4. Команды GPSS World	166
Приложение 5. Элементы стандартного отчета	168
Приложение 6. Индивидуальные задания	174
Приложение 7. Основные формулы теории массового обслуживания	186



## **ВВЕДЕНИЕ**

С точки зрения философии моделирование представляет собой один из методов познания мира. Под моделью следует понимать любое мысленное, формальное, физическое или какое-либо другое представление объекта окружающего мира, обеспечивающее изучение некоторых свойств данного объекта. Необходимо отметить, что в общем смысле модель является также объектом. Этот объект замещает объект-оригинал и создается с целью исследования объекта-оригинала. В свою очередь, моделирование - это процесс создания модели.

**Моделирование** – создание искусственного (мысленного, физического, цифрового) объекта, отображающего основные свойства оригинала, с целью фиксации или изучения свойств оригинала путем исследования свойств модели.

**Модель** – представление объекта, системы или понятия (идеи) в некоторой форме, отличной от формы их реального существования.

Польза от моделирования может быть достигнута только при соблюдении следующих достаточно очевидных условий:

- модель адекватно отображает свойства оригинала, существенных с точки зрения цели исследования;
- модель позволяет устранять проблемы, присущие проведению измерений на реальных объектах;
- затраты ресурсов на создание модели и анализа ее работы значительно меньше, чем затраты на исследование реальной системы.

Под **системой** понимают группу или совокупность объектов, объединенных какой-либо формой регулярного взаимодействия или взаимозависимости с целью выполнения определенной функции. Термин динамические системы применяется к системам, свойства которых изменяются с течением времени. В производственной деятельности с

помощью методов моделирования решаются в основном вопросы исследования функциональных характеристик систем, их взаимодействия между собой, а также прогнозирования результатов функционирования систем.

В настоящее время при анализе и синтезе больших систем получил развитие системный подход, который отличается от классического (индуктивного) подхода. Последний рассматривает систему путем перехода от частного к общему и синтезирует (конструирует) систему путем слияния ее компонент, разрабатываемых отдельно. Системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, причем исследуемый объект выделяется из окружающей среды. Важным для системного подхода является определение структуры системы – совокупности связей между элементами системы, отражающих их взаимодействие. Существуют структурные и функциональные подходы к исследованию структуры системы с ее свойствами. При структурном подходе выявляются состав выделенных элементов системы и связи между ними. При функциональном подходе рассматриваются алгоритмы поведения системы (функции - свойства, приводящие к достижению цели).

# **ГЛАВА 1. Моделирование как инструмент анализа систем**

## **1.1. Аналитические методы моделирования**

Необходимость изучения количественных и качественных изменений исследуемых систем обусловило разработку и применение мощного математического аппарата для целей моделирования. На основе математического аппарата был создан целый класс методов моделирования, называемых аналитическими. Аналитические методы позволяют получить характеристики системы как некоторые функции параметров ее функционирования. Таким образом, аналитическая модель представляет собой систему уравнений, при решении которой получают параметры, необходимые для оценки системы (время ответа, пропускную способность и т.д.). Использование аналитических методов дает достаточно точную оценку, которая, зачастую, хорошо соответствует действительности.

Смена состояний реальной системы происходит под воздействием множества как внешних, так и внутренних факторов, подавляющее большинство из которых носят стохастический характер. Вследствие этого, а также большой сложности большинства реальных систем, основным недостатком аналитических методов является то, что при выводе формул, на которых они основываются и которые используются для расчета интересующих параметров, необходимо принять определенные допущения. Тем не менее, нередко оказывается, что эти допущения вполне оправданы.

В тех случаях, когда требуется высокая степень детализации модели, учет множества факторов, в том числе случайных, это приводит к значительному усложнению математической модели. Аналитические решения, как правило, можно получить только для некоторых идеальных условий. В остальных случаях приходится довольствоваться только численными решениями, которые не могут, вообще говоря, описать общие

законы функционирования системы в широком диапазоне условий. К тому же анализ модели высокой сложности требует зачастую не меньших компьютерных ресурсов, чем имитационные модели.

Наконец, любая аналитическая модель трудно соотносится с реальным объектом.

## **1.2. Математические модели систем**

Важным этапом моделирования является создание математической модели исследуемой системы. На базе математической модели происходит анализ характеристик системы, при компьютерном моделировании на основе математической модели создается алгоритм программ для получения информации о поведении системы. Формальное описание объекта исследования необходимо также для взаимопонимания между специалистами разных областей, объединенных для решения какой-либо глобальной задачи.

При построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный, дискретно-детерминированный, дискретно-стохастический, непрерывно-стохастический, сетевой, обобщенный (или универсальный). Соответственно этим подходам были разработаны типовые математические схемы создания моделей.

1. Непрерывно-детерминированный подход использует в качестве математических моделей системы дифференциальных уравнений.

Созданные на основе этого подхода математические модели исследуются, как правило, аналитическими способами. Возможным приложением данного подхода является анализ систем автоматического управления непрерывными процессами, например, система поддержания стабильной температуры в помещении.

Задачей системы автоматического управления является изменение

выходных сигналов согласно заданному закону с определенной точностью (с допустимой ошибкой).

2. Дискретно-детерминированный подход реализуется с помощью математического аппарата теории автоматов. Система представляется в виде автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени. Математической моделью при этом подходе является конечный автомат, характеризующийся конечным множеством  $X$  входных сигналов, конечным множеством  $Y$  выходных сигналов, конечным множеством  $Z$  внутренних состояний, начальным состоянием  $Z_0$ ; функцией переходов  $g(z,x)$ ; функцией выходов  $v(z,x)$ . Автомат функционирует в дискретном автоматном времени, моментами которого являются такты (примыкающие друг к другу равные интервалы времени, каждому из которых соответствуют постоянные значения входного и выходного сигналов и внутренние состояния).

Подобный подход используется для описания таких объектов как элементы и узлы ЭВМ, устройства контроля регулирования, и управления, коммутационные устройства (например, телефонные станции).

3. Дискретно-стохастический подход использует в качестве математического аппарата вероятностные автоматы, которые можно определить как дискретные потактные преобразователи информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано статистически. Для такого автомата характерно задание таблицы вероятностей перехода автомата в некоторое состояние и появления некоторого выходного сигнала в зависимости от текущего состояния и входного сигнала.

Исследование автомата может проводиться как аналитическими, так и имитационными (например, методом статистического моделирования) методами. Этот подход применим для изучения эксплуатационных

характеристик производственных объектов (например, надежности, ремонтпригодности, отказоустойчивости и т.п.).

4. Непрерывно-стохастический подход применяется для формализации процессов обслуживания. Этот подход наиболее известен ввиду того, что большинство производственных (и не только производственных - экономических, технических и т.д.) систем по своей сути являются системами массового обслуживания. В обслуживании можно выделить две элементарные составляющие: ожидание обслуживания и собственно обслуживание, а в любой системе массового обслуживания можно выделить элементарный прибор. Соответственно в этом приборе выделяют накопитель заявок, ожидающих обслуживания (очередь); канал(ы) обслуживания; потоки событий (последовательность событий, происходящих одно за другим в какие-то случайные моменты времени); поток заявок на обслуживание, характеризующийся моментами времени поступления и атрибутами (признаками) заявок (например, приоритетами), и поток обслуживания, характеризующийся моментами начала и окончания обслуживания заявок.

Различают потоки однородных и неоднородных событий. Поток событий называется однородным, если он характеризуется только моментами поступления этих событий и задается последовательностью

$\{t_n\} = \{0 \leq t_1 \leq t_2 \dots \leq t_n \leq \dots\}$ , где  $t_n$  - момент наступления  $n$ -го события.

Потоком неоднородных событий называется последовательность событий, которые обладают индивидуальными признаками (приоритетом, принадлежностью к какому-либо источнику и т.д.)

Рассмотрим поток, в котором события разделены интервалами времени, которые являются случайными величинами. Пусть интервалы независимы между собой. Тогда поток событий называется потоком с

ограниченным последствием.

Поток событий называется ординарным, если вероятность того, что на малый интервал времени, примыкающий к моменту времени  $t$ , попадает больше одного события  $P_{>1}(t, t + \Delta t)$  пренебрежимо мала по сравнению с вероятностью того, что на этот же интервал времени попадает ровно одно событие  $P_1(t, t + \Delta t)$ . Для любого интервала верно следующее:

$$P_0(t, t + \Delta t) + P_1(t, t + \Delta t) + P_{>1}(t, t + \Delta t) = 1$$

Как сумма вероятностей событий, образующих полную группу и несовместных. Тогда для ординарного потока событий справедливо следующее:

$$P_0(t, t + \Delta t) + P_1(t, t + \Delta t) = 1, P_{>1}(t, t + \Delta t) = 0$$

Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени зависит лишь от длины этого интервала и не зависит от того, где на оси времени взят этот интервал.

Рассмотрим на оси времени ординарный поток событий и найдем среднее число событий, наступающих на интервале времени  $\Delta t$ :

$$0 \cdot P_0(t, t + \Delta t) + 1 \cdot P_1(t, t + \Delta t) = P_1(t, t + \Delta t)$$

Тогда среднее число событий ординарного потока в единицу времени (интенсивность потока):

$$\lambda = \lambda(t)$$

Для стационарного потока интенсивность постоянна.

Процесс функционирования устройства обслуживания можно представить как процесс изменения состояний его элементов во времени  $z(t)$ . Переход в новое состояние для устройства означает изменение количества заявок, которые в нем находятся. Если каналы различных устройств обслуживания соединены параллельно, то имеет место

многоканальное обслуживание (многоканальное Q-схема), а если устройства и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная Q-схема). В зависимости от емкости накопителя разделяют системы с потерями (емкость накопителя равна нулю), системы с ожиданием (емкость накопителя неограниченна) и системы смешанного типа (емкость накопителя ограничена).

Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов. Различают статические и динамические приоритеты. Статические приоритеты назначаются заранее и не зависят от состояний Q-схемы. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя на обслуживание каналом, можно выделить относительные и абсолютные приоритеты. Относительный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель, ожидает окончания обслуживания предшествующей заявки каналом и только после этого занимает канал. Абсолютный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель, прерывает обслуживание каналом заявки с более низким приоритетом и сама занимает канал.

При рассмотрении алгоритмов функционирования устройств обслуживания необходимо задать также набор правил, по которым заявки покидают накопители и каналы, например: блокировки по входу и выходу, маршрутизация заявок.

Аналитически модели систем массового обслуживания исследуются с помощью систем дифференциальных уравнений. Переменными в этих уравнениях являются вероятности переходов между состояниями

(состояния, в свою очередь, определяются количеством и местонахождением заявок в системе).

### **1.3. Имитационные методы моделирования**

С развитием вычислительной техники широкое применение получили имитационные методы моделирования для анализа систем, преобладающими в которых являются стохастические воздействия. Известный американский ученый Роберт Шеннон дает следующее определение: "Имитационное моделирование есть процесс конструирования модели реальной системы и постановки экспериментов на этой модели с целью либо понять поведение системы, либо оценить (в рамках ограничений, накладываемых некоторым критерием или совокупностью критериев) различные стратегии, обеспечивающие функционирование данной системы".

Все имитационные модели используют принцип черного ящика. Это означает, что они выдают выходной сигнал системы при поступлении в нее некоторого входного сигнала. Поэтому в отличие от аналитических моделей для получения необходимой информации или результатов необходимо осуществлять "прогон" имитационных моделей, т. е. подачу некоторой последовательности сигналов, объектов или данных на вход модели и фиксацию выходной информации, а не "решать" их. Происходит своего рода "выборка" состояний объекта моделирования (состояния - свойства системы в конкретные моменты времени) из пространства (множества) состояний (совокупность всех возможных значений состояний). Насколько репрезентативной окажется эта выборка, настолько результаты моделирования будут соответствовать действительности.

Этот вывод показывает важность статистических методов оценки результатов имитации. Таким образом, имитационные модели не формируют свое собственное решение в том виде, в каком это имеет место

в аналитических моделях, а могут лишь служить в качестве средства для анализа поведения системы в условиях, которые определяются экспериментатором.

1. Имитационное моделирование – один из универсальных мощных средств исследования сложных систем. В случаях, когда проведение натурального эксперимента невозможно, а математическая модель системы неразрешима за разумное время и с требуемой точностью или вообще не может быть создана на современном уровне развития математики, этот метод становится незаменимым для оценки эффективности и работоспособности системы, выявления «узких мест», поиска и исследования критических режимов работы. Преимущества метода имитационного моделирования особенно явны при решении экономических и производственных задач, связанных с традиционными в этих сферах рисками. При анализе той или иной сложной экономической системы компьютерный эксперимент безусловно экономичней и безопасней проведения «натурного» эксперимента. Это обуславливает востребованность данного метода в современном мире.

2. Исследование управленческих, организационных и экономических систем в подавляющем большинстве случаев выполняется с помощью методов теории массового обслуживания (ТМО). Любое производство, торговые и сервисные структуры (в том числе и компьютерные), процедуры контроля разнородных объектов (в частности, таможенного контроля) и даже бюрократические процедуры можно интерпретировать в виде последовательности одноканальных, многоканальных и многофазных систем массового обслуживания (СМО).

3. Имитационное моделирование – эффективный аппарат исследования стохастических систем, когда исследуемая система может быть подвержена влиянию многочисленных случайных факторов сложной

природы (у математических моделей для этого класса систем ограниченные возможности). Имеется возможность проводить исследование в условиях неопределенности, при неполных и неточных данных.

4. Имитационное моделирование является наиболее ценным, системообразующим звеном в системах поддержки принятия решений, т.к. позволяет исследовать большое число альтернатив (вариантов решений), проигрывать различные сценарии при любых входных данных. Главное преимущество имитационного моделирования состоит в том, что исследователь для проверки новых стратегий и принятия решений, при изучении возможных ситуаций, всегда может получить ответ на вопрос “Что будет, если? ...”. Имитационная модель позволяет прогнозировать, когда речь идет о проектируемой системе или исследуются процессы развития (т.е. в тех случаях, когда реальной системы не существует).

5. В имитационной модели может быть обеспечен различный (в том числе и очень высокий) уровень детализации моделируемых процессов. При этом модель создается поэтапно, постепенно, без существенных изменений, эволюционно.

#### **1.4. Принципы имитационного моделирования**

**Дискретно-событийный подход** (принцип “особых состояний”). В нем координаты времени меняются только когда изменяется состояние системы. Программа-планировщик намечает моменты времени наступления очередных событий в системе (поступление очередной заявки, начало-завершение обслуживания, включение-выключение устройств и пр.)

Принцип «продвижения модельного времени до ближайшего события» заключается в следующем. По всем процессам, параллельно протекающим в исследуемой системе, в каждый момент времени формируются моменты наступления «ближайшего события в будущем». Затем модельное время продвигается до момента наступления ближайшего

из всех возможных событий. В зависимости от того, какое событие оказалось ближайшим, выполняются те или иные действия. Если ближайшим событием является поступление заявки в систему, то выполняются действия, связанные с занятием устройства при условии, что он свободен, и занесение заявки в очередь, если устройство занято.

Если же ближайшим событием является завершение обслуживания заявки в устройстве, то выполняются действия, связанные с освобождением устройства и выбором на обслуживание новой заявки из очереди, если последняя не пуста. Затем формируется новый момент наступления этого же события.

Важно отметить, что единицы времени в модели не обязательно должны быть конкретными единицами времени, такими как секунда или час. Основной единицей времени в модели можно выбрать любую единицу, которая позволит получить необходимую точность моделирования. Важно помнить, единицы времени выбираются исходя из требований пользователя к точности моделирования. Какая бы единица ни была выбрана, например миллисекунда или одна десятая часа, она должна неизменно использоваться во всей модели.

На практике дискретно-событийный подход получил наибольшее распространение при анализе сложных систем. Он предпочтителен, когда события распределены неравномерно на временной оси и появляются через значительные временные интервалы.

### **Вероятностный (статистический, стохастический) подход.**

Имитационная модель – удобный аппарат для исследования стохастических систем. Стохастические системы – это такие системы, динамика которых зависит от случайных факторов. Входные, выходные переменные стохастической модели, как правило, описываются как случайные величины, функции, процессы, последовательности. Очевидно,

что однократное испытание вероятностной модели не даст полной картины процесса, так как результат одного испытания является также случайным.

Поэтому в моделировании применяется метод Монте-Карло, при котором производится серия прогонов вероятностной модели, а искомые величины обычно определяют как усредненные значения по данным большого числа реализаций процесса. Результаты моделирования обобщаются с помощью методов математической статистики. Ясно, что по закону больших чисел, чем больше число реализаций, тем больше получаемые оценки приобретают статистическую устойчивость.

**Структурный подход.** Программа модели отображает структуру реально существующей системы. Разработчик анализирует систему, выделяя в ней потоки заявок, одноканальные и многоканальные устройства обслуживания, структурные связи и маршруты движения потоков заявок. Эти реальные элементы имитируются в программе модели с помощью набора стандартных операторов и команд.

Имитационное моделирование дискретных систем со стохастическим характером функционирования, таких как системы и сети массового обслуживания, предполагает использование ряда типовых процедур, обеспечивающих реализацию соответствующих имитационных моделей. К таким процедурам, в первую очередь, относятся следующие:

- выработка (генерирование) случайных величин с заданным законом распределения;
- формирование потоков заявок, организация очередей заявок;
- имитация обслуживания в устройствах;
- организация службы времени, в том числе времени моделирования;
- сбор и статистическая обработка результатов.

#### **1.4. Проблемы применения имитационного моделирования**

Применение имитационного моделирования целесообразно при

наличии определенных условий. Эти условия определяет Р. Шеннон:

1. Не существует законченной математической постановки данной задачи, либо еще не разработаны аналитические методы решения сформулированной математической модели. К этой категории относятся многие модели массового обслуживания, связанные с рассмотрением очередей.

2. Аналитические методы имеются, но математические процедуры столь сложны и трудоемки, что имитационное моделирование дает более простой способ решения задачи.

3. Кроме оценки определенных параметров, желательно осуществить на имитационной модели наблюдение за ходом процесса в течение определенного периода.

Дополнительным преимуществом имитационного моделирования можно считать широчайшие возможности его применения в сфере образования и профессиональной подготовки. Разработка и использование имитационной модели позволяет экспериментатору видеть и "разыгрывать" на модели реальные процессы и ситуации.

Необходимо обозначить ряд проблем, возникающих в процессе моделирования систем. Исследователь должен акцентировать на них внимание и попытаться их разрешить, дабы избежать получения недостоверных сведений об изучаемой системе.

**Первая проблема**, которая касается и аналитических методов моделирования, состоит в нахождении "золотой середины" между упрощением и сложностью системы. По мнению Шеннона искусство моделирования в основном состоит в умении находить и отбрасывать факторы, не влияющие или незначительно влияющие на исследуемые характеристики системы. Нахождение этого "компромисса" во многом зависит от опыта, квалификации и интуиции исследователя. Если модель

слишком упрощена, если в ней не учтены некоторые существенные факторы, то высока вероятность получить по этой модели ошибочные данные. С другой стороны, если модель сложная и в нее включены факторы, имеющие незначительное влияние на изучаемую систему, то резко повышаются затраты на создание такой модели и возрастает риск ошибки в логической структуре модели.

Поэтому перед созданием модели необходимо проделать большой объем работы по анализу структуры системы и взаимосвязей между ее элементами, изучению совокупности входных воздействий, тщательной обработке имеющихся статистических данных об исследуемой системе.

**Вторая проблема** заключается в искусственном воспроизводстве случайных воздействий окружающей среды. Этот вопрос очень важен, так как большинство динамических производственных систем являются стохастическими, и при их моделировании необходимо качественное несмещенное воспроизведение случайности, в противном случае, результаты, полученные на модели, могут быть смещенными и не соответствовать действительности.

Существует два основных направления разрешения этой проблемы: аппаратная и программная (псевдослучайная) генерация случайных последовательностей. При аппаратном способе генерации случайные числа вырабатываются специальным устройством. В качестве физического эффекта, лежащего в основе таких генераторов чисел, чаще всего используются шумы в электронных и полупроводниковых приборах, явления распада радиоактивных элементов и т.д. Недостатками аппаратного способа получения случайных чисел является отсутствие возможности проверки (а значит гарантии) качества последовательности во время моделирования, а также невозможности получения одинаковых последовательностей случайных чисел.

Программный способ основан на формировании случайных чисел с помощью специальных алгоритмов. Этот способ наиболее распространен, так как не требует специальных устройств и дает возможность многократного воспроизведения одинаковых последовательностей. Его недостатками являются погрешность в моделировании распределений случайных чисел, вносимую по причине того, что ЭВМ оперирует с  $n$ -разрядными числами (т.е. дискретными), и периодичность последовательностей, возникающую в силу их алгоритмического получения. Таким образом, необходима разработка методов улучшения и критериев проверки качества генераторов псевдослучайных последовательностей.

**Третьей наиболее сложной проблемой** является проверка адекватности модели ее прототипу. Если прототип реально существует, основной способ – свести модель к известному, проверенному режиму и статистически сравнить результаты работы реальной системы и модели.

Адекватность моделей может быть оценена методом экспертных оценок, сравнением с другими моделями (уже подтвердившими свою достоверность), по полученным результатам. В свою очередь, для проверки полученных результатов часть из них сравнивается с уже имеющимися данными.

Один из способов обоснования адекватности разработанной модели – использование методов математической статистики. Суть этих методов заключается в проверке выдвинутой гипотезы (в данном случае - об адекватности модели) на основе некоторых статистических критериев.

Процедура оценки основана на сравнении измерений на реальной системе и результатов экспериментов на модели и может проводиться различными способами. Наиболее распространенные из них:

- по средним значениям откликов модели и системы;

- по дисперсиям отклонений откликов модели от среднего значения откликов системы;

- по максимальному значению относительных отклонений откликов модели от откликов системы.

Сложнее, если моделируется работа несуществующих (проектируемых) или ненаблюдаемых (развитие галактики) систем. В этом случае приходится довольствоваться сравнением результатов имитационных моделей с аналитическими или эмпирическими хотя бы в некоторых относительно простых случаях, в которых аналитическая модель имеет решение.

### **1.5. Статистическое моделирование**

Метод статистического моделирования на ЭВМ - основной метод получения результатов с помощью имитационных моделей стохастических систем, использующий в качестве теоретической базы предельные теоремы теории вероятностей.

Основой метода статистического моделирования является метод статистических испытаний Монте-Карло, который базируется на использовании случайных чисел, т. е. возможных значений некоторой случайной величины с заданным распределением вероятностей. Статистическое моделирование представляет собой метод получения с помощью ЭВМ статистических данных о процессах, происходящих в моделируемой системе.

Датой рождения метода Монте-Карло принято считать 1949 г., когда появилась статья под названием «The Monte Carlo method». Создателями этого метода считают американских математиков Дж. Неймана и С. Улама.

Теоретическая основа метода была известна давно. Более того, некоторые задачи статистики рассчитывались иногда с помощью случайных выборок, т.е. фактически методом Монте-Карло. Однако до появления

электронных вычислительных машин этот метод не мог найти сколько-нибудь широкого применения, ибо моделировать случайные величины вручную – очень трудоемкая работа. Таким образом, возникновение метода Монте-Карло как весьма универсального численного метода стало возможным только благодаря появлению ЭВМ.

Согласно методу Монте-Карло разработчик модели может моделировать работу множества сложных систем, испытывающих множество случайных воздействий и работающих по алгоритмам со множеством случайных параметров, и исследовать поведение систем, обрабатывая статистические данные, получаемые в результате моделирования. 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.

Суть данного метода состоит в том, что результат испытания зависит от значения некоторой случайной величины, распределенной по заданному закону. Поэтому результат каждого отдельного испытания также носит случайный характер. Проведя серию испытаний, получают множество частных значений наблюдаемой характеристики (выборку). Полученные статистические данные обрабатываются и представляются в виде численных оценок интересующих исследователя величин (характеристик системы).

Метод Монте-Карло имеет две особенности:

1) относительно простая структура вычислительного алгоритма;

2) погрешность вычислений, как правило, пропорциональна  $\frac{1}{\sqrt{N}}$ , где  $N$  – число испытаний. Отсюда видно, что для того, чтобы уменьшить погрешность в 10 раз (иначе говоря, чтобы получить в ответе еще один

верный десятичный знак), нужно увеличить  $N$  (т.е. объем работы) в 100 раз.

Добиться высокой точности таким путем невозможно. Поэтому обычно говорят, что метод Монте-Карло особенно эффективен при решении тех задач, в которых результат нужен с небольшой точностью (5-10%).

## ГЛАВА 2. Случайные величины и случайные процессы

### 2.1. Дискретные случайные величины

Дискретной называют случайную величину, значения которой могут принимать только некоторые заранее определённые значения. Например, сумма рейтинговых баллов за семестр, количество автопоездов, пересекающих таможенную границу за одну смену или число появления события при нескольких испытаниях. Число возможных значений дискретной случайной величины может быть конечным или бесконечным.

Если каждому конкретному значению дискретной случайной величины можно сопоставить вероятность ее появления, то она описывается законом распределения, который представляет собой перечень всех возможных значений случайной величины и соответствующих им вероятностей. Сумма всех вероятностей  $\sum p_i = 1$ . Такой способ задания закона распределения называется табличным. Закон распределения также может быть задан аналитически (формулой) и графически (многоугольником распределения, соединяющим точки  $(x_i; p_i)$ )

При табличном задании закона распределения дискретной случайной величины первая строка таблицы содержит возможные значения, а вторая – их вероятности:

$X_i$	$x_1$	$x_2$	...	$x_n$
$P_i$	$p_1$	$p_2$	...	$p_n$

Функция распределения случайной величины - это вероятность того, что случайная величина примет значение меньшее, чем конкретное числовое значение  $x$ :

$$F(X) = P(x < X).$$

Для дискретной случайной величины функция распределения

вычисляется для каждого значения как сумма вероятностей, соответствующих всем предшествующим значениям случайной величины.

## 2.2. Числовые характеристики дискретных случайных величин

Математическое ожидание дискретной случайной величины есть сумма произведений всех её возможных значений на их вероятности:

$$M(X) = x_1p_1 + x_2p_2 + \dots + x_n p_n = \sum_{i=1}^n x_i p_i \quad (2.1)$$

Свойства математического ожидания:

1) Математическое ожидание постоянной величины равно самой величине:

$$M(C) = C$$

2) Постоянный множитель можно выносить за знак математического ожидания:

$$M(CX) = C \cdot M(X)$$

3) Математическое ожидание суммы случайных величин равно сумме математических ожиданий слагаемых:

$$M(X_1 + X_2 + \dots + X_m) = M(X_1) + M(X_2) + \dots + M(X_m)$$

4) Математическое ожидание произведения взаимно независимых случайных величин равно произведению математических ожиданий сомножителей:

$$M(X_1 \cdot X_2 \cdot \dots \cdot X_m) = M(X_1) \cdot M(X_2) \cdot \dots \cdot M(X_m)$$

Дисперсия дискретной случайной величины есть математическое ожидание квадрата отклонения случайной величины от её математического ожидания:

$$D(X) = (x_1 - M(X))^2 p_1 + (x_2 - M(X))^2 p_2 + \dots + (x_n - M(X))^2 p_n \quad (2.2)$$

Свойства дисперсии:

1) Дисперсия постоянной величины равна нулю:  $D(C) = 0$

2) Постоянный множитель можно выносить за знак дисперсии,

предварительно возведя его в квадрат:  $D(CX) = C^2 \cdot D(X)$

3) Дисперсия суммы (разности) независимых случайных величин равна сумме дисперсий слагаемых:

$$D(X_1 \pm X_2 \pm \dots \pm X_m) = D(X_1) + D(X_2) + \dots + D(X_m)$$

Среднее квадратичное отклонение дискретной случайной величины, оно же стандартное отклонение, есть корень квадратный из дисперсии:

$$\sigma(X) = \sqrt{D(X)} \quad (2.3)$$

Мода дискретной случайной величины  $M_0(X)$  – это значение случайной величины, имеющее наибольшую вероятность. На многоугольнике распределения мода - это абсцисса самой высокой точки. Бывает, что распределение имеет не одну моду.

### 2.3. Распределение Пуассона

Пусть имеется некоторая последовательность событий, наступающих в случайные моменты времени (будем называть это потоком событий).

Среднее число событий, появляющихся в единицу времени (интенсивность потока) равна  $\lambda$ . Пусть этот поток событий - простейший (пуассоновский), т.е. обладает тремя свойствами:

1) вероятность появления  $k$  событий за определённый промежуток времени зависит только от длины этого промежутка, но не от точки отсчёта, другими словами, интенсивность потока есть постоянная величина (свойство стационарности);

2) вероятность появления  $k$  событий в любом промежутке времени не зависит от того, появлялись события в прошлом или нет (свойство «отсутствия последствия»);

3) появление более одного события за малый промежуток времени практически невозможно (свойство ординарности).

Вероятность того, что за промежуток времени  $\tau = (t_2 - t_1)$  событие произойдёт  $k$  раз, равна:

$$P(k)_\tau = \frac{\lambda\tau}{k!} e^{-\lambda\tau} \quad (2.4)$$

### Пример 2.1.

Среднее число запросов к серверу базы данных за 1 секунду равно двум. Найти вероятность того, что за 4 сек. поступит: а) три запроса; б) менее трёх запросов; в) не менее трёх запросов. Поток вызовов - простейший.

Используем формулу Пуассона.  $\lambda = 2$ ,  $\tau = 4$ . Сначала рассчитаем вероятности появления 0, 1, 2, 3 запросов за означенный интервал времени:

$$P(0) = 8^0/0! \cdot e^{-8} = e^{-8} \approx 0,000335$$

$$P(1) = 8^1/1! \cdot e^{-8} = 8 e^{-8} \approx 0,002684$$

$$P(2) = 8^2/2! \cdot e^{-8} = 32 e^{-8} \approx 0,010735$$

$$P(3) = 8^3/3! \cdot e^{-8} = 85,33 e^{-8} \approx 0,014313$$

Теперь, пользуясь теоремой сложения, определим вероятности событий:

$$а) P(k=3) = 0,014313$$

$$б) P(k<3) = P(0) + P(1) + P(2) = 0,013754$$

$$в) P(k \geq 3) = 1 - P(k < 3) = 1 - 0,013754 = 0,986246$$

*В системе GPSS распределение Пуассона имитируется библиотечной функцией POISSON (N,A), где N – номер генератора случайных чисел, A – среднее значение распределения  $A = \lambda * \tau = \lambda * (t_2 - t_1)$ . В результате вычисления этой функции получится случайное целое число, означающее конкретное количество событий в интервале времени  $\tau = (t_2 - t_1)$ . С помощью этой функции нельзя имитировать пуассоновский поток событий, т.к. в генераторе транзактов планируется не количество событий, а случайный интервал времени между ними, который является непрерывной случайной величиной и имитируется экспоненциальной функцией распределения.*

## 2.4. Непрерывные случайные величины и их характеристики

Случайную величину  $X$  будем называть непрерывной, если ее интегральная функция распределения  $F(X) = P\{x < X\}$  непрерывна и дифференцируема, за исключением, быть может, конечного числа точек.

Дифференциальной функцией распределения  $f(x)$  называют производную от интегральной функции

$$f(x) = F'(x) \quad (2.5)$$

Наряду с термином «дифференциальная функция» используют также термины – «плотность вероятности» или «закон распределения».

Свойства дифференциальной функции:

$$f(x) \geq 0;$$

$$P\{a < x < b\} = \int_a^b f(x) dx$$

$$\int_{-\infty}^{+\infty} f(x) dx = 1$$

$$F(x) = \int_{-\infty}^x f(x) dx$$

Математическим ожиданием непрерывной случайной величины называют число

$$M(X) = \int_{-\infty}^{+\infty} xf(x) dx$$

(2.6)

где  $f(x)$  – плотность вероятности, и предполагается, что интеграл сходится абсолютно.

Модой непрерывной случайной величины называется значение случайной величины, при котором плотность вероятности максимальна.

Дисперсия непрерывной случайной величины рассчитывается по формуле:

$$D(X) = \int_{-\infty}^{+\infty} (x - M(X))^2 dx$$

(2.7)

## 2.5. Закон равномерного распределения вероятностей

Распределение вероятностей называется равномерным, если на интервале, которому принадлежат все возможные значения случайной величины, дифференциальная функция имеет постоянное значение (рис. 2.1), а интегральная имеет вид, показанный на рис. 2.2.

$$f(x) = \begin{cases} 0, & x \leq a; \\ \frac{1}{b-a}, & a < x < b; \\ 0, & x \geq b \end{cases}$$

(2.8)

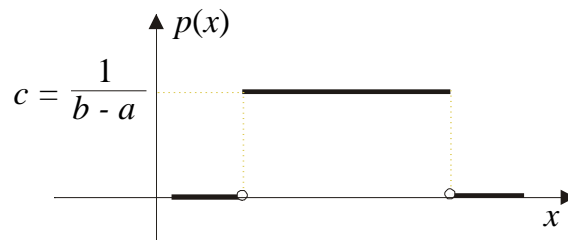


Рис. 2.1. Дифференциальная функция равномерного распределения

$$F(x) = \begin{cases} 0, & x \leq a; \\ \frac{x-a}{b-a}, & a < x < b; \\ 1, & x \geq b \end{cases}$$

(2.9)

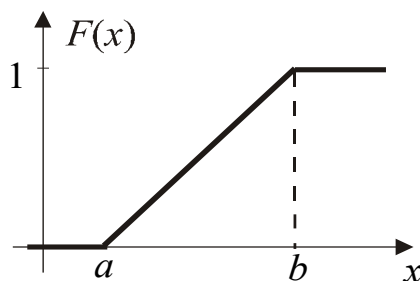


Рис. 2.2. Интегральная функция равномерного распределения

Математическое ожидание равномерного распределения:

$$M(X) = \frac{a + b}{2}$$

Дисперсия равномерного распределения:

$$D(X) = \frac{(b - a)^2}{12}$$
 Среднеквадратичное отклонение равномерного

распределения:

$$\sigma(X) = \frac{b - a}{2\sqrt{3}}$$

*В GPSS равномерно распределенные случайные величины имитируются библиотечной функцией Uniform (N,A,B), где N – номер генератора случайных чисел, A – нижняя граница, B – верхняя граница интервала.*

*Зачастую возникает необходимость имитировать равномерное распределение целых чисел. В этом случае используется целочисленная функция DUniform(N,A,B).*

*В модели можно имитировать равномерно распределенный поток транзактов двумя способами:*

1) GENERATE M,P

*где M – математическое ожидание  $(A+B)/2 = A+(B-A)/2$ ;*

*P – половина интервала  $(B-A)/2$ .*

2) GENERATE (Uniform (N,A,B))

*или GENERATE (DUniform (N,A,B)) для целочисленного распределения.*

## **2.6. Закон нормального распределения вероятностей**

Нормальное (гауссово) распределение или нечто близкое к нему часто имеет место в самых разных практических сферах. Из-за того оно и названо нормальным. В реальном многократно проводимом эксперименте в

стандартных условиях измеряемая величина будет принимать случайные значения, концентрирующиеся вблизи истинного значения (математического ожидания), но с некоторым разбросом, величина этого разброса характеризуется стандартным отклонением.

При воздействии на случайную величину множества случайных факторов, ни один из которых не имеет доминирующего значения, распределение случайной величины приближается к нормальному. Это утверждает центральная предельная теорема теории вероятностей.

Нормальное распределение имеет плотность вероятности

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} \quad (2.10)$$

где  $a$  – математическое ожидание,  $\sigma$  – среднеквадратичное отклонение.

График нормального распределения имеет куполообразную форму (рис.2.3), он симметричен относительно своего математического ожидания, а на степень его растянутости относительно математического ожидания влияет величина среднего квадратичного отклонения  $\sigma$ .

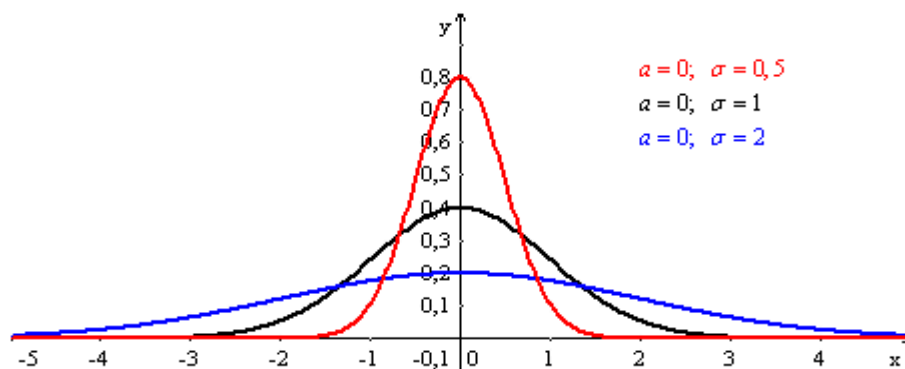


Рис. 2.3. Нормальный закон распределения случайной величины  
Интегральная функция нормального распределения вероятностей:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-a)^2}{2\sigma^2}} dx \quad (2.11)$$

Интегральная функция распределения вероятностей показывает вероятность того, что случайная величина примет значение меньшее, чем  $x$ :  $F(x) \leq P(\xi < x)$ . Численно она равна площади криволинейной трапеции, ограниченной сверху графиком плотности вероятности, снизу осью  $Ox$ , на интервале от  $-\infty$  до  $x$  (рис.2.4).

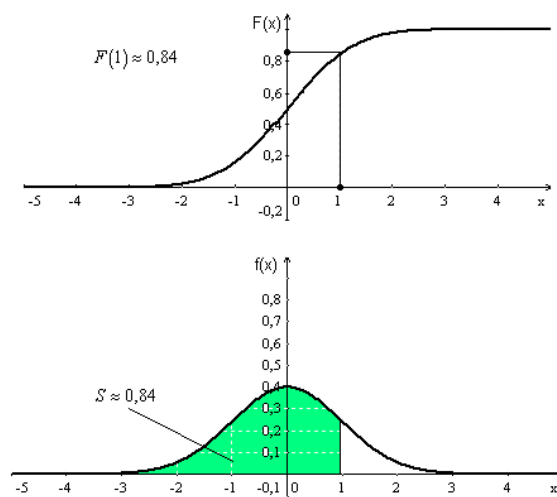


Рис. 2.4. Функция нормального распределения и закон распределения *B GPSS* нормальное распределение имитируется библиотечной функцией:

$Normal(N,A,B)$

где  $A$  – математическое ожидание,  $B$  – стандартное отклонение.

Но т.к. значения случайной величины при этом могут принимать и отрицательные значения, нужно учитывать этот факт при моделировании временных интервалов. Если  $MO \gg \sigma$ , можно применять функцию  $ABS(Normal(N,A,B))$ .

## 2.7. Закон экспоненциального распределения

Зачастую можно рассматривать функционирование сложной системы

как последовательность событий, происходящих в случайные моменты времени, т.е. через случайные временные интервалы. Например, подход автомобилей на перекресток можно рассматривать как поток событий, происходящих через случайные интервалы времени. Такого рода системы называются системами с непрерывным временем и довольно точно моделируются с помощью теории массового обслуживания. Наиболее просто математическая модель системы массового обслуживания строится, если поток событий в ней можно считать пуассоновским. В этом случае непрерывная случайная величина интервала времени между событиями описывается экспоненциальным (показательным) законом распределения

В теории потоков событий справедлива теорема, аналогичная центральной предельной теореме, но в ней речь идет не о суммировании случайных величин, а о суммировании потоков событий. Если рассматривать суммарный поток, составленный из большого числа независимых потоков, ни один из которых не оказывает преобладающего влияния на суммарный поток, то оказывается, что, когда характеристики потоков не зависят от времени, суммарный поток полностью описывается одним числом - интенсивностью потока. Для суммарного потока рассмотрим случайную величину  $T$  - длину промежутка времени между последовательными событиями. Говорят, что случайная величина  $T$ , математическое ожидание которой  $M(T) = T_{cp}$ , имеет экспоненциальное (показательное) распределение с параметром  $\lambda = 1/T_{cp} > 0$ , если она непрерывна, принимает только положительные значения, и имеет плотность распределения:

$$f(T) = \lambda e^{-\lambda T} \text{ при } 0 < T < \infty \quad (2.12)$$

На рис. 2.5 показаны графики закона (привязан к левой вертикальной оси ординат) и функции (привязан к правой оси ординат) показательного распределения с параметром  $\lambda = 1$ .

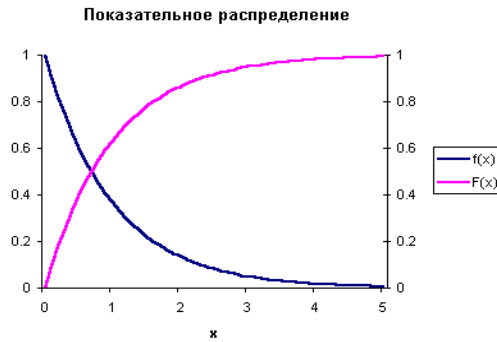


Рис. 2.5. Закон и функция экспоненциального распределения

Характеристики экспоненциально распределенной случайной величины вычисляются по формулам, приведенным в таблице:

Плотность распределения	$f(T) = \lambda e^{-\lambda T}$
Функция распределения	$F(T) = 1 - e^{-\lambda T}$
Математическое ожидание	$1 / \lambda$
Стандартное отклонение	$1 / \lambda$
Дисперсия	$1 / \lambda^2$

В GPSS экспоненциальное распределение имитируется библиотечной функцией *Exponential(N,0,T)*, где  $N$  – номер генератора случайных чисел,  $T = 1/\lambda$  – средний интервал времени между поступлениями заявок. Имитация потока заявок, поступающих в систему по экспоненциальному закону, выполняется блоком:

*GENERATE (Exponential(N,0,T))*

## ГЛАВА 3. Элементы математической статистики

### 3.1. Задачи математической статистики

Математическая статистика - это наука, которая занимается сбором, обработкой и описанием экспериментальных данных. При этом ставится цель выявить закономерности в случайном явлении по результатам испытаний (наблюдений), зафиксированных в виде числовых данных.

Статистические данные представляют собой данные, полученные в результате обследования большого числа объектов или явлений; следовательно, математическая статистика имеет дело с массовыми явлениями. С точки зрения имитационного моделирования это означает, что статистические данные необходимо накапливать по результатам достаточно большого количества событий в модели, т.е. большого количества «прогонов» модели в стандартных условиях.

Основные задачи математической статистики:

1. Определение на основе данных опыта неизвестного закона распределения случайной величины. Случайными в модели являются числовые данные, получаемые в результате модельного эксперимента.

2. Определение по данным опыта неизвестных параметров распределения.

3. Проверка статистических гипотез.

Эффективность и надежность расчетов с использованием теории вероятностей определяется исходным экспериментальным материалом и, следовательно, корректностью разработки структуры модели и подготовки исходных данных, которые, в свою очередь, также должны быть проанализированы методами математической статистики для определения числовых параметров случайных величин, задаваемых в модели.

*Первая задача* математической статистики - указать способы сбора и

группировки статистических сведений, полученных в результате наблюдений или в результате специально поставленных экспериментов.

**Вторая задача** математической статистики - разработать методы анализа статистических данных в зависимости от целей исследования.

Современная математическая статистика разрабатывает способы определения числа необходимых испытаний до начала исследования, в ходе исследования и решает многие другие задачи. Современную математическую статистику определяют как науку о принятии решений в условиях неопределенности

Итак, задача математической статистики состоит в создании методов сбора и обработки статистических данных для получения научных и практических выводов.

### **3.2. Генеральная и выборочная совокупность статистических данных**

Пусть требуется изучить совокупность однородных объектов относительно некоторого качественного или количественного признака, характеризующего эти объекты.

**Качественными** признаками объект обладает либо не обладает. Они не поддаются непосредственному измерению (например, тип товара на в торговой сети, квалификация сотрудника компании, национальность, территориальная принадлежность и т. п.).

**Количественные** признаки представляют собой результаты подсчета или измерения. В соответствии с этим они делятся на дискретные и непрерывные.

Иногда проводится сплошное обследование, т.е. обследуют каждый из объектов совокупности относительно признака, которым интересуются. На практике сплошное обследование применяют сравнительно редко. Например, если совокупность содержит очень большое число объектов, то

провести сплошное обследование физически невозможно. В таких случаях случайно отбирают из всей совокупности ограниченное число объектов и подвергают их изучению. Различают генеральную и выборочную совокупности.

**Выборочной совокупностью (выборкой)** называют совокупность случайно отобранных объектов.

**Генеральной (основной) совокупностью** называют совокупность всех объектов из которых производится выборка.

**Объемом совокупности** (выборочной или генеральной) называют число объектов этой совокупности. Например, если из 1000 деталей отобрано для обследования 100 деталей, то объем генеральной совокупности  $N = 1000$ , а объем выборки  $n = 100$ . Число объектов генеральной совокупности  $N$  значительно превосходит объем выборки  $n$ .

### **Способы выборки**

При составлении выборки можно поступать двумя способами: после того как объект отобран и над ним произведено наблюдение, он может быть возвращен либо не возвращен в генеральную совокупность. В соответствии со сказанным выборки подразделяют на повторные и бесповторные.

**Повторной** называют выборку, при которой отобранный объект (перед отбором следующего) возвращается в генеральную совокупность.

**Бесповторной** называют выборку, при которой отобранный объект в генеральную совокупность не возвращается.

На практике обычно пользуются бесповторным случайным отбором.

Для того чтобы по данным выборки можно было достаточно уверенно судить об интересующем признаке генеральной совокупности, необходимо, чтобы объекты выборки правильно его представляли (выборка должна правильно представлять пропорции генеральной совокупности) - выборка должна быть репрезентативной (представительной).

Выборка будет *репрезентативной*, если:

- каждый объект выборки отобран случайно из генеральной совокупности;
- все объекты имеют одинаковую вероятность попасть в выборку.

### 3.3. Полигон и гистограмма

Пусть теперь  $X$  – дискретная случайная величина, принимающая значения  $x_1, x_2, \dots, x_N$ , а  $x_1, x_2, \dots, x_n$  – выборка, т.е. это значения случайной величины, полученные в результате проведения опытов. Будем

полагать, что значение  $x_j$  наблюдалось  $n_j$  раз  $j = \overline{1, m}$ , причем  $\sum_{j=1}^m n_j = n$ .

Наблюдаемые значения  $x_1, x_2, \dots, x_n$  называют *вариантами*, а таблицу (табл.3.1) с упорядоченными по возрастанию вариантами и

соответствующими относительными частотами  $p_j^* = \frac{n_j}{n}$  ( $j = \overline{1, m}$ ) называют *вариационным рядом*.

Таблица 3.1

Вариационный ряд

$X=x_j$	$x_1$	$x_2$	$x_3$	...	$x_j$	...	$x_m$
$n_j$	$n_1$	$n_2$	$n_3$	...	$n_j$	...	$n_m$
$p_j^* = \frac{n_j}{n}$	$p_1^*$	$p_2^*$	$p_3^*$	...	$p_j^*$	...	$p_m^*$

Из определения относительной частоты следует, что  $\sum_{j=1}^m p_j^* = 1$ .

Если  $x_1, x_2, \dots, x_n$  – выборка реализаций непрерывной случайной величины  $X$  или  $n \gg 1$  (объем выборки большой) для дискретной случайной величины, то строится *интервальный вариационный ряд*. Методика его построения следующая (пп 1-7 отображены в табл. 3.2):

1. Определяются максимальное  $x_{\max}$  и минимальное  $x_{\min}$  значения выборки.

2. Определяется размах вариационного ряда

$$R = x_{\max} - x_{\min}.$$

3. Определяется количество интервалов

$$m = 1 + 3,22 \cdot \lg n.$$

Если  $m$  – не целое, то  $m$  округляется в большую сторону до ближайшего целого числа.

4. Определяется оптимальная ширина интервала  $\Delta x$ , позволяющая выявить характерные признаки  $X$  с минимальным количеством интервалов по формуле Стрэджеса:

$$\Delta x = \frac{R}{m} = \frac{x_{\max} - x_{\min}}{1 + 3,22 \lg n}.$$

5. Определяются границы интервалов  $[a_j, a_{j+1})$  следующим образом:

$$a_1 = x_{\min}, \quad a_{j+1} = a_j + \Delta x \leq j \leq m$$

(3.1)

6. Производится распределение вариантов по интервалам таким образом, что вариант  $x_j$  относят к интервалу  $[a_j, a_{j+1})$ , если  $a_j \leq x_j < a_{j+1}$ . Затем, распределив варианты по интервалам, подсчитывают их общее число для каждого  $j$  – го интервала  $n_j$  и вычисляют относительные частоты

$$p_j^* = \frac{n_j}{n}.$$

7. Для каждого интервала вычисляют представителя интервала

$$\bar{a}_j = \frac{a_j + a_{j+1}}{2} \quad 1 \leq j \leq m$$

Таблица 3.2

Построение интервального вариационного ряда

Номер интервала	1	2	...	$i$	...	$n$
Границы интервала $[a_j, a_{j+1})$	$a_1, a_2$	$a_2, a_3$	...	$a_j, a_{j+1}$		$a_m, a_{m+1}$
Частота попадания в интервал	$n_1$	$n_2$	...	$n_j$		$n_m$
Относительная частота попадания в интервал	$p_1^* = \frac{n_1}{n}$	$p_2^* = \frac{n_2}{n}$	...	$p_j^* = \frac{n_j}{n}$		$p_m^* = \frac{n_m}{n}$
Представитель интервала $\bar{a}_j$	$\bar{a}_1$	$\bar{a}_2$	...	$\bar{a}_j$	...	$\bar{a}_m$

Графическое представление интервального ряда носит название гистограммы. По данным дискретного вариационного ряда строят полигон частот или относительных частот.

**Полигоном частот** называют ломанную, отрезки которой соединяют точки  $(x_1; n_1), (x_2; n_2), \dots, (x_k; n_k)$ . Для построения полигона частот на оси абсцисс откладывают варианты  $x_i$ , а на оси ординат - соответствующие им частоты  $n_i$ . Точки  $(x_i; n_i)$  соединяют отрезками прямых и получают полигон частот (рис. 3.1).

**Полигоном относительных частот** называют ломанную, отрезки которой соединяют точки  $(x_1; p^*_1), (x_2; p^*_2), \dots, (x_k; p^*_k)$ , где  $p^*_i = n_i / \sum n_j$  - относительные частоты, с которыми случайная величина

принимает  $i$ -е значение. Для построения полигона относительных частот на оси абсцисс откладывают варианты  $x_i$ , а на оси ординат - соответствующие им относительные частоты  $W_i$ . Точки  $(x_i; W_i)$  соединяют отрезками прямых и получают полигон относительных частот.

В случае *непрерывного признака* целесообразно строить *гистограмму*.

*Гистограммой частот* называют ступенчатую фигуру, состоящую из прямоугольников, основаниями которых служат частичные интервалы длиной  $\Delta x$ , а высоты равны отношению  $n_i / \Delta x$  (плотность частоты).

Для построения гистограммы частот на оси абсцисс откладывают частичные интервалы, а над ними проводят отрезки, параллельные оси абсцисс на расстоянии  $n_i / \Delta x$ .

Площадь  $i$ -го частичного прямоугольника равна  $\Delta x \cdot n_i / \Delta x = n_i$  - сумме частот вариант  $i$ -го интервала; следовательно, площадь гистограммы частот равна сумме всех частот, т.е. объему выборки.

*Гистограммой относительных частот* называют ступенчатую фигуру, состоящую из прямоугольников, основаниями которых служат частичные интервалы длиной  $\Delta x$ , а высоты равны отношению  $W_i / \Delta x$  (плотность относительной частоты).

Для построения гистограммы относительных частот на оси абсцисс откладывают частичные интервалы, а над ними проводят отрезки, параллельные оси абсцисс на расстоянии  $W_i / \Delta x$  (рис. 3.2).

Площадь  $i$ -го частичного прямоугольника равна  $\Delta x \cdot W_i / \Delta x = W_i$  - относительной частоте вариант попавших в  $i$ -й интервал. Следовательно, площадь гистограммы относительных частот равна сумме всех относительных частот, т.е. единице.

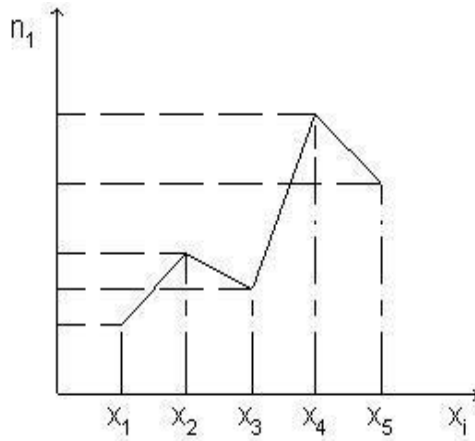


Рис. 3.1. Полигон частот

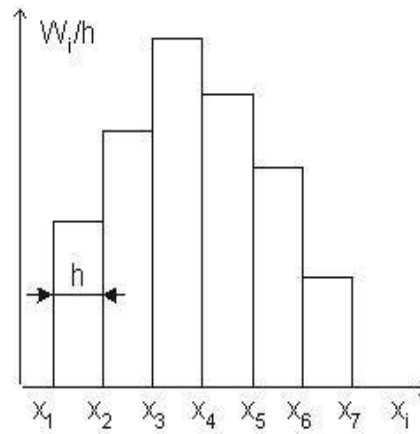


Рис. 3.2. Гистограмма  
относительных частот

### 3.4. Оценка параметров генеральной совокупности

Основными параметрами генеральной совокупности являются математическое ожидание (генеральная средняя)  $M(X)$  и среднее квадратическое отклонение  $\sigma$ . Это постоянные величины, которые можно оценить по выборочным данным. Оценка генерального параметра, выражаемая одним числом, называется точечной.

Точечной оценкой генеральной средней является выборочное среднее  $\bar{X}$ .

Выборочным средним называется среднее арифметическое значение признака выборочной совокупности.

Если все значения  $x_1, x_2, \dots, x_n$  признака выборки различны (или если данные не сгруппированы), то:

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

Если же все значения признака  $x_1, x_2, \dots, x_n$  имеют соответственно

частоты  $n_1, n_2, \dots, n_k$ , причем  $n_1 + n_2 + \dots + n_k = n$  (или если выборочное среднее вычисляется по вариационному ряду), то

$$\bar{X} = \frac{\sum_{i=1}^k x_i n_i}{n}$$

В том случае, когда статистические данные представлены в виде интервального вариационного ряда, при вычислении выборочного среднего  $\bar{X}$  значениями вариант считают середины интервалов.

Выборочное среднее является основной характеристикой положения, показывает центр распределения совокупности, позволяет охарактеризовать исследуемую совокупность одним числом, проследить тенденцию развития, сравнить различные совокупности (выборочное среднее является той точкой, сумма отклонений наблюдений от которой равна 0).

Для оценки степени разброса (отклонения) какого-то показателя от его среднего значения, наряду с максимальным и минимальным значениями, используются понятия дисперсии и стандартного отклонения.

Дисперсия выборки или выборочная дисперсия (от английского variance) – это мера изменчивости переменной. Термин впервые введен Фишером в 1918 году.

Выборочной дисперсией  $D_v$  называют среднее арифметическое квадратов отклонения наблюдаемых значений признака от их среднего значения  $\bar{X}$ .

Если все значения  $x_1, x_2, \dots, x_n$  признака выборки объема  $n$  различны, то:

$$D_v = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Если же все значения признака  $x_1, x_2, \dots, x_n$  имеют соответственно частоты  $n_1, n_2, \dots, n_k$ , причем  $n_1 + n_2 + \dots + n_k = n$ , то

$$D_{\varepsilon} = \frac{\sum_{i=1}^n n_i (x_i - \bar{x})^2}{n}$$

Дисперсия меняется от нуля до бесконечности. Крайнее значение 0 означает отсутствие изменчивости, когда значения переменной постоянны.

Среднее квадратическое отклонение (стандартное отклонение), (от английского standard deviation) вычисляется как корень квадратный из дисперсии.

$$\sigma_{\varepsilon} = \sqrt{D_{\varepsilon}}$$

Чем выше дисперсия или стандартное отклонение, тем *сильнее* разбросаны значения переменной относительно среднего.

Непараметрическими характеристиками положения являются мода и медиана.

Модой  $M_o$  называется варианта, имеющая наибольшую частоту или относительную частоту.

Медианой  $M_e$  называется варианта, которая делит вариационный ряд на две части, равные по числу вариант.

При нечетном числе вариант ( $n=2k+1$ )

$$M_e = x_{k+1},$$

а при четном числе вариант ( $n=2k$ )

$$M_e = (x_k + x_{k+1})/2.$$

## ГЛАВА 4. Основы теории массового обслуживания

### 4.1. Системы массового обслуживания

Теория массового обслуживания является основным инструментом анализа систем с массовыми запросами и устройствами их обслуживания. Такие системы называют системами массового обслуживания. Они обладают следующими характерными признаками:

- имеется входящий поток заявок на обслуживание;
- заявки выстраиваются в очередь на обслуживание в соответствии с заданной дисциплиной;
- осуществляется обслуживание заявок в устройствах в соответствии с заданными правилами;
- имеется выходящие потоки обслуженных и отвергнутых заявок.

При этом все события в такой системе являются вероятностными: заявки поступают в случайные моменты времени, находятся в очереди случайное время, обслуживаются в устройстве также в течение случайного периода времени и по выходе образуют стохастический поток.

Признаками системы массового обслуживания обладают подавляющее большинство сервисных систем в экономической, технической, организационной и других областях. Каждая СМО состоит из определенного числа обслуживающих единиц (приборов, устройств, пунктов, станций), которые будем называть каналами обслуживания. В качестве каналов обслуживания могут рассматриваться коридоры досмотра на таможенном посту, линии связи, грузовые терминалы, комплексы обработки информации, кассиры и др. По числу каналов обслуживания СМО подразделяют на одноканальные и многоканальные.

Предметом теории массового обслуживания является построение математических моделей, связывающих структуру СМО (число каналов, их производительность, характер потока заявок и т.п.) с показателями

эффективности СМО, описывающими ее способность справляться с потоком заявок.

В качестве показателей эффективности СМО используются: среднее число заявок, обслуживаемых в единицу времени (абсолютная пропускная способность); доля обслуженных заявок (относительная пропускная способность); среднее число заявок в очереди; среднее время ожидания обслуживания; средняя длина очереди; среднее время нахождения в системе; вероятность отказа в обслуживании без ожидания; вероятность того, что число заявок в очереди превысит определенное значение и т.п.

СМО делят на два основных типа (класса): СМО с отказами и СМО с ожиданием (очередью). В СМО с отказами заявка, поступившая в момент, когда все каналы заняты, покидает СМО без обслуживания (например, заявка на телефонный разговор в момент, когда все каналы заняты, получает отказ и покидает СМО необслуженной). В СМО с ожиданием заявка, пришедшая в момент, когда все каналы заняты, становится в очередь на обслуживание.

СМО с ожиданием в свою очередь подразделяются на разные виды в зависимости от того, как организована очередь: с ограниченной длиной очереди (очередь в консульство по записи) или неограниченной длиной очереди (очередь на таможенный досмотр на таможенной границе), с ограниченным временем ожидания и т.п.

Важное значение для классификации СМО имеет дисциплина постановки в очередь, когда все каналы обслуживания заняты, и выбора из нее, когда освобождается один из каналов. По этому признаку обслуживание заявки может быть организовано по принципу "первая пришла — первая обслужена", такая дисциплина именуется FIFO, "последняя пришла — первая обслужена" (дисциплина LIFO), заявки из очереди могут выбираться случайным образом или в соответствии с

приоритетом, когда в первую очередь обслуживаются наиболее важные заявки. Приоритет может быть как абсолютным, когда более важная заявка "вытесняет" с обслуживания обычную заявку (например, в случае аварийной ситуации плановые работы коммунальных служб прерываются до ликвидации аварии), так и относительным, когда более приоритетная заявка занимает место во главе очереди.

#### **4.2. Марковские процессы**

В теории массового обслуживания большое значение имеют марковские случайные процессы с дискретными состояниями и непрерывным временем.

Процесс называется процессом с дискретным состоянием, если его возможные состояния  $S_1, S_2, \dots$  можно заранее определить, и переход системы из состояния в состояние происходит «скачком», практически мгновенно.

Процесс называется процессом с непрерывным временем, если моменты возможных переходов из состояния в состояние не фиксированы заранее, а неопределенны, случайны и могут произойти в любой момент.

Процессы такого рода характерны для систем, обслуживающим потоки заявок, система скачком переходит из состояния «свободна» в состояние «занята» или «частично занята», очередь на обслуживание в случайные моменты времени скачком удлиняется при поступлении очередной заявки или укорачивается при принятии заявки на обслуживание.

Пример. Участок диагностики  $S$  состоит из двух приборов, каждый из которых в случайный момент времени может выйти из строя (отказаться), после чего мгновенно начинается ремонт узла, тоже продолжающийся заранее неизвестное, случайное время. Возможны следующие состояния системы:

$S_0$  - оба прибора исправны;

$S_1$  - первый прибор ремонтируется, второй исправен;

$S_2$  - второй прибор ремонтируется, первый исправен;

$S_3$  - оба прибора ремонтируются.

Переходы системы  $S$  из состояния в состояние происходят практически мгновенно, в случайные моменты выхода из строя того или иного прибора или окончания ремонта.

При анализе случайных процессов с дискретными состояниями удобно пользоваться геометрической схемой – графом состояний. Вершины графа – состояния системы. Дуги графа – возможные переходы из состояния в состояние. Для нашего примера граф состояний приведен на рис. 4.1.

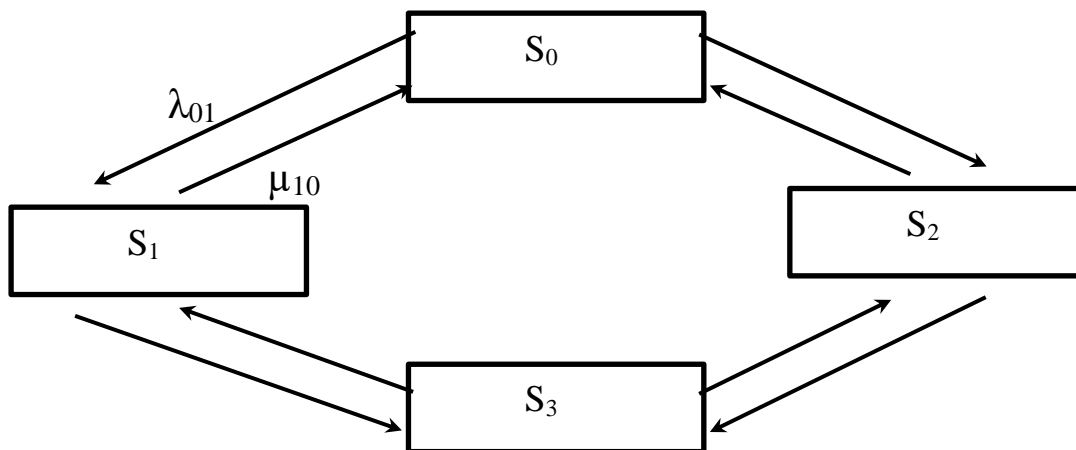


Рис. 4.1. Граф состояний системы

Переход из состояния  $S_0$  в  $S_3$  на рисунке не обозначен, т.к. предполагается, что приборы выходят из строя независимо друг от друга. Вероятностью одновременного выхода из строя обоих приборов пренебрегаем.

Такой граф, в котором направления возможных переходов обозначены стрелками, называется размеченным.

Подразумевается, что все переходы системы  $S$  из состояния в состояние происходят под действием простейших потоков событий

(потоков отказов приборов и потоков восстановлений). Если все потоки событий, переводящие систему  $S$  из состояния в состояние простейшие, то процесс, протекающий в системе, будет марковским. Из состояния  $S_0$  (оба прибора исправны) в состояние  $S_1$  (неисправен первый прибор) система переходит с интенсивностью

$$\lambda_{01} = 1/T_{cp},$$

где  $T_{cp}$  – среднее время безотказной работы первого прибора. Обратный переход происходит с интенсивностью  $\mu = 1/T_{рем}$ , где  $T_{рем}$  – время ремонта первого прибора. Аналогичные интенсивности присущи и остальным переходам: из состояния  $S_1$  в  $S_3$  и обратно и т.д.

Система характеризуется вероятностями нахождения в том или ином состоянии. Математическая модель системы строится, исходя из предположения о том, что вероятности переходов из одного состояния в другое подчиняются закону Пуассона, т.е. потоки событий в системе простейшие. Рассмотрим, к примеру, вероятность нахождения системы в состоянии  $S_1$ . Вероятность одного события в течение короткого интервала времени от  $t$  до  $t+\Delta t$  в соответствии с формулой (1.4) приближенно вычисляется:

$$P(t + \Delta t) = \lambda \Delta t \tag{4.1}$$

Если вероятность состояния  $S_1$  в момент времени  $t$  равна  $p_1(t)$ , то в момент  $t+\Delta t$  вероятность нахождения системы в состоянии  $S_1$  можно вычислить по формуле:

$$p_1(t+\Delta t) = p_1(t) + \lambda_{01}p_0 \Delta t + \mu_{31}p_3 \Delta t - \lambda_{10}p_1 \Delta t - \mu_{13}p_1 \Delta t \tag{4.2}$$

Несложные преобразования приводят к следующему равенству:

$$\frac{p_1(t + \Delta t) - p_1(t)}{\Delta t} = \lambda_{01}p_0 + \mu_{31}p_3 - \mu_{10}p_1 - \lambda_{13}p_1 \tag{4.3}$$

Переходя к пределу при  $\Delta t \rightarrow 0$  получим обыкновенное

дифференциальное уравнение первого порядка, описывающее изменение вероятности  $p_1$  во времени. Это уравнение Колмогорова-Чэпмена:

$$\begin{aligned} \frac{dp_1}{dt} &= \lambda_{01}p_0 + \mu_{31}p_3 - (\mu_{10} + \lambda_{13})p_1 \\ \frac{p_1(t + \Delta t) - p_1(t)}{\Delta t} &= \lambda_{01}p_0 + \mu_{31}p_3 - \mu_{10}p_1 - \lambda_{13}p_1 \end{aligned} \quad (4.4)$$

Сформулируем правило составления уравнений Колмогорова. В левой части каждого из них стоит производная вероятности  $i$ -го состояния. В правой части — сумма произведений вероятностей всех состояний (из которых идут стрелки в данное состояние) на интенсивности соответствующих потоков событий, минус суммарная интенсивность всех потоков, выводящих систему из данного состояния, умноженная на вероятность данного ( $i$ -го состояния). Аналогичные уравнения можно составить для всех остальных возможных состояний системы. Уравнения Колмогорова дают возможность найти все вероятности состояний как функции времени. Особый интерес представляют вероятности системы  $p_i(t)$  в предельном стационарном режиме, т.е. при  $t \rightarrow \infty$ , которые называются предельными (или финальными) вероятностями состояний.

В теории случайных процессов доказывается, что если число состояний системы конечно и из каждого из них можно (за конечное число шагов) перейти в любое другое состояние, то предельные вероятности существуют.

Предельная вероятность состояния  $S_i$  имеет четкий смысл: она показывает среднее относительное время пребывания системы в этом состоянии. Например, если предельная вероятность состояния  $S_0$  равна  $p_0 = 0,5$ , то это означает, что в среднем половину времени система находится в состоянии  $S_0$ . Очевидно, что для любого момента времени сумма всех вероятностей состояний равна единице:

$$\sum_{i=0}^3 p_i = 1$$

(4.5)

Вероятности  $p_i$  представляют собой долю времени, при котором система находится в  $i$ -м состоянии. По истечении некоторого времени, если системе сбалансирована, она переходит в равновесное состояние. Такой режим работы системы называется стационарным. При этом вероятности нахождения в том или ином состоянии становятся постоянными. Поскольку производные от констант равны нулю, уравнения (2.4) превращаются в алгебраические и вероятности  $p_i$  находятся стандартными методами решения систем линейных однородных уравнений.

Систему массового обслуживания можно представить в виде размеченного графа, в котором переходы возможны только между соседними состояниями. Обычно такой граф называют «схемой гибели и размножения», это название пришло из анализа биологических систем, в которых текущее состояние системы характеризуется количеством тех или иных живых организмов. При гибели или рождении нового организма система переходит в новое состояние. Поскольку процессы гибели и размножения рассматриваются как простейшие, т.е. в каждый момент времени может происходить только одно событие, такая система может скачкообразно переходить только в соседнее состояние. При постоянных интенсивностях переходов  $\lambda_{ij}$  и  $\mu_{ji}$  размеченный граф такой системы представляется в виде, показанном на рис. 4.2.

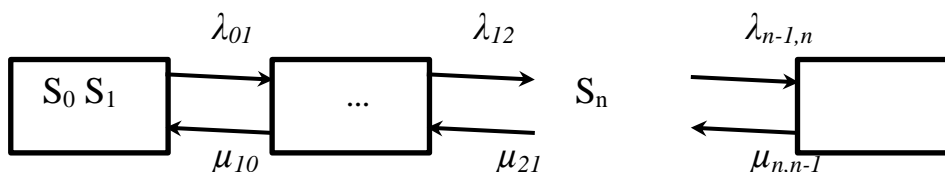


Рис. 4.2. Граф состояний системы массового обслуживания

По аналогии с (2.4) можно построить систему дифференциальных

уравнений для этого графа и перейти к стационарной задаче. Система алгебраических уравнений для предельных вероятностей в этом случае представляется в виде:

$$\begin{aligned}
 \lambda_{01}p_0 &= \mu_{10}p_1 \\
 (\lambda_{12} + \mu_{10})p_1 &= \lambda_{01}p_0 + \mu_{21}p_2 \\
 &\dots \\
 (\lambda_{i,i+1} + \mu_{i,i-1})p_i &= \lambda_{i-1,i}p_{i-1} + \mu_{i+1,i}p_{i+1} \\
 &\dots \\
 \mu_{n,n-1}p_n &= \lambda_{n-1,n}p_{n-1} \\
 \sum_{i=0}^n p_i &= 1
 \end{aligned} \tag{4.6}$$

На базе этой системы можно получить решения и основные показатели эффективности для типовых систем массового обслуживания.

### 4.3. Одноканальные устройства с отказами

Одноканальное устройство (ОКУ) с отказами может обслуживать в каждый текущий момент времени только одну заявку. Следует отметить, что в данном случае канал обслуживания единственный. Этот канал принимает пуассоновский поток заявок, интенсивность которого равняется постоянной величине  $\lambda$ . Если заявка прибыла в канал, который в данный момент не является свободным, она получает отказ и больше не числится в системе. Обслуживание заявок осуществляется в течение случайного времени  $T_{об}$ , распределение которого реализуется в соответствии с показательным законом с параметром  $\mu = 1/T_{об}$  (см. формулу 1.12).

При этом, если устройство занято обслуживанием, очередная заявка, пришедшая в это время, не может остаться в системе и покидает ее необслуженной. Такая система имеет всего два состояния: «ОКУ занято» и «ОКУ свободно». Граф системы представлен на рис. 4.3.

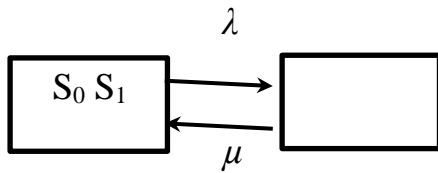


Рис.4.3. Граф одноканальной СМО с отказами

Уравнения Колмогорова для такой системы:

$$\begin{aligned} \frac{dp_0}{dt} &= -\lambda p_0 + \mu p_1 \\ \frac{dp_1}{dt} &= \lambda p_0 - \mu p_1 \end{aligned} \quad (4.7)$$

при начальных условиях, которые определяются в предположении, что в начальный момент времени ОКУ свободно:  $p_0 = 1; p_1 = 0$ . Решение такой системы:

$$\begin{aligned} p_0 &= \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t} \\ p_1 &= \frac{\lambda}{\lambda + \mu} - \frac{\mu}{\lambda + \mu} e^{-(\lambda + \mu)t} \end{aligned} \quad (4.8)$$

График изменения вероятностей  $p_0$  и  $p_1$  приведен на рис 4.4.

Предельные вероятности в установившемся режиме после окончания переходного периода:

$$\begin{aligned} p_0 &= \frac{\mu}{\lambda + \mu} \\ p_1 &= \frac{\lambda}{\lambda + \mu} \end{aligned} \quad (4.9)$$

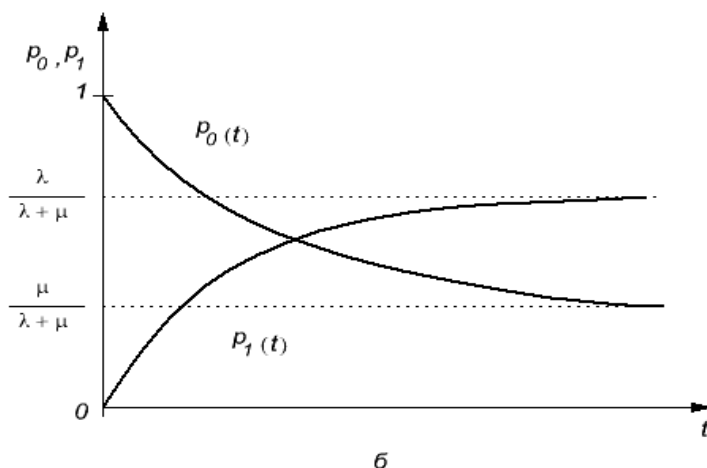


Рис. 4.4. График изменения вероятностей состояний ОКУ с отказами

Опираясь на результат (4.9) можно получить основные показатели эффективности ОКУ с отказами. Очевидно, доля обслуженных заявок, иначе говоря, относительная пропускная способность устройства, определяется вероятностью того, что в момент прихода заявки устройство окажется свободным:

$$Q = p_0 = \frac{\mu}{\lambda + \mu} \quad (4.10)$$

Абсолютная пропускная способность определяет количество заявок, обслуженных устройством в единицу времени. Количество заявок, поступающих в систему в единицу времени равно  $\lambda$ , из них обслуживается доля  $Q$ , соответственно количество заявок, обслуженных в единицу времени:

$$A = \lambda Q = \frac{\lambda \mu}{\lambda + \mu} \quad (4.11)$$

Отказ в обслуживании происходит при занятом устройстве, поэтому вероятность отказа в обслуживании равна вероятности занятости устройства:

$$P_{отк} = p_1 = \frac{\lambda}{\lambda + \mu} \quad (4.12)$$

#### 4.4. Многоканальные СМО с отказами

Рассмотрим систему с  $n$  идентичными каналами обслуживания.

Состояния  $n$  - канальной системы:

- $S_0$  - система полностью свободна;
- $S_1$  - занят один канал, остальные каналы свободны;
- $S_2$  - занято два канала, остальные каналы свободны;
- .....;
- $S_i$  - занято  $i$  каналов, остальные каналы свободны;
- .....;
- $S_n$  - заняты все  $n$  каналов.

Размеченный граф состояний рассматриваемой системы представлен на рис. 4.5.

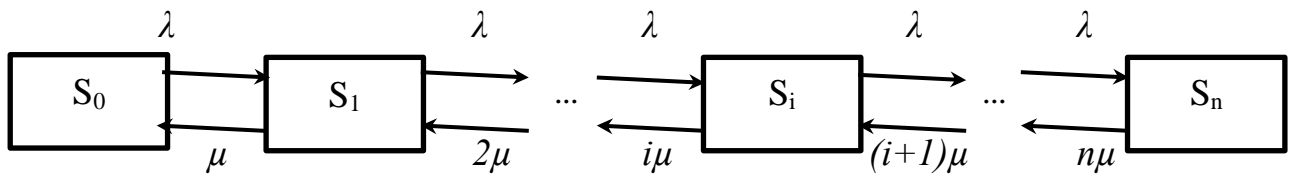


Рис. 4.5. Граф многоканальной системы с отказами

Переходы системы в более загруженное состояние происходят с интенсивностью поступления заявок, поэтому у стрелок, идущих слева направо, везде стоит интенсивность  $\lambda$ . Интенсивности же потоков обслуживания увеличиваются по мере включения все большего числа каналов обслуживания.

Поскольку входной поток и поток обслуживания являются пуассоновскими, то процесс является марковским и, пользуясь правилом, сформулированным уравнением (4.4), можно составить систему дифференциальных уравнений для вероятностей состояний (4.13).

Начальными условиями для решения этих уравнений являются

следующие значения вероятностей состояний в нулевой момент времени:

$$p_0(0) = 1 ; p_1(0) = p_2(0) = \dots = p_k(0) = \dots = p_n(0) = 0$$

Аналитическое решение для получения значений вероятностей состояний в функции времени довольно сложно. При необходимости целесообразно воспользоваться имеющимися стандартными программами на ЭВМ и получить значения с требуемой точностью и в интересующие нас моменты времени.

$$\begin{aligned} \frac{dp_0}{dt} &= -\lambda p_0 + \mu p_1 \\ \frac{dp_1}{dt} &= \lambda p_0 - (\lambda + \mu) p_1 + 2\mu p_2 \\ \frac{dp_2}{dt} &= \lambda p_1 - (\lambda + 2\mu) p_2 + 3\mu p_3 \\ &\dots \\ \frac{dp_k}{dt} &= \lambda p_{k-1} - (\lambda + k\mu) p_k + (k+1)\mu p_{k+1} \\ &\dots \\ \frac{dp_n}{dt} &= \lambda p_{n-1} - n\mu p_n \end{aligned} \tag{4.13}$$

Практическое значение имеют предельные вероятности состояний  $P_0, P_1, P_2, \dots, P_k, \dots, P_n$ . Для этого систему дифференциальных уравнений (4.13) преобразуем в систему алгебраических уравнений, приравняв все производные нулю.

Обозначим  $\rho = \frac{\lambda}{\mu}$ . Тогда выражение для вероятности того, что рассматриваемая система свободна, имеет вид:

$$P_0 = \left[ 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} \right]^{-1} \tag{4.14}$$

Для любых других вероятностей состояний получаем

$$P_k = \frac{\rho^k}{k!} P_0 \quad (4.15)$$

Заявка получает отказ, если приходит в момент, когда все  $n$  каналов заняты. Таким образом, вероятность отказа  $P_{отк}$  равна :

$$P_{отк} = P_n = \frac{\rho^n}{n!} P_0 \quad (4.16)$$

Относительная пропускная способность  $Q$  есть вероятность того, что заявка будет принята к обслуживанию, т.е. численно дополняет  $P_{отк}$  до единицы:

$$Q = 1 - P_{отк} \quad (4.17)$$

Абсолютная пропускная способность

$$A = \lambda(1 - P_{отк}) \quad (4.18)$$

Среднее число занятых каналов  $n_z$  (в данном случае оно совпадает со средним числом заявок, находящихся в системе) может быть определено из тех соображений, что абсолютная пропускная способность  $A$  есть среднее число заявок, обслуживаемых в единицу времени. Один занятый канал обслуживает в среднем за единицу времени  $\mu$  заявок. Таким образом, среднее число занятых каналов будет равно частному от деления  $A$  на  $\mu$ , т.е.

$$n_z = \frac{A}{\mu} = \rho(1 - P_{отк}) \quad (4.19)$$

#### 4.5. Одноканальные системы с ожиданием

Рассмотрим сначала одноканальную систему с ограниченной очередью, в которой имеется  $m$  мест.

Возможные состояния этой системы:

- S0 - канал свободен;
- S1 - канал занят, очереди нет;
- S2 - канал занят, одна заявка стоит в очереди;

- .....
- $S_i$  - канал занят,  $i-1$  заявок стоит в очереди
- .....
- $S_{m+1}$  - канал занят,  $m$  заявок стоит в очереди (накопитель полностью загружен).

Заявка, приходящая в момент, когда система находится в состоянии  $S_{m+1}$ , т.е. все места в очереди заняты, получает отказ. Граф состояний рассматриваемой системы приведен на рис. 4.6.

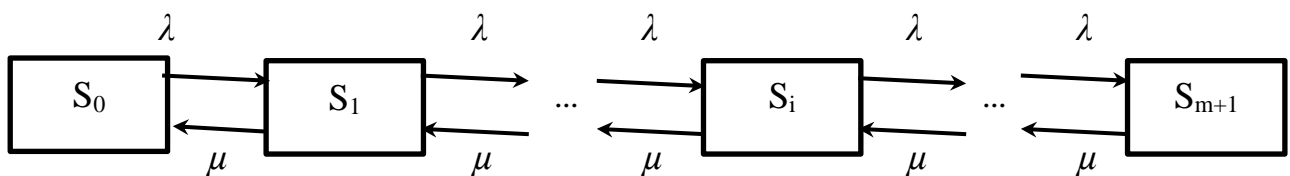


Рис. 4.6. Граф одноканальной системы с ожиданием

Для предельных вероятностей состояний система алгебраических уравнений имеет решение:

$$\begin{aligned}
 P_1 &= \rho P_0 \\
 P_2 &= \rho^2 P_0 \\
 &\dots \\
 P_k &= \rho^k P_0 \\
 P_{m+1} &= \rho^{m+1} P_0
 \end{aligned} \tag{4.20}$$

где, в соответствии с (2.14) при  $n=1$  получаем:

$$P_0 = [1 + \rho + \rho^2 + \dots + \rho^n]^{-1} \tag{4.21}$$

Выражение в скобках представляет собой геометрическую прогрессию с первым членом 1 и знаменателем прогрессии, равным  $\rho$ . Суммируя эту прогрессию, находим

$$P_0 = \frac{1-\rho}{1-\rho^{m+2}} \quad (4.22)$$

Формула (4.22) справедлива только при  $\rho \neq 1$ . При  $\rho = 1$  получаем неопределенность вида  $0/0$ , раскладывая которую по правилу Лопиталья, получим:

$$P_0 = \frac{1}{m+2} \quad (4.23)$$

Вероятность отказа  $P_{отк}$  определим, исходя из того, что заявка отвергается в том случае, когда занят единственный канал обслуживания и заняты все  $m$  мест в очереди, т.е. как следует из формул (4.20) и (4.22)

$$P_{отк} = P_{m+1} = \frac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}} \quad (4.24)$$

Относительная пропускная способность

$$Q = 1 - P_{отк} = \frac{1-\rho^{m+1}}{1-\rho^{m+2}} \quad (4.25)$$

Абсолютная пропускная способность

$$A = \lambda Q \quad (4.26)$$

Среднее число заявок  $L_{оч}$ , находящихся в очереди, определим как математическое ожидание дискретной случайной величины  $R$  - числа заявок, находящихся в очереди. Величину  $L_{оч}$  можно получить, умножая число заявок в очереди на соответствующие вероятности и производя суммирование

$$L_{оч} = 1 * P_2 + 2 * P_3 + \dots + (k-1) * P_k + \dots + m * P_{m+1}$$

Учитывая выражение для  $P_k$  ( $k = 2, 3, \dots, m+1$ ) и для  $P_0$ , после соответствующих преобразований можно получить

$$L_{оч} = \frac{\rho^2(1-(m+1-m\rho)\rho^m)}{(1-\rho)(1-\rho^{m+2})} \quad (4.27)$$

Общее число заявок в системе  $L_{смо} = L_{оч} + L_{обсл}$ , где  $L_{обсл}$  - математическое ожидание числа заявок, находящихся в устройстве обслуживания

$$L_{обсл} = 0 * P_0 + 1(1-P_0) = \frac{\rho - \rho^{m+2}}{1 - \rho^{m+2}} \quad (4.28)$$

т.е.

$$L_{смо} = L_{оч} + \frac{\rho - \rho^{m+2}}{1 - \rho^{m+2}} \quad (4.29)$$

Среднее время ожидания  $T_{ож}$  будет очевидно равно средней длине очереди, умноженной на средний интервал времени между поступлениями заявок. Для пуассоновского потока заявок этот интервал  $t_{cp} = 1/\lambda$ . Таким образом, имеем

$$T_{ож} = \frac{L_{оч}}{\lambda} \quad (4.30)$$

Среднее время обслуживания одной заявки  $t_{cp} = 1/\mu$ . Математическое ожидание времени обслуживания равно  $Q/\mu$ , общее среднее время пребывания заявки в системе

$$T_{смо} = \frac{L_{оч}}{\lambda} + \frac{Q}{\mu} \quad (4.31)$$

Для одноканальной системы с бесконечной очередью, установившийся режим возможен только при  $\rho < 1$ . В противном случае очередь растёт неограниченно. Вероятности состояний системы для случая  $\rho < 1$  можно получить из выражений для рассмотренной выше системы путём предельного перехода  $m \rightarrow \infty$ . Показатели качества такой системы приведены в Приложении 1.

#### 4.6. Многоканальная система с ожиданием

На  $n$  каналов системы поступает поток заявок с интенсивностью  $\lambda$ , интенсивность обслуживания одного канала -  $\mu$ , число мест в очереди -  $m$ .

Возможные состояния системы:

- $S_0$  - система полностью свободна;
- $S_1$  - занят один канал, остальные каналы свободны;
- .....;
- $S_k$  - занято  $k$  каналов, остальные каналы свободны;
- .....;
- $S_n$  - заняты все  $n$  каналов;
- $S_{n+1}$  - заняты все  $n$  каналов, одна заявка в очереди;
- .....;
- $S_{n+r}$  - заняты все  $n$  каналов,  $r$  заявок в очереди;
- .....;
- $S_{n+m}$  - заняты все  $n$  каналов, заняты все  $m$  мест в очереди.

В состояниях  $S_0 \div S_n$  - очереди нет.

Размеченный граф такой системы приведен на рис. 4.7.

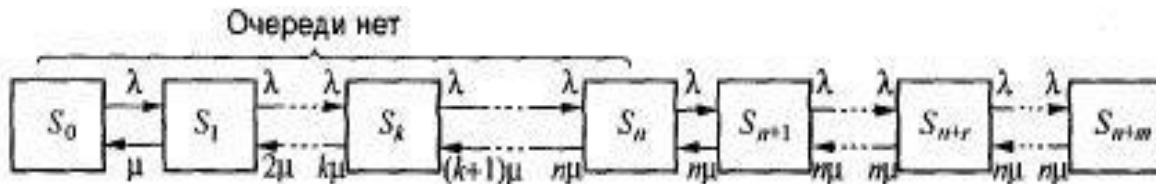


Рис. 4.7. Граф МКУ с ожиданием

По изложенной ранее методике можно составить систему алгебраических уравнений для такой системы и, обозначив

$\chi = \rho/n = \lambda/n\mu$ , получить решение в виде

$$P_0 = \left[ \sum_{k=0}^n \frac{\rho^k}{k!} + \frac{\rho^n}{n!} \frac{\chi - \chi^{m+1}}{1 - \chi} \right]^{-1} \quad (4.32)$$

Для остальных вероятностей состояний, используя уже имеющийся

принцип решения, получаем:

Пользуясь этим решением, можем записать выражения для

$$\left. \begin{aligned}
 P_1 &= \frac{\rho}{1!} P_0 \\
 P_2 &= \frac{\rho^2}{2!} P_0 \\
 &\dots\dots\dots \\
 P_n &= \frac{\rho^n}{n!} P_0 \\
 P_{n+1} &= \frac{\rho^{n+1}}{n \cdot n!} P_0 \\
 P_{n+2} &= \frac{\rho^{n+2}}{n^2 \cdot n!} P_0 \\
 &\dots\dots\dots \\
 P_{n+m} &= \frac{\rho^{n+m}}{n^m \cdot n!} P_0
 \end{aligned} \right\}$$

показателей качества рассматриваемой системы. Вероятность отказа означает, что заявка получает отказ, если все каналы и все места в очереди заняты.

$$P_{отк} = \frac{\rho^{n+m} P_0}{n^m n!} \tag{4.33}$$

Относительная пропускная способность дополняет вероятность отказа до единицы.

$$Q = 1 - P_{отк} \tag{4.34}$$

Абсолютная пропускная способность

$$A = \lambda(1 - P_{отк}) \tag{4.35}$$

Среднее число занятых каналов

$$n_3 = \rho \left[ 1 - \frac{\rho^n \chi^m P_0}{n!} \right] \quad (4.36)$$

Среднее число заявок в очереди также рассчитаем как математическое ожидание дискретной случайной величины количества заявок в очереди

$$L_{оч} = \frac{\rho^{n+1} P_0}{n n!} \cdot \frac{1 - \chi^m (m+1 - m\chi)}{(1 - \chi)^2} \quad (4.37)$$

Общее число заявок в системе

$$L_{смo} = L_{оч} + n_3 \quad (4.38)$$

Среднее время ожидания

$$T_{ож} = \frac{L_{оч}}{\lambda} \quad (4.39)$$

Время пребывания в системе

$$T_{смo} = \frac{L_{оч}}{\lambda} + \frac{Q}{\mu} \quad (4.40)$$

Выражения для параметров системы с бесконечной длиной очереди получаются из предыдущих формул путём предельного перехода  $m \rightarrow \infty$  и приведены в Приложении 1. Полученные выражения справедливы только при  $\chi < 1$ , иначе очередь растёт до бесконечности.

Одной из важнейших проблем имитационного моделирования является проверка адекватности моделей. Во многих случаях модель сложной системы можно представить в виде набора устройств обслуживания и очередей, для которых имеются аналитические решения, приведенные выше. Совпадение результатов моделирования с аналитическими решениями свидетельствует о нормальной работе отдельных элементов модели, хотя и не исчерпывает полностью вопрос адекватности модели сложной системы в целом. В частности, с помощью

теории затруднительно проанализировать потоки отказов элементов системы, нестационарные режимы работы, скачкообразные изменения свойств системы, обратные связи и пр.

Основные формулы для показателей эффективности систем массового обслуживания приведены в Приложении 7.

#### **Задание 4.1**

*Затраты на поддержание одной телефонной линии в колл-центре страховой компании составляют 40 000 руб. в месяц. Средний интервал между звонками 5,5 минут, среднее время беседы с клиентом - 15 минут. Местная АТС может удерживать до 5 звонков в режиме ожидания. Только 5% дозвонившихся заключают договор страхования, каждый договор приносит 10 000 руб. дохода.*

*Определить оптимальное количество телефонных линий для получения максимальной чистой прибыли за три года при 8-часовом рабочем дне и 24 рабочих днях в месяц (построить график зависимости, полученной прибыли за месяц от количества тлф. линий и выявить максимум). Исследовать диапазон от 1 до 10 телефонных линий.*

*Решить задачу с помощью табличного процессора Excel.*

### **ГЛАВА 5. Среда имитационного моделирования GPSS World**

#### **5.1. Концепция и возможности объектно-ориентированной моделирующей системы**

GPSS World является языком, проблемно ориентированным на имитационное моделирование систем массового обслуживания. Представление жизни модели как движения во времени заявок, перемещающихся в модели и обслуживающихся в устройствах очень естественно для основной части задач имитационного моделирования. Язык GPSS World, в своем последнем виде, достаточно прост для усвоения основных приемов моделирования. Строгая логика построения моделей

способствует развитию навыков анализа сложных систем. Автоматический сбор статистики облегчает задачу анализа системы и облегчает проведение модельных экспериментов.

Для многих реальных систем моделирование на GPSS World гораздо легче, чем другими методами. Компактные программы и возможность использования графического интерфейса позволяют ускорить создание прототипов моделей. При этом на каждую из них можно получить быстрый отклик после улучшений, проведенных пользователем. Этот последний фактор стал причиной широкого использования имитационного моделирования на практике.

Основные функции системы моделирования GPSS:

1) предоставление программных средств для формализованного описания дискретных компонентов моделируемой системы, дисциплин очередей и обслуживания, числовых и статистических результатов моделирования;

2) осуществление координации событий, определение путей прохождения транзактов, изменение режимов работы генераторов транзактов и компонентов системы.

Транзакт – это формальный запрос на какое-либо обслуживание. Транзакт в отличие от обычных заявок, которые рассматриваются при анализе моделей массового обслуживания, имеет набор динамически изменяющихся особых свойств и параметров. Пути движения транзактов по графу модели определяются логикой функционирования компонентов модели.

Транзакт является динамическим объектом модели, имитирующим движение заявок в системе обслуживания.

Транзакт может выполнять следующие действия:

– захватывать ресурсы и использовать их некоторое время, а затем

освобождать;

- порождать группы (семейства) других транзактов;
- поглощать другие транзакты конкретного семейства;
- определять времена обслуживания, накапливать информацию о пройденном пути и иметь информацию о своем дальнейшем пути и о путях других транзактов.

Основные параметры транзактов:

- уникальный идентификатор транзакта;
- идентификатор (номер) семейства, к которому принадлежит транзакт;
- время жизни транзакта;
- приоритет – неотрицательное целое число;
- параметры обслуживания в каком-либо обслуживающем устройстве, включая вероятностные характеристики.

Примеры транзактов:

- единица груза на складе;
- транспортная единица (корабль, автомобиль, вагон и пр.);
- заявка на товар;
- обрабатываемый документ;
- потребитель услуги и т.д.

Блоки модели представляют собой центры обслуживания транзактов. В блоках транзакты могут задерживаться, обслуживаться, порождать семейства новых транзактов, уничтожать другие транзакты. Транзакт всегда принадлежит одному из блоков модели.

Событиями в модели являются факты выхода из генератора очередного транзакта, регистрации транзакта в очереди и выхода из нее, входа-выхода из устройств обслуживания, блокировки и разблокировки устройств, временного вывода транзактов из процесса моделирования и

возвращения, определение маршрута движения очередного транзакта и т.д. Разработчик модели не может управлять этими событиями вручную, но задает правила возникновения этих событий на этапе разработки модели. Функция управления событиями отдана специально управляющей программе-планировщику, определяющему порядок наступления событий в модели.

## **5.2. Управление модельным временем**

Как уже упоминалось выше, модельное время отличается от астрономического, позволяя моделировать как микро- так и макро процессы. Например, процессы на молекулярном уровне происходят за микроскопические доли секунды, экономические – за минуты и часы, геотектонические – за тысячи и миллионы лет. Разработчик модели может настроить модельные часы на любой масштаб времени путем задания единицы модельного времени (ЕМВ) на этапе проектирования.

Модельное время в системе GPSS организовано в соответствии с дискретно-событийным принципом. Это означает, что после очередного события, часы «пропускают» время, в которое события отсутствуют и сразу переходят к моменту, на который спрогнозировано следующее событие, т.е. время в модели наращивается до ближайшего основного события. Модель переходит от события к событию, а модельные часы регистрируют моменты наступления этих событий. Таким образом, функционирование модели протекает в искусственном (не в реальном и не в машинном) времени, обеспечивая появление событий в требуемом логикой работы исследуемой системы порядке и с надлежащими временными интервалами между ними. При этом надо учитывать, что элементы реальной системы функционируют одновременно (параллельно), а компоненты программной модели действуют последовательно, так как реализуются с помощью ЭВМ последовательного действия. Поскольку в различных частях объекта

моделирования события могут возникать одновременно, то для сохранения адекватности причинно-следственных временных связей необходимо в языке имитационного моделирования предусмотреть механизм разрешения «временных узлов», т.е. определить порядок наступления совпадающих по времени событий.

### **5.3. Объекты языка GPSS**

**Наименование категорий:** операционная, аппаратная, динамическая, вычислительная, статистическая, запоминающая, группирующая.

**Наименование типов объекта:** блоки, транзакты, устройства, логические ключи, арифметические и булевские переменные, функции, очереди, таблицы, ячейки, матрицы ячеек, группы, списки.

**Блоки.** С объектами связаны определенные совокупности блоков, описываемых функционирование самой моделируемой системы либо содержащих дополнительную информацию о порядке моделирования.

Блоки, задают логику функционирования модели системы и определяют пути движения транзактов между объектами аппаратной категории. В блоках могут происходить события четырех основных типов:

1. Создание или уничтожение транзактов;
2. Изменение числового атрибута объекта;
3. Задержка транзакта на определенный период времени;
4. Изменение маршрута движения транзакта в модели.

Версия GPSS, реализованная в системе GPSS World, содержит 53 типа блоков.

В зависимости от назначения основные блоки подразделяются на несколько групп.

1. Блоки, осуществляющие модификацию атрибутов транзактов:
  - генерирование и уничтожение транзактов GENERATE, SPLIT, TERMINATE, ASSEMBLE;

- временная задержка ADVANCE;
- синхронизация движения двух – MATCH и нескольких – GATHER транзактов;

- изменение приоритета транзактов PRIORITY;
- изменение параметров транзактов ASSIGN, INDEX, MARK, PLUS.

2. Блоки, изменяющие маршруты движения транзактов (блоки передачи управления): DISPLACE, TRANSFER, LOOP, TEST, GATE.

3. Блоки, связанные с группирующей категорией: ADOPT, ALTER, EXAMINE, JOIN, REMOVE, SCAN.

4. Блоки, описывающие объекты аппаратной категории:

- одноканальные устройства (технические средства) SEIZE , RELEASE, PREEMPT, RETURN, FUNAVAIL, FAVAIL;

- многоканальные устройства (памяти) ENTER , LEAVE , SAVAIL, SUNAVAIL;

- ключи (логические переключатели) LOGIC .

5. Блоки, сохраняющие необходимые значения для дальнейшего использования: SAVEVALUE, MSAVEVALUE .

6. Блоки для получения статистических результатов:

- очереди QUEUE , DEPART;

- таблицы TABULATE .

7. Блоки для организации списков пользователя: LINK, UNLINK.

Алфавитный указатель блоков GPSS World приведен в Приложении 2. В Приложении 3 приведен предметный указатель блоков с описанием операндов блоков.

**Операционная категория.** Блоки и транзакты – два основных типа объектов языка GPSS. Практически все изменения состояния модели происходят в результате входа транзактов в блоки и выполнения ими своих функций.

**Аппаратная категория.** Язык GPSS оперирует тремя группами оборудования:

1. Одноканальные устройства (**ОКУ**) представляют собой оборудование, которое в любой момент времени может быть занято только одним транзактом. Например, один канал передачи данных, одноканальный ремонтный орган, один станок изготовления деталей, одно транспортное средство.

2. Многоканальные устройства (**МКУ**) предназначены для имитации оборудования, осуществляющего параллельную обработку. Они могут быть использованы одновременно несколькими транзактами. МКУ можно использовать в качестве аналога, например, многоканального ремонтного органа, нескольких каналов связи.

3. Логические ключи предназначены для моделирования такого оборудования, как переключатели, имеющие только два состояния.

К группе ОКУ относятся блоки **SEIZE**, **RELEASE**, **PREEMPT**, **RETURN**. Введение в моделирующую программу описания устройства позволяет автоматически регистрировать статистическую информацию.

Группу МКУ образуют блоки **ENTER**, **LEAVE** и команда описания **STORAGE**.

Для управления ключами используется блок LOGIG. Предусмотрено три режима изменения ключа: сброс в «0»; установка в «1»; инвертирование изменения состояния ключа на противоположное.

**Динамическая категория.** Динамические объекты – это транзакты, которые являются аналогами заявок в реальной системе. Они могут представлять собой различные типы заявок и событий даже в одной модели.

Каждый транзакт имеет набор параметров, которые присваиваются при рождении транзакта (порядковый номер, время появления) или создаются пользователем путем присвоения параметру транзакта заданного

значения. Все транзакты по умолчанию имеют нулевой приоритет, пользователь может увеличивать приоритет либо непосредственно в генераторе транзактов, либо в теле модели. С транзактами связаны блоки, основные из которых можно подразделять на пять групп:

*группа задержки* состоит из единственного блока ADVANCE;

*группа создания* – из блоков GENERATE, SPLIT, ASSEMBLE;

*группа уничтожения транзактов* состоит из единственного блока TERMINATE;

*группа изменения маршрутов транзактов* – из блоков TRANSFER, LOOP, GATE, TEST. Блок TRANSFER имеет шесть основных режимов использования;

*группа синхронизации транзактов* включает в себя блоки MATCH и GATHER. Сопряженные блоки MATCH не допускают продвижения транзакта, поступившего первым, пока в сопряженный блок не поступил второй транзакт.

**Вычислительная категория.** В вычислительной категории используются объекты трех видов: арифметические переменные, логические (булевы) переменные и функции. Кроме этого данные могут сохраняться и изменяться в параметрах транзактов, сохраняемых ячейках и матрицах сохраняемых ячеек. Арифметические переменные описываются командой VARIABLE, булевы – BVARIABLE.

Функции описываются с помощью команды FUNCTION. Основные типы функций – дискретная и непрерывная (кусочно-линейная).

Все объекты GPSS, в том числе и объекты вычислительной категории, имеют числовые атрибуты. В процесс моделирования GPSS автоматически регулирует и корректирует определенную информацию различных объектов, используемых в модели. Доступ к этой части информации осуществляется с помощью стандартного числового атрибута. Имя

стандартного числового атрибута состоит из двух частей. Первая часть указывает групповое имя (префикс). Оно идентифицирует одновременно и тип элемента (т.е. прибор, многоканальное устройство, очередь) и тип информации (например, счетчик занятий прибора, нагрузку многоканального устройства). Вторая часть идентифицирует конкретного члена группы (т.е. какой прибор, какое многоканальное устройство).

Для вызова значений объектов вычислительной категории, которые задаются пользователем, префиксы таковы:

**V\$Name** – вызов значения переменной с именем Name;

**X\$Name** – вызов значения сохраняемой ячейки с именем Name;

**P\$Name** – вызов значения параметра транзакта с именем Name;

**FN\$Name** – вызов значения функции с именем Name;

**BV\$Name** – вызов значения булевской переменной с именем Name.

Кроме этого СЧА автоматически создаются системой для всех объектов GPSS. Числовые характеристики, присущие модели в целом (модельное время, значение счетчика завершения и пр.) называются системными числовыми атрибутами. Полный список числовых атрибутов приведен в Приложении 1.

**Статистическая категория.** В языке GPSS используется 2 типа статистических объектов: очереди и таблицы.

Очередь выделяется блоками QUEUE и DEPART. Частоты попадания заданного СЧА регистрируются блоком TABULATE. Характеристики таблиц вводятся командой TABLE.

Для сохранения в модели числовой информации отводятся специальные ячейки основной памяти SAVEVALUE. Совокупности ячеек можно организовать в матрицы. Для записи информации в ячейки служит блок SAVEVALUE, в матрицы – MSAVEVALUE. Матрица обязательно описывается в сегменте описаний с помощью команды MATRIX.

Присвоение начальных значений ячейкам и матрицам осуществляется блоком INITIAL, с помощью которого можно устанавливать также логические ключи в состояние «1». Например, блок INITIAL LS3 установит третий логический ключ в состояние «1».

**Группирующая категория.** Из объектов группирующей категории рассмотрим блоки LINK и UNLINK, позволяющие переводить транзакты из списка текущих активных событий в списки пользователя неактивных событий и обратно. Использование списков пользователя позволяет моделировать различные дисциплины обслуживания сообщений.

#### **5.4. Синтаксис элементов языка**

**Алфавит.** Алфавит языка GPSS состоит из латинских букв от A до Z; цифр от 0 до 9 и специальных символов.

Русские буквы могут использоваться только в комментариях.

**Числа.** В языке GPSS различают два типа чисел полные и действительные. Признаком действительного числа является десятичная точка. Числа могут занимать в памяти ЭВМ два байта (полусловные) и четыре байта (полнословные).

**Идентификаторы.** Они должны содержать алфавитно-цифровые символы и знаки подчеркивания, причем не совпадающие со служебными словами языка GPSS. Идентификаторы используются для формирования имен объектов (устройств, очередей, переменных, меток и пр.) Именованье объектов в GPSS может выполняться двумя способами в виде числового имени и символического.

#### **5.5. Структура программы на языке GPSS**

Модель в среде GPSS состоит их предложений, каждое предложение занимает одну строку, содержащую либо управляющий оператор (например, START), либо объявление имени объекта, либо функциональный оператор (блок).

Комментарий к любой строке, в том числе и заголовок сегмента, может быть написан в этой же строчке после символа “;” (точка с запятой). Если комментария в строке нет, то точка с запятой может не ставиться.

Структура программы на языке GPSS состоит из следующих сегментов:

1. Сегмент описаний, содержащий:

- объявление многоканальных устройств в формате:  
*имя\_MKV STORAGE количество каналов MKV*
- описание функций;
- описание констант;
- описание переменных (для задания конкретных значений переменным или их вычисления по алгебраическим формулам);
- описание таблиц.

2. Сегменты имитации, содержащий последовательность блоков, т.е. последовательность функциональных операторов, по сути представляющий собой программу моделирования;

3. Сегмент завершения процесса моделирования, которое осуществляется либо по количеству транзактов, либо по времени завершения.

Функциональные операторы или блоки являются собственно языком описания модели. Каждый блок занимает одну строку GPSS-программы и имеет следующую структуру:

*[метка] БЛОК список\_операндов[; комментарий]*

Поля в списке операндов отделяются друг от друга запятой и имеют фиксированные позиции, назначение и количество полей зависит от типа блока. С точки зрения программиста блоки являются именами подпрограмм, а поля блока – параметрами подпрограммы.

## 5.6. Организация поступления транзактов в модель

GENERATE - это блок, через который транзакты входят в модель.

Блок имеет следующий формат записи:

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

Скобки [ ] означают, что данный операнд является необязательным.

Не существует ограничений на число различных блоков GENERATE в одной модели.

Интервалы времени между последовательными появлениями транзактов блока GENERATE называют интервалом поступления. Все разработчики должны задавать спецификацию распределения интервалов времени поступления в блоке GENERATE. Информация, необходимая для этого, задается операндами A и B.

Операнд A – средний интервал времени между последовательными поступлениями транзактов в модель.

Операнд B – задает модификатор, который изменяет значения интервала генерации транзактов по сравнению с интервалом, указанным операндом A. Есть два типа модификаторов: модификатор-интервал и модификатор-функция.

С помощью модификатора-интервала задается равномерный закон распределения времени между моментами появления транзактов, при этом A – среднее значение времени, B – полуширина интервала возможных значений случайной величины.

С помощью модификатора-функции задается интервал поступления транзактов, вычисляемый как произведение операнда A на значение функции, заданной операндом B.

Операнды A и B могут быть именем, положительным числом, выражением в скобках или непосредственно СЧА.

Когда операнды A и B задают в виде констант (B - модификатор-интервал), они должны быть неотрицательными числами. При вычислении

разности значений  $(A-B)$ , заданных операндами  $A$  и  $B$ , получается нижняя граница интервала, а при вычислении суммы  $(A+B)$  - верхняя граница. После генерации очередного транзакта разыгрывается случайное число из полученного интервала, это и будет значение времени, через которое следующий транзакт выйдет из блока GENERATE.

К примеру, задан интервал времени, равномерно распределенный от 4 до 12. Предположим, что транзакт входит в модель в момент модельного времени 25.6. После того, как этот транзакт попадет в следующий блок модели, планировщик GPSS разыграет случайное значение из интервала времени, равного  $8 \pm 4$ . Пусть разыгранным значением будет число 9.7. Тогда планировщик назначает блоку GENERATE появление следующего транзакта на момент модельного времени  $25.6 + 9.7 = 35.3$ .

Детерминированное значение интервалов времени задается, если операнд  $B = 0$ .

Более сложные интервалы времени поступления транзактов (не по равномерному закону) могут быть заданы с использованием модификатора-функции или библиотечных случайных функций. Под действием модификатора-функции значение операнда  $A$  умножается на значение функции, заданной операндом  $B$ .

При любом способе вычисления интервала времени значение операнда  $B$  не должно превышать значения операнда  $A$ , в противном случае в блоке GENERATE может быть получен отрицательный интервал времени, который вызовет останов по ошибке "Отрицательное время задержки".

Три дополнительных операнда:  $C$  – задание момента появления первого транзакта,  $D$  - ограничитель количества транзактов в данном генераторе в течение времени моделирования,  $E$  - уровень приоритета.

Операнд  $D$  задает граничное значение общего числа транзактов, которые могут войти в модель через данный блок GENERATE в течение

времени моделирования. Когда это число достигнуто, данный блок GENERATE перестает быть активным. Если не определено граничное значение (операнд D не используется), блок GENERATE остается активным в течение всего времени моделирования, т. е. по умолчанию ограничения на количество создаваемых транзактов нет.

Операнд E устанавливает приоритет каждого из транзактов, входящих в модель через данный блок GENERATE. Для задания приоритетов с целью повышения эффективности работы GPSS World рекомендуется использовать последовательность целых чисел 0, 1, 2, 3... Чем выше число, тем выше приоритет. Если операнд E не используется, по умолчанию приоритет генерируемых данным блоком GENERATE транзактов равен нулю.

Операнды D и E могут задаваться также, как и операнды A, B и C, но при этом принимать значения только целых положительных и целых чисел соответственно.

В любом блоке GENERATE должен быть обязательно задан либо операнд A, либо операнд D. Нельзя использовать в качестве операнда параметры транзактов.

Необходимо также помнить, что транзакт не должен входить в блок GENERATE. Если транзакт пытается это сделать, возникает ошибка выполнения. Примеры записи блоков GENERATE:

*с операндом A:*

**GENERATE 38.6** ;генерация регулярной последовательности транзактов с интервалом 38.6 единиц модельного времени (EMV)

*с операндами A и B:*

**GENERATE 7.25,X\$Otk** ;поток транзактов с равномерно распределенными интервалами со средним 7.25 и полуинтервалом, заданным СЧА сохраняемой ячейки с именем "Otk";

**GENERATE X\$Sohr, FN\$Function\_2**; момент появления очередного транзакта вычисляется как произведение СЧА сохраняемой ячейки “Sohr” на значение (СЧА) функции с именем “Function\_2”;

*с операндами A и C:*

**GENERATE 7.3,,4.1** ;регулярный поток транзактов с интервалами 7.3, первый появится в момент 4.1;

**GENERATE (Normal(3,V\$Sredn,V\$Sko)),,V\$Nach**  
;генерация нормально распределенного потока танзактов со средним, заданным переменной “Sredn”, и среднеквадратичным отклонением, заданным переменной “Sko”, начиная с момента времени, заданного переменной “Nach”;

*с операндами A, B, E:*

**GENERATE 13.3,2.8,,1** ;генерация равномерно распределенного потока со средним 13,3; полуинтервалом 2,8 и приоритетом = 1.

Конечно, все вычисляемые данные (переменные, функции) должны быть определены в сегменте описаний, а сохраняемые ячейки определены с помощью команд INITIAL до начала генерации.

### ***Задание 5.1***

*а) создать генератор потока заявок с равномерно распределенным интервалом времени поступления в диапазоне 4-10 ЕМВ;*

*б) создать генератор потока заявок со средним интервалом времени 20 и полуинтервалом, равным номеру по списку группы, деленному на 3;*

*в) создать генератор потока заявок типа б) с момента модельного времени = 2,4 и с приоритетом транзактов = 3;*

*г) сгенерировать 5 транзактов с момента времени 350 со средним 100 и полуинтервалом 60 ЕМВ, в параметр “Name” вписать собственные имя*

и фамилию, в параметр “Number” вписать порядковый номер транзакта, в его параметр.

### 5.7. Параметры транзактов. Изменение приоритета

В процессе моделирования в любой точке модели пользователь может дополнить структуру данных транзакта любыми числовыми, текстовыми и булевыми величинами путем записи данных в параметр транзакта блоком

**ASSIGN A,B,[C]**

Операнд A определяет имя (номер) параметра транзакта, которому присваивается новое значение. Если такого параметра не существовало, в этот момент он создается. Операнд B – значение, которое записывается в указанный параметр. Если первый операнд используется в формате A+ или A-, то к существующему значению операнда A плюсуется или вычитается из него величина, заданная операндом B.

Операнд C задает имя (номер) функции-модификатора, при этом значение операнда B умножается на значение операнда C.

Примеры записи блока ASSIGN:

**ASSIGN Power,54.3** ;параметру “Power” присваивается 54.3

**ASSIGN Line+,Q\$ATS3** ;к значению параметра “Line” прибавляется значение длины текущей очереди ATS3;

**ASSIGN Param\_3-,5.85,Seven** ;из значения параметра Param\_3 вычитается произведение 5.85 на вычисленное значение функции “Seven”;

**ASSIGN Name,"Prokofiy"** ;параметру с именем Name присваивается строка “Prokofiy”

**ASSIGN Tr1-,(Normal(2,10,5),Function\_2** ;вычисляются выражение в скобках и функция с именем “Function\_2”, перемножаются и полученное произведение вычитается из значения параметра с именем Tr1

Параметры транзактов могут также изменяться блоками PLUS, INDEX, MARK. С помощью блока MARK A в параметр транзакта с именем,

заданным в операнде А, вписывается текущее абсолютное время модели. В некоторой точке модели после этого можно обратиться к СЧА MP\$A, чтобы определить время прохождения транзактом участка модели между блоком MARK и блоком, в котором пользователь обратился к СЧА MP\$A.

Примеры использования блока MARK:

```
...  
MARK      StartTime  
...  
ASSIGN    Period,MP$StartTime  
...
```

В некоторой точке модели (начало участка) в параметр “StartTime” вписано текущее время, а в другой точке модели (конец участка модели) вычислена длительность прохождения участка модели транзактом и вписана в параметр “Period”.

Приоритет транзакта в процессе его продвижения по модели может быть изменен блоком **PRIORITY N**, где N – новое значение приоритета.

*Примечание 1.* В тех случаях, когда операндом блока может служить только параметр транзакта, префикс P\$ не используется. Это относится к блокам ASSIGN, PLUS, INDEX, MARK и блоку задания цикла LOOP (будет рассмотрен позже).

*Примечание 2.* В случаях, когда используются не имена объектов, а присвоенные им номера, знак \$ в префиксе не используется. Соотнести имя и номер объекта можно с помощью команды EQU (эквивалентно) в сегменте описаний:

```
MyName    EQU 55
```

### **Задание 5.2**

Присвоить параметру транзакта с именем “StudentNum” свой номер по списку группы.

Увеличить этот номер на три единицы.

Вписать в параметр с именем “LocTime” текущее время.

*Вписать в параметр "Period" время от момента записи "LocTime".*

### **5.8. Удаление транзактов из модели и завершение моделирования**

Транзакты удаляются из модели, попадая в блок TERMINATE (завершить). Блоки TERMINATE всегда позволяют войти всем транзактам, которые пытаются это сделать. В модели может быть любое число блоков TERMINATE. Блок имеет следующий формат записи:

**TERMINATE [A]**

Значением операнда А является число единиц, на которое блок TERMINATE уменьшает содержимое счетчика завершения, определяющего момент окончания моделирования. Операнд А может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*<параметр>. По умолчанию значение операнда А равно нулю. В этом случае транзакт уничтожается, а значение счетчика завершения не меняется.

Счетчик завершения представляет собой ячейку памяти с именем TG1, которая хранит положительное целое число. Это число записывается в ячейку TG1 командой START в начале процесса моделирования.

В процессе моделирования транзакты попадают в блок TERMINATE и в соответствии со значением операнда А вычитают определенное число из счетчика завершения. При достижении содержимым счетчика нуля моделирование завершается. В модели может быть много блоков TERMINATE, но счетчик завершения только один.

Когда пользователь подготавливает модель, он задает время моделирования, указывая в операторе START значение счетчика завершения. Поскольку пути прохождения транзактов в модели имеют различный физический смысл, каждый блок TERMINATE может либо уменьшать, либо не уменьшать содержимое счетчика завершения.

Рассмотрим пример, в котором блок TERMINATE и команда START

используются для управления временем моделирования. Промоделировать поведение системы в течение нужно в течение 10 часов, затем моделирование должно быть закончено. За единицу модельного времени возьмем 1 мин, тогда время моделирования равно  $10 \cdot 60 = 600$  единицам.

Любая модель на GPSS состоит из одного или нескольких сегментов. Для управления временем моделирования разработчик:

- 1) включает в модель сегмент из блоков GENERATE и TERMINATE;
- 2) в блоке TERMINATE в качестве операнда A использует 1;
- 3) во всех прочих блоках TERMINATE модели операнд A не используется.

### Пример 5.1

```
GENERATE 2,1 ;Поток транзактов-заявок в модели
...
TERMINATE ;Удаление обслуженных заявок

;Сегмент завершения
GENERATE 600 ;Генерация одного завершающего транзакта через 600
EMB
TERMINATE 1 ;Удаление и списание единица со счетчика завершения
START 1 ;Задание содержимого счетчика завершения (равно 1)
```

В процессе моделирования транзакты, которые двигаются в других сегментах модели, время от времени выводятся из модели блоках TERMINATE, но они не оказывают воздействия на счетчик завершения. В момент модельного времени 600 EMB завершающий транзакт выходит из блока GENERATE 600 и сразу же переходит в следующий блок TERMINATE 1. Поскольку операнд A блока содержит 1, то из счетчика завершения вычитается 1. Командой START 1 задано содержимое счетчика завершения, равное 1. Поэтому при удалении транзакта блоком TERMINATE 1 со счетчика списывается 1, счетчик обнуляется и процесс моделирования останавливается в момент времени, равный 600.

Предположим, что в команде START было указано число 10, т. е. десять прогонов модели, и в счетчик завершения записано число 10: TG1 =

10 . После первого вычитания содержимое ячейки  $TG1 = 9$ , т. е. не равно нулю. Поэтому моделирование продолжается. После десяти прогонов, т. е. вычитания десяти единиц,  $TG1 = 0$ . Планировщик прекратит моделирование после десяти отрезков времени, заданных блоком GENERATE в сегменте завершения.

Команда START используется для запуска процесса моделирования. Она имеет следующий формат записи:

**START    A,[B],C,[D]**

Операнд A задает значение счетчика завершения, определяющего момент окончания прогона модели. Может быть только целым положительным числом. Операнд B – операнд вывода статистики. Он может быть NP ("нет вывода данных") либо опущен. При задании NP стандартный отчет не выводится. По умолчанию выводится стандартный отчет. Операнд C не используется и сохранен для совместимости с описаниями ранних версий GPSS. Операнд D определяет необходимость вывода содержимого списков событий. Если операнд D указать любым положительным целым числом, например, 1, то списки текущих и будущих событий включаются в стандартный отчет и выводятся. Если операнд D опущен, то по умолчанию списки в стандартном отчете не выводятся.

Команду START можно сразу указывать в конце программы модели при ее подготовке. Тогда после трансляции модели, т. е. создания объекта "процесс моделирования", сразу начинается моделирование. Этот же оператор можно вводить в программу модели в интерактивном режиме. Однако может возникнуть необходимость завершить моделирование не по истечении какого-то времени, а после обработки определенного количества транзактов, имитирующих, например, изготовленные детали, переданные по каналу связи сообщения и т. д. В этом случае сегмент задания времени моделирования не нужен.

Для организации такого способа завершения моделирования необходимо сделать следующее. В блоках TERMINATE, которые выводят из модели транзакты, имеющие смысл тех же изготовленных деталей или переданных сообщений, указать число, на которое уменьшается счетчик завершения моделирования. В команде START также указать число, деление которого на указанное в блоке TERMINATE число даст требуемое количество изготовленных деталей или переданных сообщений. Например, требуется завершить моделирование после изготовления 100 деталей. В модели это может быть так:

```
...  
TERMINATE      1  
...  
TERMINATE      1  
...  
TERMINATE  
START      100
```

Блоков TERMINATE, которые выводят из модели транзакты, соответствующие изготовленным деталям, может быть несколько. Все эти блоки должны иметь 1 в качестве операнда A. У остальных блоков TERMINATE, если они есть в модели, операнд A должен быть опущен.

### ***Задание 5.3***

*1. Напишите сегмент модели, который моделирует поступление 14 транзактов с равномерным законом распределения (14-32);*

*2. Смоделируйте приход в нулевой момент времени пяти транзактов с уровнем приоритета 25;*

*3. Смоделируйте ситуацию, когда транзакты поступают в модель каждые 30-60 минут (первый появляется на 20-й минуте). Единица модельного времени 0,1 минута.*

*4. Напишите сегмент модели, который моделирует поступление 50 транзактов с интервалом 10 – 20 миллисекунд с приоритетом 2 через 2 секунды после начала моделирования.*

5. Напишите сегмент модели, в котором моделирование заканчивается после того, как через модель пройдут 300 транзактов, которые должны поступать в модель каждые 21-29 ед. модельного времени;

6. Напишите сегмент модели, в котором моделирование заканчивается по истечении времени моделирования работы системы 8 часов (единица модельного времени – 1 секунда).

7. Напишите сегмент модели, в котором генерируется 5 транзактов с момента времени 350 со средним 100 и полуинтервалом 60 ЕМВ, в параметр “Name” впишите собственные имя и фамилию, в параметр “Number” добавьте порядковый номер по списку, в параметр “YesNo” – значение булевской переменной “OnOut” (см. префиксы объектов вычислительной категории).

7. Смоделируйте поступление транзактов в течение 4 часов с интервалом 5,5 – 7,7 секунды, каждому присвоить приоритет, равный порядковому номеру транзакта.

## ГЛАВА 6. Числовые данные и объекты вычислительной категории

Прежде, чем говорить об объектах GPSS следует указать, что обращение к именам тех или иных объектов в теле модели выполняется с употреблением следующих префиксов:

V\$Name – обращение к переменной пользователя с именем Name.

FN\$Name – значение функции с именем Name.

BV\$Name – значение булевой переменной с именем Name (0 или 1).

X\$Name – обращение к значению, записанному в ячейку с именем Name.

P\$Name – обращение к параметру транзакта.

### 6.1. Стандартные числовые атрибуты

Стандартные числовые атрибуты (СЧА) объектов GPSS – это числовые (текстовые, булевские) характеристики, присущие объекту. Например, СЧА переменной есть конкретное числовое значение, вычисляемое в момент обращения (вызова) данной переменной. Т.е. при упоминании переменной в тексте модели происходит вычисление ее значения и это значение применяется для той цели, для которой переменная была вызвана. СЧА устройств характеризуют количество транзактов, вошедших в устройство на текущий момент модельного времени, коэффициент занятости устройства и другие параметры.

К отдельной категории следует отнести числовые атрибуты, характеризующие систему в целом. Такие атрибуты будем называть системными, к ним относятся:

AC1 (C1) – абсолютное время модели;

TG1 – величина счетчика завершения;

Кроме перечисленных иногда к системным атрибутам относят номер активного транзакта (транзакта, обрабатываемого системой в текущий момент времени) XN1, и время пребывания активного транзакта в системе

M1. Однако их можно отнести и к категории атрибутов транзактов, именно так мы и будем поступать.

СЧА могут использоваться в качестве операндов практически в любом типе блоков.

#### **Пример 4.1**

```
GENERATE      10,5  
ASSIGN   ServisTime,20  
...  
GENERATE      150,30  
ASSIGN   ServisTime,250  
...  
ADVANCE  P$ServisTime
```

Здесь два разнородных потока транзактов попадают в один блок задержки, но время их задержки определяется значением, записанным в параметр транзакта с именем ServisTime.

Если конкретный операнд конкретного блока имеет ограничения на вид используемых в нем СЧА, то при его вводе в текстовом редакторе GPSS осуществляется автоматический контроль допустимости СЧА. Кроме того, при вводе СЧА можно обратиться к контекстной помощи (клавиша «?»), которая покажет перечень всех СЧА, допустимых для данного операнда.

### **6.2. Параметры транзактов. Изменение приоритета**

В процессе моделирования в любой точке модели пользователь может дополнить структуру данных транзакта любыми числовыми, текстовыми и булевыми величинами путем записи данных в параметр транзакта блоком

```
ASSIGN  A,B,[C]
```

Операнд А определяет имя (номер) параметра транзакта, которому присваивается новое значение. Если такого параметра не существовало, в этот момент он создается. Операнд В – значение, которое записывается в указанный параметр. Если первый операнд используется в формате А+ или А-, то к существующему значению операнда А плюсуется или вычитается из него величина, заданная операндом В.

Операнд С задает имя (номер) функции-модификатора, при этом значение операнда В умножается на значение операнда С.

Примеры записи блока ASSIGN:

**ASSIGN Power,54.3** ;параметру “Power” присваивается 54.3

**ASSIGN Line+,Q\$ATS3** ;к значению параметра “Line” прибавляется значение длины текущей очереди ATS3;

**ASSIGN Param\_3-,5.85,Seven** ;из значения параметра Param\_3 вычитается произведение 5.85 на вычисленное значение функции “Seven”;

**ASSIGN Name,"Prokofiy"** ;параметру с именем Name присваивается строка “Prokofiy”

**ASSIGN Tr1-,(Normal(2,10,5),Function\_2** ;вычисляются выражение в скобках и функция с именем “Function\_2”, перемножаются и полученное произведение вычитается из значения параметра с именем Tr1

Параметры транзактов могут также изменяться блоками PLUS, INDEX, MARK. С помощью блока MARK A в параметр транзакта с именем, заданным в операнде A, вписывается текущее абсолютное время модели. В некоторой точке модели после этого можно обратиться к СЧА MP\$A, чтобы определить время прохождения транзактом участка модели между блоком MARK и блоком, в котором пользователь обратился к СЧА MP\$A.

Примеры использования блока MARK:

...  
**MARK StartTime**

...  
**ASSIGN Period,MP\$StartTime**

...  
В некоторой точке модели (начало участка) в параметр “StartTime” вписано текущее время, а в другой точке модели (конец участка модели) вычислена длительность прохождения участка модели транзактом и вписана в параметр “Period”.

Приоритет транзакта в процессе его продвижения по модели может

быть изменен блоком **PRIORITY N**, где N – новое значение приоритета.

*Примечание 1. В тех случаях, когда операндом блока может служить только параметр транзакта, префикс P\$ не используется. Это относится к блокам ASSIGN, PLUS, INDEX, MARK и блоку задания цикла LOOP (будет рассмотрен позже).*

*Примечание 2. В случаях, когда используются не имена объектов, а присвоенные им номера, знак \$ в префиксе не используется. Соотнести имя и номер объекта можно с помощью команды EQU (эквивалентно) в сегменте описаний:*

**MyName EQU 55**

### **Задание 5.2**

*Присвоить параметру транзакта с именем “StudentNum” свой номер по списку группы.*

*Увеличить этот номер на три единицы.*

*Вписать в параметр с именем “LocTime” текущее время.*

*Вписать в параметр “Period” время от момента записи “LocTime”.*

### **Атрибуты транзактов**

P\$Name – параметр транзакта.

M1 – время нахождения активного транзакта в модели от момента его выхода из блока GENERATE.

MP\$Name – время прохождения участка модели активным транзактом. Вычисляется по формуле  $MP\$Name = (\text{текущее время}) - P\$Name$ . Для использования СЧА MP\$Name нужно в P\$Name предварительно занести отметку времени блоком MARK Name.

### **Атрибуты блоков**

N\$Name – число входов транзактов в блок с именем Name.

W\$Name – число транзактов, находящихся в блоке с именем Name.

### **Атрибуты функций и переменных**

FN\$Name – значение функции с именем Name.

V\$Name – значение арифметической переменной.

BV\$Name – значение булевой переменной с именем Name (0 или 1).

Полный перечень числовых атрибутов приведен в Приложении 1.

### 6.3. Арифметические константы и переменные

Формат задания числовых констант в модели:

Name        **EQU** A

Команда EQU задает константе с именем Name числовое значение A.

Обычно эта команда используется для задания набора числовых начальных данных, но может выполнять и процедуру замены имени объекта GPSS на целочисленный номер.

Многие числовые данные и арифметические выражения в модели можно задать с помощью арифметических переменных. Переменные обязательно должны быть описаны пользователем в сегменте описаний. Примеры описания переменных:

**Freq VARIABLE        134.2** ;переменной с именем Freq задано постоянное значение 134,2;

**Intens VARIABLE        (V\$Freq-17,8)^2**        ;переменная Intens вычисляется как квадрат разности переменной Freq и константы 17,8.

Вызов значения переменной (ее СЧА) производится с использованием префикса V\$ (V\$Freq, V\$Intens). СЧА переменной может использоваться в качестве операнда в большинстве блоков.

Описание арифметической переменной осуществляется в строке описания до ее первого использования, как правило, в сегменте описаний. В одной строке описывается одна переменная. В поле метки обязательно записывается имя описываемой переменной, в поле операции записывается слово VARIABLE, а в поле операндов - арифметическое выражение, составленное из СЧА, знаков арифметических операций и круглых скобок.

В арифметическом выражении можно использовать следующие знаки арифметических операций, список в порядке приоритета выполнения:

$\wedge$  - возведение в степень;

$\#$  - умножение;

$/$  - деление;

$\backslash$  - целочисленное деление (остаток от деления отбрасывается);

$@$  - остаток от деления,

$+$  - сложение;

$-$  - вычитание.

Пусть, например, переменная V\$Sum описана строкой

**Sum VARIABLE S1+S2**

,тогда значение V\$Sum будет определяться как суммарное содержимое первого и второго многоканальных устройств. Если переменную V\$Crf описать в виде

**Crf VARIABLE R\$Post\_2+S\$Post\_4**

,то её значением будет сумма числа свободных каналов МКУ Post\_2 и числа занятых каналов МКУ Post\_4.

Как при описании переменных, так и в операндах блоков можно использовать библиотечные математические функции: ABS(A); ATN(A) – арктангенс A; COS(A); EXP(A); INT(A) – целая часть числа A; EXP(A); LOG(A) – натуральный логарифм; SIN(A); SQR(A); TAN(A). В качестве аргумента функций (A) могут использоваться константы или СЧА.

При описании арифметических переменных необходимо иметь в виду следующее:

1) Равноприоритетные операции в арифметическом выражении при отсутствии скобок выполняются слева направо. Операции  $\#$ ,  $/$ ,  $@$  имеют приоритет по отношению к операциям  $+$ ,  $-$ .

2) Синтаксис арифметических выражений соответствует синтаксису,

принятому в языках программирования высокого уровня. Например, аргументы любых функций должны заключаться в скобки.

### ***Задание 7.1***

*Описать переменные:*

1)  $R = 3,2;$

2)  $n = 2;$

3)  $K = R * \ln(2,74+n);$

4)  $D = 1000 * K;$

5)  $Z = 12 * 250 + 400;$

5)  $P = D - Z$

С помощью переменных можно задать изменяющуюся во времени интенсивность потока транзактов или меняющуюся интенсивность обслуживания. Для этого достаточно алгебраически задать закон изменения интервалов времени между появлениями транзактов или времени обслуживания.

### **Пример 6.2**

Интенсивность потока автомобилей в течение одной смены меняется по квадратичному закону от минимального значения 10/час до максимального 90/час к середине смены (длительность смены 8 часов) и падает опять до начального значения к концу смены. Смоделировать поток автомобилей на пост, считая его пуассоновским.

#### **Решение.**

Закон изменения интенсивности представляет собой «перевернутую» параболу, проходящую через точки с координатами (0,10); (4,90); (8,10).

Общее выражение для параболы:  $y = ax^2 + bx + c$ . Подставляя координаты точек в это выражение, получим три уравнения для трех неизвестных коэффициентов:

$$a * 0 + b * 0 + c = 10$$

$$a*4^2 + b*4 + c = 90$$

$$a*8^2 + b*8 + c = 10$$

Решая эту систему относительно  $a$ ,  $b$  и  $c$ , получим значения коэффициентов и можем записать закон изменения интенсивности:

$$I = -5t^2 + 40t + 10$$

Теперь можно описать переменные  $Int$  и  $Tau$ , вычисляющие интенсивность потока и интервалы времени между приходами транзактов. СЧА переменной  $V\$Tau$  используем как параметр библиотечной экспоненциальной функции распределения и сгенерируем пуассоновский поток транзактов (в минутах):

```
;Сегмент описаний. EMB==1 мин.  
Time VARIABLE C1/60  
Int VARIABLE -5#V$Time^2+40#V$Time+10  
Tau VARIABLE 60/V$Int
```

```
;Сегмент генерации  
GENERATE (Exponential(1,0,V$Tau))
```

## **Задание 7.2**

*Время беседы консультанта с клиентом (в минутах) зависит от сложности вопроса следующим образом:*

$$T = 5*\ln(2,718 + n) \pm 3 \text{ (мин)}.$$

*Имеется пять уровней сложности “n” от 1 до 5. Задать уровень сложности в переменной, сгенерировать равномерно распределенный поток клиентов с интервалами времени 20-40 минут. Уровень сложности вписать в параметр транзакта как целочисленную равномерно распределенную величину. Смоделировать работу консультанта в течение 8 часов.*

## **6.4. Логические (булевы) переменные**

Обозначение булевой переменной имеет вид **BV\$name**. Этот СЧА, как и арифметическая переменная, описывается пользователем в сегменте

описаний модели. Булева переменная описывается в строке описания переменной. В одной строке описывается одна переменная. В поле метки обязательно записывается имя переменной, в поле операций - слово BVARIABLE, а в поле операндов - булево выражение. Булево выражение составляется из элементов пяти типов: СЧА, отношений, стандартных логических атрибутов (СЧА), знаков логических операций и скобок. Значением булевой переменной является число 1, когда булево выражение истинно и число 0 - в противном случае. Подчеркнем, что значением булевой переменной являются именно числа.

Отношение представляет собой запись двух СЧА, соединенных одним из знаков:

‘G’ - больше чем,

‘L’ - меньше чем,

‘E’ - равно,

‘NE’ - не равно,

‘LE’ - меньше или равно,

‘GE’ - больше или равно.

Отношение имеет значение «истинно» в том и только в том случае, если значения входящих в него СЧА отвечают смыслу соединяющего знака. Например, переменные BV\$Log1 и BV\$Log2 , описанные строками:

**Log1 BVARIABLE V\$Arg1’G’5**

**Log2 BVARIABLE V\$Arg1’E’V\$Arg2,**

равны единице, если СЧА переменной V\$Arg1 > 5 и V\$Arg1 = V\$Arg2.

Обращение к логической переменной производится с префиксом BV\$Name.

Если СЧА в логическом выражении не входит в отношение, то ему приписывается логическое значение (в этом случае СЧА должен быть заключен в скобки). Логическое значение «ложно» приписывается в случае,

если численно СЧА равен нулю, во всех остальных случаях приписывается значение «истинно».

Стандартные логические атрибуты (СЛА) в GPSS характеризуют состояние одноканальных и многоканальных устройств, а также логических ключей в модели. Список стандартных логических атрибутов устройств и логических ключей приведен в Приложении 1.

СЛА имеет значение «истинно» тогда и только тогда, когда выполняются соответствующие ему условие - в приведенном перечне оно записано напротив вида индикатора. Указатель  $j$  в обозначении индикатора задается так же, как и в обозначении СЧА, т.е. может быть номером, косвенным номером или именем объекта.

Знаки логических операций в описании булевой переменной - это знак логического сложения (+) и знак логического умножения (#). Операция умножения имеет приоритет по отношению к операции сложения. Например, булева переменная BV\$Resurs, описанная строкой:

**Resurs      BVARIABLE      FNU1#FNU2+SNF1#SNF2,**

равна единице тогда и только тогда, когда одноканальные устройства 1 и 2 оба свободны, или МКУ 1 и 2 обе не заполнены.

Следует иметь в виду, что, хотя все промежуточные операции в булевой переменной имеют логические значения «истинно» или «ложно», сама булева переменная имеет числовое значение 0 или 1.

### ***Задание 6.3***

*Описать булевы переменные:*

- 1)  $E =$  «истина», когда коэффициент (из задания 3.1)  $K \geq 1,3$ ;*
- 2)  $G = 0$ , когда  $D < Z$ ;*
- 3)  $NS =$  «истина», когда номер активного транзакта больше или равен 100;*
- 4) Булева переменная равна 1, если текущая длина очереди Line5*

меньше или равна значения, записанного в параметр активного транзакта *ContrQ*.

### **6.5. Ячейки и матрицы ячеек**

Для того чтобы в модели организовать запоминание каких-либо данных и вывод их в отчет, в языке GPSS имеются специальные СЧА, называемые ячейками сохраняемых величин или ячейками **SAVEVALUE**. СЧА ячейки имеет префикс вида  $X_j$ , где  $j$  – номер ячейки, или  $X\$Name$ , где *Name* – имя ячейки.

Запись значений в ячейку осуществляется при входе транзакта в блок **SAVEVALUE A,B**

Поле *A* этого блока интерпретируются как номер или имя ячейки, поле *B* – как записываемое в нее значение.

Если в поле *A* записан СЧА, то его значение рассматривается как номер ячейки.

Пусть, например, транзакт проходит в модели через следующую цепочку блоков:

**SAVEVALUE Oper,157**

**SAVEVALUE 1,X\$Oper**

**SAVEVALUE X1,C1**

Тогда будут выполняться следующие действия:

запись числа 157 в ячейку  $X\$Oper$ ;

запись числа 157 в ячейку 1;

запись текущего времени в ячейку номер 157.

В начальный момент времени, когда в ячейки еще ничего не записывалось, все они имеют нулевое значение.

Если к обозначению СЧА в поле *A* блока **SAVEVALUE** приписать знак + или -, то в ячейке вместо записи значения будет производиться добавление или, соответственно, вычитание этого значения. Например,

прохождение транзактом блоков

**SAVEVALUE SV1+,S\$Mem**

**SAVEVALUE Got-,V\$Diff**

вызовет добавление числа занятых единиц МКУ с именем Mem в ячейку SV1 и вычитание СЧА переменной Diff из ячейки X\$Got. Поскольку в качестве операнда A блока SAVEVALUE может быть только имя сохраняемой ячейки, префикс X\$ в операнде A не используется. Но при обращении к СЧА (значению) сохраняемой ячейки в любом другом месте модели обязательно использование префикса X\$.

В качестве операнда B может быть арифметическое выражение, в этом случае, как и всегда при вычислениях в теле модели, оно берется в скобки:

**SAVEVALUE Rez,(V\$Ain+X\$Zvay^2/P\$Dray)**

Здесь в сохраняемую ячейку Rez записывается результат вычисления выражения в скобках, где упоминаются переменная Ain, сохраняемая ячейка Zvai и параметр транзакта Drai.

Наряду с ячейками в моделях на языке GPSS можно использовать матрицы ячеек. В отличие от простых ячеек матрицы перед использованием должны быть описаны. Для описания матрицы применяется строка описания матрицы. В поле метки этой строки записывается имя описываемой матрицы, в поле операции - слово MATRIX, в поле операндов - параметры матрицы: в поле A записывают любое слово или оставляют поле пустым, в поле B указывают число строк матрицы, в поле C - число столбцов. Например, описать матрицу ИТОГИ размерами 2\*4 можно с помощью строки:

**ИТОГИ MATRIX 2,4**

В начальный момент любая матрица содержит только нулевые значения.

После того как матрицы описаны, можно использовать в качестве

СЧА индексированные переменные - ячейки этих матриц. Ячейка матрицы имеет обозначение вида **MX\$name(a,b)**, где name – имя матрицы, a - СЧА, значение которого задает номер строки матрицы, b - СЧА, задающий номер столбца. Например, обозначение **MX\$ITOGI(1,2)** соответствует ячейке в первой строке и втором столбце матрицы ITOGI.

Запись значений в ячейки матриц осуществляется с помощью блоков **MSAVEVALUE**. Поле A этого блока задает имя матрицы, поле B - номер строки, поле C - номер столбца, поле D - записываемое значение.

Например, при входе транзакта в блок

**MSAVEVALUE Itogi,2,4,V\$Abs**

в матрицу ITOGI, в ячейку, расположенную на пересечении второй строки и четвертого столбца, записывается значение переменной V\$Abs.

В поле A блока **MSAVEVALUE** после имени матрицы может быть приписан знак + или -. В этом случае значение операнда D добавляется или соответственно вычитается в ячейке матрицы, аналогично тому, как это делается в блоке **SAVEVALUE**.

#### **Задание начальных значений ячеек и матриц**

Начальные значения ячеек можно сделать отличными от нуля. Для этого применяется строка описания ячеек **INITIAL**. В поле A этой строки указывается СЧА, обозначающий ячейку или ячейку матрицы, в поле B – задаваемое начальное значение.

Например, строка

**INITIAL X\$Sever,-10**

установит в качестве начального значения ячейки с именем Sever число -10.

#### **Задание 6.4**

*Подсчитать количество транзактов, вышедших из модели и записать это значение в сохраняемой ячейке. Сделать двумя способами: учитывая каждый транзакт, выходящий из модели или учитывая номер*

очередного транзакта. Значения СЧА всех переменных из задания 4.1 также сохранить в СЯ.

### **Задание 6.5**

Вывести в отчет (записать в СЯ) количество транзактов, номера которых кратны 3. Воспользоваться для этого булевой переменной и операцией определения остатка от деления.

### **6.6. Функции**

Функция — это объект вычислительной категории, СЧА которого обозначается в виде **FN\$name**. Функция описывается пользователем в виде численной зависимости от аргумента, обозначенного каким-либо СЧА. Формат описания функции таков:

**MyFunctionD**     **FUNCTION Arg,Dn** – для дискретной функции;  
**MyFunctionC**     **FUNCTION Arg,Cn** – для непрерывной функции.

Здесь «MyFunctionD» и «MyFunctionC» – имена соответственно дискретной и непрерывной функций; «Arg» – СЧА, используемый в качестве аргумента функции; «D» – обозначение дискретной и «C» – непрерывной функции; «n» – количество точек, которыми задается функциональная зависимость.

Обращение к функции в теле модели означает вычисление ее значения при текущем значении аргумента. Обращение к функции производится с использованием префикса FN\$name или FNj.

Рассмотрим на примерах, как это делается.

Пусть необходимо, чтобы значение FN\$KL1 вычислялось в модели через аргумент S\$MEM в соответствии с графиком, изображенным на приведенных ниже рисунках. Аргументом функции в данном случае является текущее количество занятых каналов в МКУ с именем MEM.

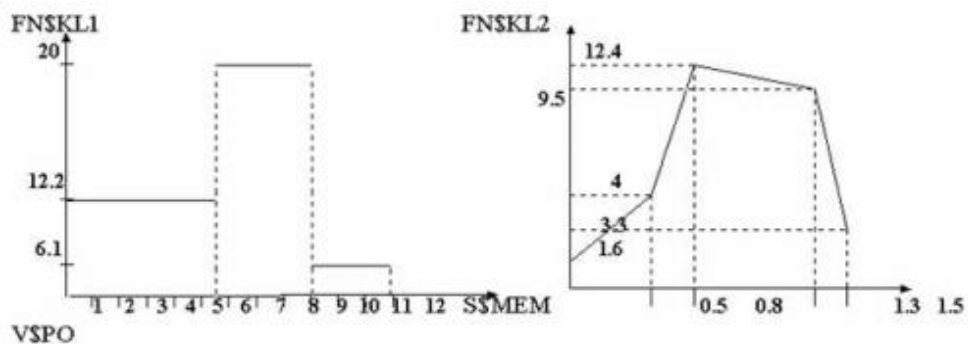


Рис. 7.1. Графики дискретной и непрерывной функций

Функция, вычисляемая в соответствии с таким кусочно-постоянным графиком, является дискретной. Она описывается следующими строками:

```
KL1 FUNCTION      S$MEM,D3  
5.5,12.2,/9,20/12,6.1
```

Первая строка описания функции должна содержать имя (метку) функции в поле метки, слово FUNCTION в поле операции и два операнда в поле операндов. Первый операнд является обозначением того СЧА, который выбран в качестве аргумента функции. В данном примере это S\$MEM – количество занятых единиц памяти MEM. Второй операнд состоит из буквы D, обозначающей, что данная функция дискретная, и из числа точек, которыми задается график функции. Координаты точек, задающих график, записываются в последующих строках описания функции. Из примера видно, что каждая точка задается парой координат, разделенных запятой, причем вначале указывается абсцисса точки, потом ордината. Координаты разных точек разделяются косой чертой. Функция может задаваться любым числом точек. Запись координат точек начинается с 1-й позиции строки и может быть продолжена на последующих строках.

Перенос записи координат на следующую строку следует производить, начиная с новой пары координат. При этом ту пару координат, которая оказывается на строке последней, не обязательно заканчивать разделяющей косой чертой.

Абсциссы точек графика, указываемые в строках описания функции, должны быть все упорядочены по возрастанию.

Следует обратить внимание на то, что в описании дискретной функции задаются только те точки, которые расположены на правых концах горизонтальных отрезков.

График функции FN\$KL2, изображенный на рис. 7.1, получен путем соединения нескольких точек ломаной линией. Функция, задаваемая таким графиком, называется непрерывной (фактически она является кусочно-непрерывной). Функция FN\$KL2 в данном случае описывается строками:

```
KL2 FUNCTION      V$Poi,C5  
0,1.6/0.5,4/0.8,12.4/1.3,9.5/1.5,3.3
```

Как видим, отличия описания непрерывной функции от дискретной состоят в том, что во-первых, в первой строке вместо признака D дискретной функции должен быть признак C непрерывной функции; а во-вторых, первой парой чисел являются координаты самой левой точки графика.

Если при вычислении непрерывной функции значение ее аргумента выходит за диапазон, заданный в графике, то график автоматически «продолжается» горизонтальными прямыми: влево - на уровне первой заданной точки графика, вправо – на уровне последней.

### **Пример 6.3**

Чем длиннее очередь к устройству обслуживания, тем меньше желания очередного клиента стоять в этой очереди. Опишем вероятность отказа от услуги в зависимости от длины очереди с именем Ochered дискретной функцией. Допустим, эта вероятность приближенно описывается следующим образом: при длине очереди до 3 человек вероятность ухода клиента равна 0,1, при длине очереди до 10 человек вероятность увеличивается до 0,5; до 15 – 0,8; более 15 – равна 1. Для того, чтобы задать последнюю вероятность выберем любое число больше 15,

например 20, для всех значений больше 20 дискретная функция сохранит значение 1. Такая дискретная функция описывается следующим образом:

**VerUhoda FUNCTION Q\$Ochered,D4**  
**3,.1/10,.5/15,.8/20,1**

Если необходимо вычислять такую вероятность более точно, функцию можно описать как непрерывную, список пополнится левой точкой:

**VerUhoda FUNCTION Q\$Ochered,C5**  
**0,0/3,.1/10,.5/15,.8/20,1**

Важным частным случаем рассмотренных видов функций являются функции с аргументом RNj. Если аргументом функции выступает RNj, то подразумевается, что в качестве аргумента функции используется псевдослучайное число, заключенное в диапазоне от 0 до 0,999999 включительно, вычисляемое с помощью j-го генератора случайных чисел. Когда аргументом функции является случайное число, то функция также будет принимать случайные значения. Функция от аргумента RNj используется для моделирования случайных величин, имеющих требуемые законы распределения вероятностей. Например, дискретная функция, описанная строками:

**DIS FUNCTION RN1,D4**  
**0.1,1/0.3,5/0.4,7/1.0,8**

будет принимать значение 1 с вероятностью 0,1, значение 5 с вероятностью 0,2, значение 7 с вероятностью 0,1 и значение 8 с вероятностью 0,6.

Действительно, график функции FN\$DIS имеет вид, показанный на рис. 7.2. Поэтому функция FN\$DIS примет при ее очередном вычислении значение 1, если RN1 попадет в интервал от 0 до 0,1. Вероятность этого равна 0,1. Значение 5 для FN10 будет получено с вероятностью попадания RN1 в интервал от 0,1 до 0,3. Эта вероятность равна  $0,3 - 0,1 = 0,2$ . Аналогично определяются вероятности значений функции, равные 7 и 8.

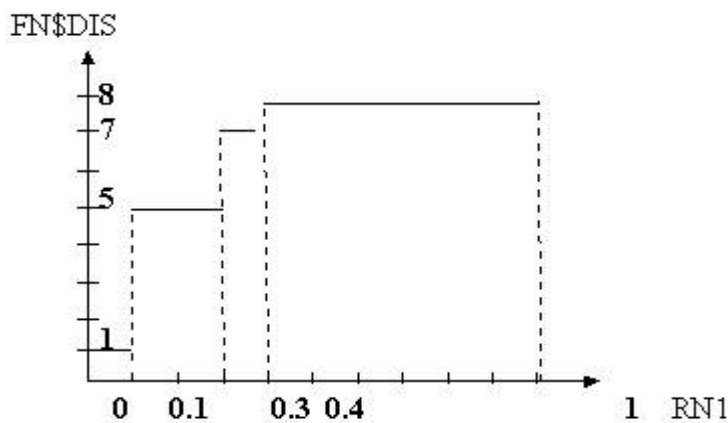


Рис. 7.2. График случайной функции

В GPSS World содержится несколько библиотечных функций задания неравномерного распределения, которые можно использовать в качестве операндов блоков, например:

**GENERATE (Exponential(1,0,65))** ;генерируется поток транзактов с экспоненциальным распределением интервалов прихода со средним 65;

**ADVANCE (Uniform(2,10,30))** ;производится задержка транзакта на случайное равномерно распределенное время от 10 до 30 ЕМВ;

**SAVEVALUE Norm,(Normal(3,25,10))** ;в ячейку Norm, записывается случайная нормально распределенная величина со средним 25 и СКО=10;

**ASSIGN Rating+,(DUniform(4,0,20))** ;в параметр транзакта с именем Rating плюсуется балл, начисляемый за сдачу зачета – случайное целое равномерно распределенное от 0 до 20 число.

### **Задание 6.6**

*Задать дискретную и непрерывную функции зависимости относительной интенсивности автомобильного движения в течение суток. Относительная интенсивность:*

*00.00 до 04.00 – 0,1*

*04.00 до 06.00 – 0.3*

06.00 до 08.00 – 0,6

08.00 до 10.00 – 1,0

10.00 до 16.00 – 0,7

16.00 до 20.00 – 1,0

20.00 до 22.00 – 0,8

22.00 до 24.00 – 0,2

### **Задание 6.7**

Задать непрерывной функцией закон распределения случайной величины:

Аргумент	Значение функции
0	0
0,2	2
0,4	5
0,6	3
1,0	1

### **Задание 6.8**

Задать дискретную случайную функцию (аргумент – СЧА генератора случайных чисел  $RN_j$ ), в которой функция принимает значение 7 при значениях аргумента от 0 до 0,5 и значение 10 при значениях аргумента от 0,5 до 1,0.

### **Задание 6.9**

Максимальную интенсивность подхода машин на таможенный пост принимаем за 1. В течение суток эта интенсивность меняется следующим образом: с 0 до 7.00 – 0,1; с 7.00 до 10.00 – возрастает до 1; с 19.00 до 22.00 – падает до 0,5; и до 24.00 – падает до 0,1. Описать интенсивность непрерывной функцией, зависящей от времени суток. Время суток описать переменной, вычисляемой как остаток от деления текущего времени на длину суток в минутах.

В течение недели интенсивность потока меняется следующим

*образом: ПН – 1; ВТ – 0,8; СР – 0,7; ЧТ – 0,6; ПТ – 0,8; СБ – 1; ВС – 0,5.*

*Сгенерировать поток транзактов, имитирующих подход машин, максимальная интенсивность которого составляет 90 машин в час.*

.

## ГЛАВА 7. Моделирование систем

### 7.1. Задержка транзактов в блоках ADVANCE

Блок ADVANCE A,[B] (задержать) предназначен для задержки транзактов на заданные разработчиком модели интервалы модельного времени. Тем самым имитируется обслуживание заявки в устройстве в течение определенного времени или задержка транзакта по какой-либо другой причине (ознакомление покупателя с товаром, изучение правил декларирования товаров и др.)

Обязательный операнд A задает время задержки транзакта в блоке ADVANCE. Необязательный операнд B является модификатором-функцией или модификатором-интервалом. Значение операнда B используется здесь для модификации значения операнда A так же, как в блоке GENERATE.

Любой транзакт входит в блок ADVANCE беспрепятственно. В нем транзакт задерживается на период модельного времени, величина которого определяется операндами A и B. После этого транзакт направляется из блока ADVANCE к следующему блоку.

В поле A задержка транзакта может быть задана нулевой. В этом случае транзакт будет задержан в блоке ADVANCE на нулевое время, т.е. момент выхода транзакта из блока будет равен моменту его входа в блок.

Например, в блоке ADVANCE 0 транзакты не будут задерживаться.

Блок ADVANCE 1000 будет задерживать транзакты ровно на 1000 ед. модельного времени.

В блоке ADVANCE 1000,P\$Coock любой транзакт будет задерживаться на случайное время, выбранное из диапазона «1000 ± значение параметра транзакта с именем Coock».

Блок ADVANCE P\$ServiceTime,P\$Half\_Int задержит транзакты на случайное, равномерно распределенное время. Среднее значение и полуинтервал должны быть ранее в модели записаны в параметры транзакта

с именами ServiceTime и Half\_Time блоками ASSIGN:

**ASSIGN ServiceTime,10 ;В параметр записано число 10**

**ASSIGN Half\_Int,3 ;В параметр записано число 3**

Случайное время задержки (обслуживания) может быть задано и с помощью библиотечной или пользовательской случайной функцией. Способы задания таких функций рассмотрены в главе 6.

## **7.2. Операции занятия и освобождения устройств**

Операции занятия и освобождения одноканальных устройств выполняются в блоках SEIZE (занять) и RELEASE (освободить).

При вводе транзакта в блок SEIZE выполняется операция занятия устройства, номер или имя которого задается операндом А блока SEIZE. Номер устройства может быть задан в виде любого СЧА.

Занятие устройства транзактом выполняется следующим образом. Когда транзакт направляется из какого-нибудь блока в блок SEIZE, симулятор проверяет, свободно ли требуемое устройство. Если устройство не свободно, т.е. занято или захвачено, то транзакт не может войти в блок SEIZE. Он остается в предыдущем блоке до тех пор, пока устройство не освободится. Если же устройство свободно, то симулятор без задержки передвигает транзакт в блок SEIZE и устройство переходит в состояние «занято». В этот же момент времени транзакт направляется к следующему за SEIZE блоку.

В последующем движении по модели транзакт может освободить занятое устройство. Освобождение устройства происходит в блоке RELEASE, в поле А которого указывается имя (номер) освобождаемого устройства.

При составлении моделей пользователь должен соблюдать правило: освободить устройство может только тот транзакт, который его занимает. Если транзакт попытается освободить устройство, занятое или захваченное

другим транзактом, симулятор прервет выполнение модели и выдаст сообщение об ошибке.

Если несвободное устройство пытаются занять несколько транзактов с помощью одного или разных блоков SEIZE, то все они до момента освобождения устройства задерживаются в тех блоках, из которых пытались перейти в блок (блоки) SEIZE. Когда транзакты задерживаются на входе блока SEIZE, они регистрируются в списке задержки, который является аналогом очереди и создается при каждом устройстве. В этом списке транзакты упорядочиваются по приоритетам: любой транзакт с более высоким приоритетом помещается впереди любого другого транзакта, имеющего более низкий приоритет. Если же приоритеты у каких-либо транзактов одинаковы, то такие транзакты упорядочиваются между собой по времени прихода: впереди оказывается транзакт, который раньше обратился к устройству. Приоритетное упорядочение является в GPSS основным способом упорядочения задерживаемых транзактов и применяется не только при обращениях к несвободному устройству, но и во многих других случаях. В момент освобождения устройства его занимает первый по списку из задержанных транзакт.

Таким образом, перед занятым устройством выстраивается очередь транзактов, ожидающих обслуживания. Для учета статистики очереди (времени ожидания, длины очереди и др.) используются блоки QUEUE (регистрация входа в очередь) и DEPART (регистрация выхода из очереди). Следует подчеркнуть, что очередь создается независимо от этих блоков, они только учитывают ее статистику.

Постановку в очередь и занятие устройства можно выполнять как однотипными, так и разнотипными транзактами в любых участках модели.

Транзакт может занимать любое число устройств. Освобождать занятые устройства транзакт может в любом порядке.

### Пример 7.1.

Рассмотрим оформление двух типов товаров, один из которых оформляется в течение 6-10 минут, другой – в течение 8-14 минут (равномерные законы распределения). Время оформления второго типа товара впишем в параметр транзакта. Очередь общая, но у товаров второго вида приоритет выше. Интервалы поступления заявок случайные, означены в блоках GENERATE.

Ограничение времени моделирования определяется сегментом завершения через 24 часа.

*;EMB = 1 минута*

*;Сегмент декларирования товаров первого типа*

GENERATE	14,6	;Поток заявок первого типа
QUEUE	Q_Custom	;Регистрация входа в очередь
SEIZE	Custom	;Занятие устройства
DEPART	Q_Custom	;Выход из очереди
ADVANCE	8,2	;Обслуживание, время указано ;непосредственно в блоке
RELEASE	Custom	;Освобождение устройства
TERMINATE		;Удаление транзакта

*;Сегмент декларирования товаров второго типа*

GENERATE	27,20,,,2	;Поток заявок второго типа с приоритетом 2
ASSIGN	Time2,11	;Запись среднего времени в параметр ;транзакта Time2
ASSIGN	HTime2,3	;Запись полуинтервала в параметр ;транзакта HTime2
QUEUE	Q_Custom	;Регистрация входа в очередь
SEIZE	Custom	;Занятие устройства
DEPART	Q_Custom	;Выход из очереди
ADVANCE	P\$Time2,P\$HTime2	;Обслуживание, время считывается ;из параметров транзакта
RELEASE	Custom	;Освобождение устройства
TERMINATE		;Удаление транзакта

*;Сегмент завершения*

GENERATE	(24#60)	;Завершающий транзакт через 24 часа
TERMINATE	1	;Удаление транзакта и списание 1 со счетчика ;завершения
START	1	;Запуск модели и задание счетчика завершения

Рассмотрим логику выполнения этой модели. Первые две строки модели являются комментарием и транслятором не воспринимается. Блок GENERATE имитирует приход заявок на оформление товара первого типа, он генерирует транзакты через каждые  $14 \pm 6$  ЕМВ. Каждый транзакт, поступивший из блока GENERATE, попадает в блок QUEUE Q\_Custom, в котором регистрируется время входа в очередь.

Если транзакт направляется в блок SEIZE Custom, в то время как устройство Custom свободно, то он входит в этот блок, занимает устройство, одновременно выходит из очереди, проходя через блок DEPART, переходит в блок ADVANCE 8,2 и задерживается в нем. Таким образом, в течение  $8 \pm 2$  ед. модельного времени устройство Custom будет занято. Транзакты, пытающиеся в это время попасть в блок SEIZE, будут задержаны в очереди и упорядочены по приоритетам. Следовательно, задерживаемые транзакты будут в точности изображать живую очередь.

Из блока ADVANCE 8,2 транзакт переходит в блок RELEASE и освобождает устройство Custom. В этот же момент времени транзакт переходит в блок TERMINATE и уничтожается.

Поток товаров второго вида имитируется вторым блоком GENERATE. Эти товары попадают в ту же очередь и в то же устройство, что и первые, однако, благодаря повышенному приоритету опережают в очереди товары первого вида.

Модель работает до тех пор, пока блоком GENERATE в сегменте завершения не будет сгенерирован один транзакт, который, попав сразу в блок TERMINATE 1, удаляется, списав при этом единицу из счетчика завершения, тем самым обнулив его и остановив работу модели. Результаты моделирования отображаются в файле стандартного отчета. Формат этого файла приведен в Приложении 5.

### *Задание 7.1*

*Пусть транзакты имеют одинаковые приоритеты и изображают кинозрителей, а устройство представляет кассира в кассе кинотеатра.*

*Кинозрители приходят через каждые  $20 \pm 10$ с, знакомятся с афишей в течение  $25 \pm 5$ с и занимают очередь в кассу. Каждый кинозритель приобретает у кассира билеты в течение  $20 \pm 5$ с, в зависимости от числа покупаемых билетов и необходимости получения сдачи. Общее число мест в кинозале и, соответственно, билетов – 200. Все пришедшие в кинотеатр решают купить билет.*

### **7.3. Операции блокирования и разблокирования одноканальных устройств**

Транзакты имеют возможность блокировать устройства, т.е. делать их недоступными, не занимая этих устройств. При этом могут задаваться разные режимы управления теми транзактами, которые будут пытаться занять или захватить заблокированное устройство.

Блокирование устройств выполняется с помощью блока FUNAVAIL (устройство недоступно):

#### **FUNAVAIL A [, B, C, D, E, F, G, H]**

у которого поле А обязательно, а поля В – Н могут отсутствовать.

В полях задаются:

А – имя устройства или диапазон устройств;

В – задание режимов RE или CO. Если задано RE, то транзакт, который в момент блокирования занимает (возможно) устройство, с обслуживания снимается и в дальнейшем не претендует на его занятие. В этом случае в поле С задается метка, на которую пересылается снятый с обслуживания транзакт. Если в поле В записано CO, то обслуживаемый транзакт продолжает обслуживаться до конца, но следующий по очереди транзакт в прерванное устройство войти не может;

С – метка блока ;

D – имя параметра транзакта. В режиме удаления (RE) в него записывается время, оставшееся до конца обслуживания в ОКУ;

E – определяет поведение ранее прерванных транзактов на период недоступного состояния устройства. Ранее прерванные транзакты могли образовываться, если на обслуживание поступал транзакт с более высоким приоритетом и правом прерывать обслуживание текущего транзакта. Здесь может быть записано RE или CO: при RE они исключаются из борьбы за устройство и передаются по метке поля F, а при CO продолжают претендовать на дообслуживание после восстановления доступности устройства и его освобождении от более приоритетных транзактов;

F – номер или метка блока;

G – определяет поведение транзактов, которые будут обращаться к устройству в период его недоступности. Запись CO означает, что они будут ожидать устройство в обычном порядке, запись RE – что они исключаются из борьбы и передаются по метке поля H;

H – метка или номер блока.

Разблокирование устройств выполняется блоком FAVAIL (устройство доступно):

**FAVAIL A**

где A – имя, номер или диапазон номеров устройств (например, FAVAIL 4-8).

Имитировать перерывы в работе устройства можно в отдельном сегменте модели, в котором генерируется блокирующий транзакт, происходит его задержка на время недоступности устройства и разблокирование устройства после задержки.

**GENERATE V\$Blok ;Появление блокирующего транзакта в момент,  
;заданный переменной V\$Blok**

**ASSIGN BlockTime,(Uniform(3,10,20))**

**FUNAVAIL Ustr ;Блокировка ОКУ Ustr**

**ADVANCE P\$BlockTime ;Продолжительность блокировки считывается из**

FAVALE Ustr ;параметра транзакта BlockTime, принимающего значения 15+-5)  
;Разблокирование устройства

В момент времени, записанный в переменной Blok, ОКУ с именем Ustr блокируется на время, вписанное в параметр BlockTime блокирующего транзакта. По истечении этого времени устройство открывается для обслуживания.

### **Задание 7.2**

*Смоделировать работу отделения банка с одним оператором и одной кассой.*

*Поток клиентов, желающих открыть, пополнить или закрыть депозит, – равномерный с интервалом  $13 \pm 5$  минут. Оформление договора на открытие депозита проводится оператором за  $12 \pm 5$  минут, после этого клиент проходит в кассу для внесения (или получения) наличных, это занимает  $4 \pm 2$  минуты. Кассир сдает выручку инкассатору между 7 и 8-м часами работы в течение 10-20 минут. Время моделирования 10 часов. Определить средние длины очередей к оператору и в кассу.*

### **Задание 7.3**

*Смоделировать работу отделения банка из задания 7.2, но с учетом того, что каждые  $1,5 \pm 0,5$  часа приходит клиент для выполнения операций по текущему счету, которая занимает  $15 \pm 5$  минут. Каждый клиент затем посещает кассу. Среднее время и полуинтервал операции по текущему счету записать в параметры транзакта.*

*\*Забегая немного вперед, когда будут рассмотрены процедуры маршрутизации транзактов, опишем возможность не повторять всю модель для второго потока. Есть такой блок TRANSFER, который может безусловно перенаправлять транзакты. Это первое. Второе, транзакты НИКОГДА не могут входить в генератор. Поэтому, чтобы слить два*

потока в один, нужно «перепрыгнуть» через генератор, вот так, например:

<b>GENERATE</b>	<b>5,3</b>	<b>;Первый поток</b>
<b>TRANSFER</b>	<b>,Jump</b>	<b>;Безусловная пересылка на метку</b>
<b>Jump</b>		
<b>GENERATE</b>	<b>8,4</b>	<b>;Второй поток</b>
<b>Jump QUEUE</b>	<b>VotSuda</b>	<b>;Оба потока попадают в одну</b>
<b>колею...</b>		
<b>...</b>		<b>;тело модели</b>
<b>TERMINATE</b>		<b>;Удаляем всех без разбора</b>

#### 7.4. Моделирование многоканальных устройств

Совокупность нескольких идентичных параллельно работающих каналов обслуживания заявок называется **многоканальным устройством (МКУ)**. В ряде источников многоканальные устройства называются «память», следуя терминологии разработчика системы. Но мы будем пользоваться термином «МКУ». В МКУ могут обслуживаться несколько транзактов одновременно. Вход транзактов в МКУ возможен всегда, кроме случаев, когда заняты все каналы или МКУ «выключено», т.е. находится в недоступном (заблокированном) состоянии. Каждое МКУ, как и любой другой объект GPSS, должен иметь собственное имя. Имя присваивается командой в сегменте описаний, который располагается первым в модели, до начала имитации:

**Name STORAGE A**

Name – имя МКУ; STORAGE – команда описания МКУ; операнд A – количество каналов устройства.

Занятие и освобождение МКУ имитируется парой блоков:

**ENTER Name,[B]**

**LEAVE Name,[B]**

Name – имя занимаемого МКУ; необязательный операнд В – количество каналов, занимаемых (освобождаемых) при входе (выходе) одного транзакта. Т.е. один транзакт, входя в МКУ, может занимать несколько каналов обслуживания. При отсутствии операнда [В] по умолчанию транзакт занимает только один канал.

Транзакт не обязан освобождать такое же число каналов МКУ, какое занимал. Он может также освобождать МКУ, которую не занимал. Транзакт имеет право занимать и освобождать любое количество МКУ и операции занятия и освобождения могут при этом чередоваться в произвольном порядке.

Имитация обслуживания заявки осуществляется обычным образом, т.е. путем задержки блоком **ADVANCE**.

### Пример 7.2

Рассмотрим работу автомойки, состоящей из трех рабочих мест. Автомобили подъезжают на мойку с интервалами времени распределенными по экспоненциальному закону со средним 10 минут, процесс мойки занимает  $25 \pm 5$  минут. В случае, когда все места заняты, автомобиль ожидает своей очереди. Все автомобили имеют одинаковые приоритеты. Процесс рассмотреть в течение 24 часов.

*;Сегмент описаний. EMB = 1 минута*

Auto\_Shower            STORAGE    3            ;Емкость мойки = 3 места

*;Сегмент имитации*

GENERATE (Exponential(1,0,10))

;Поток а/м

QUEUE    Sh\_Line

;Очередь на мойку

ENTER    Auto\_Shower

;Занятие места на мойке

DEPART   Sh\_Line

;Выход из очереди

ADVANCE   25,5

;Процесс мойки

LEAVE    Auto\_Shower

;Завершение мойки

TERMINATE

;Отъезд с мойки

*;Сегмент завершения*

GENERATE (24#60)

;Завершающий транзакт

TERMINATE 1

;Удаление и списание 1 со счетчика завершения

START    1

;Запуск модели и завершение по времени

#### **Задание 7.4**

*Смоделировать работу склада с двенадцатью терминалами. Разгрузка автопоезда длится от 70 до 110 минут, автопоезд занимает сразу два терминала, интервалы прибытия, распределены по экспоненциальному закону, в среднем каждые 40 минут. Фуры прибывают в среднем каждые 11 минут, занимают один терминал, разгружаются 40-80 минут. Все машины стоят в общей очереди. Работу склада проанализировать в течение года. Склад работает круглосуточно 350 дней в году.*

*Оценить длину очереди на разгрузку и среднее время ожидания в очереди.*

#### **7.5. Операции блокирования и разблокирования многоканальных устройств**

Многоканальные устройства могут также блокироваться и разблокироваться, как и одноканальные. Блокирование МКУ выполняется с помощью блока

**SUNAVAIL A**

где А – имя (номер) МКУ. Во время недоступности МКУ блок ENTER задерживает транзакты, блокируя вход транзактов в МКУ, а блок LEAVE – нет, т.е. процесс обслуживания транзактов, находящихся в МКУ на момент его блокирования, продолжается до завершения и обслуженные транзакты могут свободно выходить из МКУ и двигаться дальше по модели.

Доступность МКУ восстанавливается посредством блока

**SAVAIL A**

в котором операндом А задается имя (номер) МКУ.

Например, если в Примере 7.2 мы хотим симитировать часовой перерыв на обед с 13 до 14 часов (машины, прибывшие в это время, ждут

окончания перерыва, поэтому изменения в поток транзактов-заявок не вносим), можно добавить в модель сегмент блокировки и разблокировки:

### Пример 7.3

*;Сегмент блокировки и разблокировки*

GENERATE „(13#60),1	;Один блокирующий транзакт в 13.00
SUNAVAIL Auto_Shower	;Закрытие мойки
ADVANCE 60	;Мойка закрыта на 60 минут
SAVAIL Auto_Shower	;Открытие мойки после обеда
TERMINATE	;Удаление блокирующего транзакта

Если такой перерыв нужно имитировать ежедневно в течение года, необходимо воспользоваться операцией вычисления остатка от деления по модулю:

### **Задание 7.5.**

*Смоделировать работу парикмахерской с тремя креслами и кассой.*

*Поток клиентов – экспоненциальный со средним временем 25 минут.*

*Стрижка занимает  $30 \pm 10$  минут, расчет с кассой  $2 \pm 1$  минуты, стрижка стоит  $800 \pm 100$  рублей.*

*В среднем каждый час (экспоненциальный закон) приходит клиент на стрижку и окраску волос, окраска волос занимает 60–90 минут. Стоимость окраски составляет  $1000 \pm 200$  рублей.*

*Временные и денежные данные записать в параметры транзактов.*

*Рассчитать баланс парикмахерской за месяц, если месячные платежи на содержание парикмахерской (аренда, коммунальные платежи, зарплата сотрудников, материалы) составляет от 500 до 600 тысяч рублей в месяц (задать дискретной случайной функцией с шагом 20 тыс.)*

*Парикмахерская работает 10 часов в день без выходных. Смоделировать 30 дней работы*

### **Задание 7.6.**

*Смоделировать работу таможенного склада из задания 3.4 с учетом того, что аварии электрооборудования случаются в среднем раз в два дня*

*(экспоненциальный закон), при этом работа всего склада останавливается. На устранение аварии требуется от 10 до 30 минут (равномерный закон). Для расчета этих времен воспользоваться переменными и встроенными случайными функциями  $Exponential(j,0,MO)$  и  $Uniform(j,A,B)$ . Последняя – функция равномерного распределения,  $j$  – номер случайного генератора,  $A$  и  $B$  – левая и правая границы интервала. В штате один дежурный электрик. По статистике очереди и коэффициенту загрузки устройства определить, принесет ли заметный эффект увеличение штата электриков вдвое, если увеличение штата электриков приводит к сокращению времени устранения аварии на 30%?*

### **Задание 7.7**

*Смоделировать работу поста таможенного досмотра большегрузных автомобилей в течение месяца. Пост работает круглосуточно. На первом этапе производится визуальный осмотр двумя таможенниками (два канала обслуживания). Осмотр длится 10-20 минут. После этого машины поступают на технический пост, на котором могут одновременно контролироваться до пяти автомобилей. Контроль автомобилей с продуктовыми грузами длится от 20 до 40 минут, автобусов от 30 до 60 минут, фур с товарами – от 60 до 120 минут. Проверка таможенных документов осуществляется одним специалистом («Контролер») в течение времени, распределенного по нормальному закону со средним 5 и СКО 2 минуты.*

*Автомобили прибывают на досмотр в случайном порядке (т.е. в любой момент времени может поступить любой тип) с интервалами времени, распределенными по экспоненциальному закону, автобусы в среднем каждые 20 минут, рефрижераторы – 10, фуры – 15. (Рекомендация: время обслуживания на техническом посту вписать в параметры транзактов в соответствии с их временем контроля).*

*Контролер делает перерывы в работе каждые 2 часа на 15 минут.  
Таможенники уходят на пересменку каждые 12 часов на 10 минут.*

*Оценить очереди и коэффициенты загрузки.*

### **Задание 7.8**

*Морские суда прибывают в торговый порт для разгрузки. В порту имеется два буксира, обеспечивающие ввод и вывод судов с рейда в порт. Первый тип судов (малого тоннажа) может разгружаться на одном из 6 малых причалов, второй (большого тоннажа) – на одном из трех больших. При штормах, которые случаются в среднем раз в две недели (экспоненциальный закон) и длятся от 2 до 20 часов (использовать Uniform), буксиры не работают, суда ожидают на рейде, работа на причалах продолжается. Каждые 8 часов причалы останавливают работу на полчаса для пересменки персонала, проверки техники и заправки транспорта горючим.*

*Временные параметры приведены в таблице.*

<i>Операции</i>	<i>Тип судна</i>	
	<i>1</i>	<i>2</i>
<i>Интервал прибытия, мин</i>	<i>100-160</i>	<i>330-450</i>
<i>Время ввода в порт, мин</i>	<i>23-37</i>	<i>33-57</i>
<i>Время разгрузки, час</i>	<i>10-14</i>	<i>14-22</i>
<i>Время вывода из порта, мин</i>	<i>15-25</i>	<i>25-45</i>

*Все временные характеристики вписать в параметры транзактов.  
Оценить среднее время ожидания на рейде судами 1 и 2 типов.*

### **7.6. Статистика очередей**

Блок QUEUE может быть помещен перед любым блоком модели, в

котором может возникнуть задержка. Отметим, что очередь к занятому устройству автоматически организуется пакетом моделирования независимо от того, есть в программе блок QUEUE или нет. По очередям в отчете отражается информация: имя или номер очереди(QUEUE), максимальная длина очереди за время моделирования (MAX), минимальная длина очереди (CONT.), число входов в очередь (ENTRIES), число входов в очередь без последующего ожидания - нулевые входы (ENTRIES(0)), средняя длина очереди (AVE.CONT), среднее время пребывания в очереди (AVE.TIME), среднее время пребывания в очереди при учете только ненулевых входов(AVE.(0)). Транзакты в процессе перемещения по модели могут задерживаться перед некоторыми блоками, если условия входа не выполняются. Примерами таких блоков из рассмотренных выше являются блоки:

- **SEIZE** (если ранее вошедший в этот блок транзакт не вошел в блок RELEASE);
- **ENTER** (если требуемая текущим транзактом емкость памяти больше емкости свободного участка данной памяти);
- **GATE** и **TEST** (если в этих блоках не указан альтернативный выход и проверяемое условие не выполняется).

При поступлении транзактов на вход задерживающих блоков образуются очереди. Для сбора статистики об очереди в местах задержки транзактов ставят блок входа в очередь QUEUE A,B, где A – имя или номер очереди, B – число элементов, добавляемых в очередь транзактом, по умолчанию B = 1. При входе в блок QUEUE текущая длина очереди получает приращение на величину, задаваемую полем B. Блок выхода из очереди имеет вид DEPART A, B, где A – имя или номер очереди, B – число элементов, удаляемых из очереди транзактом.

Таким образом, работа очереди в GPSS-модели отображается двумя

блоками:

**QUEUE A,B**

**DEPART A,B**

Очереди имеют следующие СЧА:

$Q\$j$  – текущая длина очереди  $j$ ;

$QM\$j$  – максимальная длина очереди  $j$ ;

$QA\$j$  – средняя длина очереди  $j$ ;

$QC\$j$  – число входов в очередь  $j$ ;

$QZ\$j$  – число входов в очередь с нулевым временем пребывания  
(транзакт прошел через блок QUEUE, не задерживаясь в очереди);

$QT\$j$  – среднее время пребывания в очереди  $j$ , включая нулевые  
входы;

$QXSj$  – среднее время пребывания в очереди  $j$  без нулевых входов.

## Глава 8. Маршрутизация транзактов и построение логической схемы работы модели

Маршрутизация транзактов производится:

- блоком прямого перемещения *DISPLACE*
- блоком проверки условий *TEST*;
- блоком пересылки *TRANSFER*;
- блоком цикла *LOOP*;
- блоками проверки состояния устройств *GATE*;
- блоками логических ключей *LOGIK* в совокупности с блоками проверки состояния логических ключей *GATE LR* и *GATE LS*.

### 8.1. Блок прямого перемещения

*DISPLACE*        A,B,[C],[D]

A – номер перемещаемого транзакта;

B – метка блока, к которому перемещается транзакт;

C – имя параметра перемещаемого транзакта, в который записывается время до конца обслуживания, если транзакт находился в списке будущих событий (на обслуживании);

D – метка выхода активного (перемещающего) транзакта; при ее отсутствии транзакт перемещается к следующему блоку.

При перемещении прерванные обслуживания не сбрасываются и из очереди транзакт не удаляется. Поэтому, если по логике модели необходимо освободить устройство или вывести из очереди, необходимо пропустить перемещаемый транзакт через блоки *RELEASE* или *DEPART*.

**Пример:** Удалить 3-й транзакт из модели

```
GENERATE „,10            ;Одновременное поступление 10 транзактов
DISPLACE 3,Uhod         ;Удаление 3-го
...
```

В данном примере третий транзакт будет удален из модели, не начав обслуживаться, первым же транзактом, поступившим в блок DISPLACE. Необходимо учесть, что, если бы в момент удаления 3-й транзакт находился на обслуживании, то устройство осталось бы занятым даже после его удаления и дальнейшая работа модели нарушилась бы. Если бы он находился в очереди, нарушилась бы статистика очереди. Поэтому необходимо вводить блоки, проверяющие состояния устройства и очереди, и выводить из этих устройств и очередей удаляемый транзакт.

## 8.2. Блок проверки числовых значений

Самый универсальный блок, позволяющий проверять практически любые условия: сравнивать числовые величины, проверять значение булевой переменной, проверять состояние устройств.

**TEST X A,B,[C]**

X – условный оператор (L, LE, E, NE, G, или GE);

A,B – сравниваемые величины; в случае, если A – булева переменная, B может принимать значения только 0 или 1;

C – метка блока альтернативного выхода, в случае, когда не выполняется условие. Если условие выполняется, транзакт переходит к следующему блоку.

В частности, с помощью блока TEST удобно организовать дисциплину обслуживания с ограниченной длиной очереди

### Пример 8.1

*Очередная заявка выводится из модели без обслуживания, если в очереди больше 5 заявок. Определить долю необслуженных заявок.*

*;Сегмент описаний*

**Dlina VARIABLE 6 ;Задание контрольной величины**

*;Сегмент имитации*

**GENERATE 10,5 ;Поток транзактов**

**TEST L Q\$Line,V\$Dlina,NotServ ;Если очередь Line длиннее 6,  
;транзакт переходит в строку с меткой NotServ**

```

SAVEVALUE      ServTr+,1      ;Подсчет числа обслуженных заявок
QUEUE          Line
SEIZE          OKU_1
DEPART        Line
ADVANCE       12,6
RELEASE       OKU_1

NotServ SAVEVALUE      AllTr+,1      ;Подсчет числа необслуженных
;заявок
TERMINATE

;Сегмент завершения
GENERATE 60000
SAVEVALUE      Dola,((X$AllTr-X$ServTr)/X$AllTr)      ;Доля
необслуженных
TERMINATE      1
START          1

```

Все транзакты попадают в блок TEST, производится проверка длины очереди с именем Line. Если в ней меньше шести транзактов, активный транзакт переходит к следующему блоку и становится в очередь на обслуживание. В противном случае – переходит в строку, помеченную меткой NotServ , добавляет единицу к СЧА сохраняемой ячейки для подсчета всех необслуженных транзактов и удаляется из модели без обслуживания. Доля необслуженных транзактов рассчитывается в сегменте завершения.

**Примечание 1.** Всегда, когда в блоке производится проверка каких-либо условий, при выполнении этих условий транзакт продвигается к следующему блоку, в случае невыполнения – продвигается по альтернативному пути, означенному меткой (режим альтернативного выхода), или задерживается на входе в проверяющий блок до выполнения условия (режим отказа во входе).

**Примечание 2.** Относится к блокам TEST, GATE, UNLINK. Блоки работают в двух режимах:

*Режим отказа во входе.* Метка альтернативного блока отсутствует. Транзакт, при невыполнении условий (к примеру, в блоке

**TEST E F\$Cum,0** СЧА ОКУ F\$Cum равно 1, т.е. устройство занято), не может войти в блок TEST и помещается в список повторных попыток, который просматривается при каждом последующем изменении состояния любого устройства в модели. В случае изменения состояния ОКУ Cum на «свободно», т.е. F\$Cum=0, транзакт входит в блок TEST и проходит на следующий блок (это может. В блоке TEST, работающем в режиме отказа, нельзя проверять состояние параметров транзакта (формат **TEST G P\$Param,4** применять нельзя) т.к. транзакт в этом случае может навсегда «застрять» в списке повторных попыток.

*Режим альтернативного выхода.* Если операнд C присутствует, транзакт всегда может войти в блок **TEST** и продвигаться дальше, в зависимости от выполнения проверяемого условия, либо к следующему блоку (условие выполнено) либо к блоку с меткой (номером), означенном операндом C. В этом случае можно проверять и параметр транзакта, например блок **TEST G P\$Param,4,Go** всегда открыт для входа, при входе транзакта проверяется значение его параметра с именем Param. Если выполняется  $P\$Param > 4$ , транзакт переходит к следующему блоку, если нет – переходит к блоку, помеченному меткой Go.

### **Задание 8.1**

*Дополнить нижеприведенную модель условием: распределить транзакты между двумя устройствами, в зависимости от времени существования транзакта в модели. Все транзакты со временем жизни меньше 10, отправить на ОКУ Short (время обслуживания 3-5 ЕМВ), остальные – на ОКУ Long (время обслуживания 1-3 ЕМВ). Определить долю транзактов, обслуженных в устройстве Short.*

**GENERATE (Exponential(3,0,3)),,100000**  
**ADVANCE (Uniform(6,9,12))**

**SEIZE Short**  
**ADVANCE 4,1**

**RELEASE Short**

**SEIZE Long**

**ADVANCE 2,1**

**RELEASE Long**

**TERMINATE 1**

**START 100000**

### **8.3. Блок пересылки транзактов**

**TRANSFER[A],[B],[C],[D]**

A – операнд задания режима работы

B – первая метка;

C – вторая метка;

D – шаг нумерации блоков для режима ALL.

Режимы работы блока TRANSFER:

1) , (запятая, операнд A отсутствует) – режим безусловной пересылки в строку, означенную первой меткой, все остальные операнды отсутствуют;

**TRANSFER ,Met1**

Транзакт, вошедший в блок TRANSFER безусловно переходит на блок, помеченный меткой Met1, если же этот блок занят, транзакт задерживается в блоке TRANSFER;

2) **.n** – статистический, выбор случайным образом одного из двух блоков, помеченных метками;

**TRANSFER .125,Met1,Met2**

С вероятностью 0,125 транзакты отправляются на метку **Met2**, с вероятностью 0,875 – на метку **Met1**. или на следующий блок, Если же второй операнд отсутствует:

**TRANSFER .125,,Met2**

с вероятностью 0,875 транзакт переходит на следующий блок.

3) **BOTH** – последовательный выбор из двух блоков;

**TRANSFER BOTH,Met1,Met2**

Последовательно проверяются два возможных пути, если невозможно двинуться ни по одному, транзакт задерживается в блоке **TRANSFER** и проверка повторяется в следующий момент модельного времени. Т.е. сначала проверяется возможность входа в блок, помеченный меткой Met1, если такой возможности нет, делается попытка направить транзакт в блок, помеченный меткой Met2. Опрос этих блоков производится при каждом изменении модельного времени до тех пор пока не появится возможность продвинуть транзакт к одному из блоков. Если первая метка отсутствует, проверяется возможность войти в следующий за блоком TRANSFER блок (**TRANSFER BOTH, Met2** попытается направить транзакт в следующий блок, затем в блок с меткой Met2).

4) **ALL** - последовательный выбор из нескольких блоков;

**TRANSFER ALL, Met1, Met2**

Аналогично, но последовательно проверяются несколько блоков между метками;

5) **PICK** – выбор случайным образом одного из нескольких блоков;

**TRANSFER PICK, Met1, Met2**

Аналогично, но несколько блоков между метками Met1 и Met2 проверяются случайным образом. При этом, если в режимах BOTH и ALL заявка направляется в первый освободившийся блок, в режиме PICK заявка ожидает освобождения блока, к которому она была направлена блоком TRANSFER.

6) **FN** - функциональный;

**TRANSFER FN, Func1, N**

Транзакт пытается перейти к блоку с номером, вычисляемым как:  
FN\$Func1+N

7) **P** – параметрический

**TRANSFER P, Param1, N**

Транзакт пытается перейти к блоку с номером P\$Param1+N

## Пример 8.2

Допустим, на мойке (см. Пример 7.2) осуществляется еще и полировка кузова на двух рабочих местах. Процесс полировки занимает  $80 \pm 20$  минут. 20% машин подвергаются после мойки обезжириванию за 100 руб., это занимает от 10 до 16 минут. Только 9% водителей, прошедших мойку решают отполировать машину. Мойка стоит от 400 до 800 руб, полировка от 1500 до 3000 руб. Подсчитать доход мойки за месяц.

В этом случае модель преобразуется к следующему виду:

```
;EMB = 1 минута
;Сегмент описания мойки.
Wash      STORAGE 3      ;Емкость мойки = 3 места
A_Time    VARIABLE 10    ;Среднее время подхода а/м = 10 минут
W_Time    VARIABLE 25    ;Среднее время мойки = 25 минут
W_HTime   VARIABLE 5     ;Полуинтервал времени мойки = 5 минут
M_Time    VARIABLE 30#24#60 ;Время моделирования

;Сегмент описания участка полировки
Polish    STORAGE 2      ;Участок полировки 2 места
P_Time    VARIABLE 80    ;Среднее время полировки
P_HTime   VARIABLE 20    ;Полуинтервал времени полировки

;Сегмент имитации мойки
GENERATE (Exponential(1,0,V$A_Time)) ;Поток а/м на мойку
ASSIGN Price,(DUniform(3,400,800)) ;Цена мойки
QUEUE    QWash          ;Очередь на мойку
ENTER    Wash           ;Занятие места на мойке
DEPART   QWash          ;Выход из очереди на мойку
ADVANCE  V$W_Time,V$W_HTime ;Процесс мойки
TRANSFER .8,,OutW      ;80% не требуют обезжиривания
ASSIGN   Price+,100    ;20% доплачивают за обезжиривание
ADVANCE  13,3          ;Процесс обезжиривания
OutW LEAVE Wash        ;Завершение мойки
TRANSFER .91,,Out      ;91% уходят без полировки

;Сегмент имитации полировки
ASSIGN Price+,(DUniform(4,1500,3000)) ;Добавляется цена полировки
QUEUE    QPolish       ;Очередь на полировку
ENTER    Polish         ;Занятие места на участке
полировки
DEPART   QPolish       ;Выход из очереди на полировку
ADVANCE  V$P_Time,V$P_HTime ;Процесс полировки
```

Out	LEAVE	Polish		;Завершение полировки
	SAVEVALUE	Profit+,P\$Price		;Запись уплаченной цены
	TERMINATE			;Отъезд с мойки

;Сегмент завершения  
**GENERATE** V\$M\_Time ;Завершающий транзакт  
**TERMINATE** 1 ;Удаление и списание 1 со счетчика завершения  
**START** 1 ;Запуск модели и завершение по времени

### **Задание 8.2**

*Звонки в колл-центр туристической фирмы поступают по экспоненциальному закону со средним 10 минут. Звонки принимают два менеджера по продажам. Звонки переводятся на первого менеджера, который беседует с клиентом 10-15 минут, если он занят – на второго, который ведет беседу 15-20 минут (использовать BOTH). Из-за разного опыта работы первый менеджер перенаправляет клиентов к начальнику отдела в 10% случаев, второй – в 30%. Начальник отдела решает проблему за 4-10 минут.*

*Определить количество клиентов, обслуженных в течение месяца (рабочий день 8 часов) первым, вторым менеджерами и начальником отдела и коэффициенты их загрузки. Стоит ли держать второго менеджера или его нужно уволить? А может необходимо нанять еще одного?*

**GENERATE (Exponential(2,0,10))**

**SEIZE** Man1  
**ADVANCE** (Uniform(1,10,15))  
**RELEASE** Man1

**SEIZE** Man2  
**ADVANCE** (Uniform(1,15,20))  
**RELEASE** Man2

**SEIZE** Boss  
**ADVANCE** (Uniform(1,4,10))  
**RELEASE** Boss

**GENERATE (60#8#24)**

**TERMINATE**      1  
**START**            1

### **Задание 8.3**

*Фирма «Декларант», имеющая четырех сотрудников-декларантов оформляет документацию на три вида товаров (заявки на оформление стоят в общей очереди):*

*А) продовольственные, заявки на оформление поступают по экспоненциальному закону со средним 80, время оформления от 40 до 90 (равномерный закон);*

*Б) бытовая техника, заявки на оформление поступают по экспоненциальному закону со средним 100, время оформления от 80 до 130;*

*В) автомобили, заявки на оформление поступают по экспоненциальному закону со средним 200, время оформления от 180 до 320;*

*Определить среднюю длину очереди и среднее время ожидания за месяц при рабочем дне 12 часов и 25 рабочих днях в месяце.*

### **Задание 8.4**

*Заявки распределяются по четырем различным ОКУ равновероятно. Так как блоки устройств расположены не подряд, следует направить заявки сначала на группу блоков TRANSFER, работающих в режиме безусловной передачи, а каждое устройство пометить своей меткой.*

## **8.4. Организация циклов**

**LOOP      А,В**

А – параметр транзакта, в котором содержится число повторений какого-либо участка модели. Поскольку в качестве этого операнда может использоваться только параметр транзакта, префикс P\$ не нужен.

**LOOP      CycleParam,Start.**

При этом цикл будет повторяться количество раз, записанное в параметр транзакта с именем «CycleParam».

В – метка начала цикла (в примере – Start) . Транзакт будет возвращаться блоком LOOP на метку, означенную в операнде В (StartOfCycle). Метка должна стоять выше блока LOOP.

### **Пример 8.3**

Канцелярия готовит документ на подпись начальнику управления в течение 3-7 часов. При этом 20% документов отправляется на доработку, требующую 2-4 часа. Определить среднее время на подготовку документа (всего 1000 документов).

```
Povtor FUNCTION RN2,D2 ;Функция задания случайного
.8,1/1,2 ;количества проходов

GENERATE 1,,1000 ;Генрация 1000 док-в с интервалом 1 час
ASSIGN TimeD,(Uniform(1,3,7)) ;Время разработки записываем в параметр
ASSIGN NumD,(FN$Povtor) ;Количество проходов записываем в
;параметр
Dor ADVANCE P$TimeD ;Процесс разработки документа
ASSIGN TimeD,(Uniform(3,2,4)) ;Назначение времени доработки
LOOP NumD,Dor ;Отправка на доработку 20% документов

TERMINATE 1
START 1000
```

### **Задание 8.5**

*Поток пациентов коммерческого медицинского центра – пуассоновский с интенсивностью 4 человека в час. Половине пациентов требуется первая процедура, которая нужно пройти (по 1 в день) 3 раза и длится 8+-2 минуты. Еще 30% - вторая процедура 5 раз с длительностью 9+-3 минуты, остальным - третья процедура 6 раз с длительностью 11+-3 минуты.*

*Медцентр работает 12 часов в день без выходных. Цена 1 процедуры 800 рублей за сеанс, 2-й - 600 руб., 3-й - 1200 руб.*

*Подсчитать доход медцентра за месяц.*

### **Задание 8.6**

*На прием экзамена у одного студента преподавателю отводится по нормативу 25 минут. На пересдачу экзамена – 35 минут. По статистике 20% студентов не сдают экзамен с первой попытки, и половина из них вынуждена пересдавать второй раз. Смоделировать процесс и рассчитать общее время на прием экзамена от 100 студентов.*

*Задание выполнить с помощью блока цикла, при этом уровень подготовки студентов (количество сдач 1, 2 или 3) задать дискретной случайной функцией и записать количество попыток в параметр транзакта. Время сдачи или пересдачи также занести в параметр транзакта, используя блоки ASSIGN и TEST.*

### **Задание 8.7**

*Заявки обрабатываются на двух параллельных участках, куда направляются равновероятно. Время обработки на первом составляет 3-11, на втором – 2-10 ЕМВ. При этом на первом участке 5%, а на втором 7% заявок требуют повторной обработки. Повторная обработка производится в приоритетном режиме. По окончании обработки заявки направляются в двухканальное устройство завершения обработки, где среднее время обслуживания 20% заявок – 3 ЕМВ, остальных – 5 ЕМВ.*

*Рассмотреть обработку 1000 заявок.*

- 1. Разработать маршруты движения.*
- 2. Определить количество повторно обработанных заявок на первом участке.*
- 3. Определить среднее время пребывания заявки в модели.*

## Глава 9. Списки пользователя

Пользователь имеет возможность на время выводить транзакты из списка текущих событий (т.е. приостанавливать движение транзактов по модели), внося их в т.н. «список пользователя». Список пользователя (СП) представляет собой набор транзактов, аналогичный очереди. Транзакты, помещенные в список пользователя не просматриваются и не перемещаются по модели до тех пор, пока пользователь не возвратит их в модель. Основное отличие списка пользователя от очереди состоит в том, что транзакты могут входить в список пользователя и выходить из нее не только в порядке поступления, но и в любом другом порядке, заданном в GPSS-модели.

С каждым списком пользователя связан индикатор, который может находиться в одном из состояний: “установлен” (равен 1, при этом список открыт для входа транзактов) или “сброшен” (равен 0, список закрыт). В начале имитации все индикаторы списков пользователя сброшены.

Важное (но не единственное) применение списков пользователя – имитация сложных дисциплин обслуживания в СМО: “последним пришел – первым обслужен” (LIFO), ограничение на время пребывания в очереди и т.д.

### 9.1. Внесение транзакта в список пользователя

Для имитации входа транзакта в список пользователя применяется следующий блок:

**LINK      A,B,[C]**

где **A** – имя списка пользователя;

**B** – порядок включения транзактов в список пользователя. Операнд **B** может принимать следующие значения:

*FIFO*: транзакт помещается в конец списка пользователя;

*LIFO*: транзакт помещается в начало списка пользователя;

*PR*: транзакты размещаются в списке пользователя в порядке приоритетов;

*P\$имя\_параметра*: транзакты размещаются в списке пользователя в порядке возрастания указанного параметра (чем меньше значение параметра, тем ближе располагается транзакт к началу списка пользователя, т. е. в обратном параметре приоритета, если мы записали приоритет в параметр транзакта);

*MI*: транзакты размещаются в списке в порядке времени пребывания в модели;

**C** – метка блока, на который передается транзакт, если в момент его поступления в блок LINK индикатор списка пользователя оказывается сброшенным, т.е. список оказывается закрытым для входа (если список, имитирующий очередь, пуст, транзакт проходит ее без задержки на метку, указанную в операнде **C**). В этом случае транзакт не включается в список пользователя. После перенаправления транзакта на метку **C** индикатор списка пользователя переводится в состояние “установлен”, т.е. список открывается и следующие транзакты уже заносятся в него и ожидают вывода пользователем. Если операнд **C** не задан, то индикатор списка установлен (=1), все транзакты входящие в блок LINK безусловно включаются в список, заданный операндом **A** в порядке, определяемом операндом **B**.

### **Пример 9.1**

**LINK        Otstoy,FIFO**

Все транзакты, входящие в блок, заносятся в СП «Otstoy» в порядке поступления и дальше по модели не продвигаются.

**LINK        Otstoy,Stroy,Vyhod**

Первый подошедший транзакт находит список закрытым и переходит на блок с меткой «Vyhod», но при этом открывает список для следующих

транзактов. Эти транзакты размещаются в СП в порядке возрастания параметра транзакта «Stroy» (чем больше параметр, тем дальше от начала списка). Транзакты, выводимые в дальнейшем из СП пользователем, также переходят на метку» Vyhod».

#### **Стандартные числовые атрибуты списков пользователя**

**CA\$имя\_списка** – среднее число транзактов в СП;

**CC\$имя\_списка** – общее число транзактов, побывавших в СП;

**CH\$имя\_списка** – текущее число транзактов в СП;

**CM\$имя\_списка** – максимальное число транзактов в СП;

**CT\$имя\_списка** – среднее время пребывания транзакта в СП.

#### **9.2. Вывод транзактов из списков пользователя**

Транзакт, вошедший в список пользователя, не может продвигаться дальше (например, в следующий блок), пока он не будет выведен из списка пользователя другим транзактом, вошедшим в блок:

**UNLINK [X] A,B,C,D,E,F**

где X – обозначение операции сравнения: L – меньше, LE – меньше или равно, E – равно, NE – не равно, G – больше, GE – больше или равно;

A – имя списка пользователя, из которого выводятся транзакты;

B – метка оператора, на который передаются транзакты, выводимые из списка пользователя;

C – количество транзактов, выводимых из списка пользователя. Если этот операнд не указан или указано слово ALL, то из списка пользователя выводятся все транзакты;

D – может принимать одно из следующих значений:

*BACK*: транзакты выводятся из списка пользователя, начиная с ее конца (количество выводимых транзактов определяется операндом C);

*Имя (номер) параметра*: для всех транзактов, находящихся в списке пользователя, выполняется операция сравнения указанного параметра с

величиной, указанной в операнде E. Из списка пользователя выводятся те транзакты, для которых выполняется условие, заданное операцией сравнения [X]. Если операнд D задает параметр, а E не используется, параметр транзакта из списка пользователя сравнивается с таким же параметром активного транзакта. Если они равны, транзакт, находящийся в списке, выводится из списка на метку B;

*Имя логической переменной:* из списка пользователя выводятся те транзакты, для которых выполняется условие, заданное логической переменной;

E – величина, с которой сравнивается указанный в операнде D параметр транзактов, находящихся в списке пользователя. Операнд E используется только в том случае, если используется оператор отношения. В этом случае обязательно требуется операнд D. Список пользователя проверяется, начиная с начала. Если условие, заданное оператором отношения, выполняется для операндов D и E, исключается каждый транзакт, (вплоть до предела (операнд C)). Если в операнде E используется параметр транзакта, он вычисляется относительно активного транзакта.;

F – метка блока, на который передается транзакт, вошедший в блок UNLINK, если не удастся вывести из списка пользователя столько транзактов, сколько указано в операнде C (например, из-за того, что количество транзактов в списке пользователя меньше указанного в операнде C), или не удастся вывести ни одного транзакта (например, из-за того, что условие, заданное операндами D и E, не выполняется ни для одного транзакта).

Если после прохождения транзакта через блок UNLINK оказывается, что в списке пользователя нет транзактов, то индикатор списка пользователя сбрасывается (список закрывается, чтобы следующий транзакт мог его миновать).

Для определенности в дальнейшем будем называть транзакт, проходящий через блок UNLINK «выбивающим» (он освобождает из СП транзакты для их дальнейшего продвижения по модели), а транзакты, выходящие при этом из СП – «выбываемыми».

### **Пример 9.2**

#### **UNLINK Spisok,GoGo,2**

При входе выбивающего транзакта в этот блок первые два по списку транзакта из СП Spisok выходят из него и направляются в блок с меткой GoGo. Выбывающий транзакт направляется к следующему за UNLINK блоку.

### **Пример 9.3**

#### **UNLINK Spisok,GoGo,2,Param**

Из СП выводятся на метку GoGo только те два транзакта, параметр Param которых равен такому же параметру выбивающего транзакта. Выбывающий транзакт направляется к следующему за UNLINK блоку.

### **Пример 9.4**

#### **UNLINK E Spisok,GoGo,3,ParamNew,X\$ContrNum**

Из СП выводятся на метку GoGo только те транзакты, параметр ParamNew которых равен СЧА сохраняемой ячейки ContrNum. Выбывающий транзакт направляется к следующему за UNLINK блоку.

### **Пример 9.5**

#### **UNLINK Spisok,GoGo,1,BV\$NumIf**

Из СП выводятся на метку GoGo один транзакт, если булева переменная NumIf = 1, т.е. «истина». Выбывающий транзакт направляется к следующему за UNLINK блоку.

### **Пример 9.6**

Проверку условия можно выполнить и непосредственно в блоке UNLINK, если использовать операцию сравнения:

### **UNLINK G Spisok,GoGo,5,Group1,10,Out**

Из СП на метку GoGo выводятся пять транзактов, у которых значение параметра Group1 больше десяти. Если таких транзактов в СП меньше пяти, выбивающий транзакт направляется в блок с меткой Out, а индикатор СП устанавливается в открытое состояние.

#### **Пример 9.7**

### **UNLINK LE Nakop,Obr,3,Par2,10,Kon**

При поступлении транзакта в этот оператор выполняются следующие действия. Из списка пользователя Nakop, с его начала, выводятся транзакты, у которых значение параметра Par2 не превышает 10. Количество транзактов, выводимых из списка пользователя, не превышает трех. Если заданное условие ( $Par2 \leq 10$ ) выполняется более чем для трех транзактов, находящихся в СП, то выводятся только три из них – те, что находятся ближе к началу списка. Транзакты, выведенные из СП, направляются на оператор с меткой Obr. Если из списка пользователя выведено три транзакта, то транзакт, прошедший через оператор UNLINK, направляется на оператор, следующий за UNLINK, а если выведено менее трех транзактов – то на оператор с меткой Kon.

#### **Задание 9.1**

*В редакции студенческого журнала имеется один копировальный аппарат. Заказы на выполнение копировальных работ поступают в среднем через каждые 30 минут; поток заказов можно считать пуассоновским. Копирование одного листа занимает примерно 1,5 минуты (эту величину можно считать постоянной). Размер заказа обычно составляет от одного до 25 листов.*

*Требуется выбрать дисциплину обслуживания, обеспечивающую минимальное среднее время выполнения заказов. Предлагаются три варианта дисциплины обслуживания сигналов: а) дисциплина FIFO*

(обслуживание заказов в порядке поступления); дисциплина LIFO; дисциплина, при которой в первую очередь обслуживаются заявки малого объема (чем меньше объем, тем выше приоритет). Для выбора лучшей дисциплины обслуживания необходимо провести имитационное моделирование выполнения 1000 заказов с использованием различных дисциплин обслуживания.

а) Дисциплина обслуживания FIFO

```
VREM TABLE      M1,20,20,12
GENERATE (EXPONENTIAL(1,0,30)),,1000
ASSIGN List,(1.5#DUNIFORM(2,1,25))
QUEUE OCH
SEIZE XEROX
DEPART OCH
ADVANCE P$List
RELEASE XEROX
TABULATE VREM
TERMINATE 1
START 1000
```

Здесь в блоке *ASSIGN List,(1.5#DUNIFORM(2,1,25))* вычисляется время, необходимое для выполнения заказа; эта величина присваивается параметру транзакта *List*. Для сбора данных о времени выполнения заказа используется таблица *VREM*. Так как требуется выполнить моделирование 1000 заказов, модель запускается на выполнение командой *START 1000*.

По результатам моделирования среднее время выполнения заказа составило 42,572 мин (эта величина указана в колонке *MEAN* таблицы *VREM* в отчете).

С помощью списка пользователя организовать следующие дисциплины очереди:

б) дисциплина LIFO (первым выполняется заказ, поступивший последним);

в) обслуживание по минимальному времени (в первую очередь выполняются заказы, для выполнения которых требуется меньшее время).

### **Задание 9.2**

*Готовые изделия, выпускаемые в цехе, поступают на площадку для временного хранения. В среднем цех выпускает одно изделие в пять минут; интервалы между моментами выпуска изделий можно считать случайными величинами, распределенными по экспоненциальному закону.*

*Перевозка готовых изделий с площадки на участок упаковки выполняется транспортной тележкой. Тележка вмещает 20 изделий; если в момент приезда тележки на площадке оказывается менее 20 изделий, то вывозятся все имеющиеся изделия. Перевозка готовых изделий на участок упаковки занимает от 10 до 15 минут. Так как тележка обслуживает несколько цехов, интервалы между приездами тележки в цех составляют  $1 \text{ час} \pm 10 \text{ минут}$ .*

*На участке упаковки имеются две упаковочные машины. Упаковка одного изделия занимает от 5 до 15 минут.*

*Требуется разработать GPSS-модель для анализа процесса перевозки и упаковки изделий в течение 24 часов.*

### **Задание 9.3**

*В буфер-накопитель сервера обработки данных со спутников погоды поступают файлы с фотографиями, показывающие состояние атмосферы. Обработка одного файла с фотоматериалом (расшифровка, конвертация, печать, корректировка прогноза) занимает от 2 до 5 мин. Поток файлов можно считать пуассоновским; в среднем интервал между моментами поступления файлов составляет 3 мин. Если информация ожидает обработки в буфере более тридцати минут, то она считается устаревшей и ее обработка становится нецелесообразной (файл отправляется в архив без обработки).*

*Требуется разработать модель для анализа работы сервера в течение двадцати четырех часов. Подсчитать количество обработанных*

файлов и количество отправленных в архив.

#### **Задание 9.4**

*На остановку маршрутного такси пассажиры подходят по экспоненциальному закону со средним 1 минута. В такси 10 мест. Терпения у пассажира на остановке хватает на 10 минут, после этого он уходит на другой транспорт. Такси подходят с интервалом от 3 до 15 минут. В такси при высадке пассажиров на остановке освобождается от 0 до 10 свободных мест, после этого пассажиры с остановки занимают свободные места. В начальный момент времени все места в маршрутках свободны.*

*Рассчитать процент обслуженных пассажиров за 18 часов работы транспорта.*

#### **Задание 9.5**

*Пограничный контроль на границе осуществляется в двух коридорах. Каждый из них закрывается на 15 минут раз в 2-3 часа. Одновременно может быть закрытым только один коридор. Очереди в оба коридора независимые, но, если закрывается один коридор, все туристы перемещаются в очередь к другому и обратно вернуться уже не могут.*

*Туристы подъезжают в автобусах по 40-50 человек в каждом, автобусы приезжают по экспоненциальному закону в среднем раз в два часа. Процедура контроля занимает 1-3 минуты.*

*Оценить среднюю и максимальную длины очередей, коэффициент загрузки коридоров.*

*Как изменится время ожидания контроля при открытии третьего аналогичного коридора, если из закрытого коридора туристы перераспределяется по двум оставшимся очередям равновероятно.*

*Смоделировать работу поста в течение суток.*

## ГЛАВА 10. Проверка состояния устройств и синхронизация транзактов

### 10.1. Проверка ОКУ

**GATE X A,[B]**

X – условный оператор, связанный с ОКУ (NU – свободно, U – занято);

A – имя (номер) ОКУ;

B – метка альтернативного выхода

Блок GATE может работать в режиме **отказа** (операнд B отсутствует) или **альтернативного выхода** (операнд B присутствует). В режиме отказа транзакт не может войти в блок GATE, он будет задержан в блоке, предшествующем блоку GATE, до тех пор, пока не будет выполнено условие, заданное условным оператором X.

#### Пример 10.1

```
...  
ADVANCE 1  
GATE NU Parent  
TRANSFER ,Course  
...
```

Очередной транзакт попытается перейти из блока **ADVANCE** в блок **GATE**, но удастся ему это только тогда, когда освободится одноканальное устройство **Parent**, после этого транзакт пройдет к следующему за GATE блоку и будет отправлен в строку, помеченную меткой **Course**.

В режиме альтернативного выхода транзакт всегда может войти в блок GATE, и, если условие выполнено, перейти к следующему блоку, если же условие не выполнено, транзакт переходит в строку, помеченную меткой альтернативного выхода.

#### Пример 10.2

```
...  
ADVANCE 1  
GATE NU Parent,Prov
```

...  
Если ОКУ **Parent** свободно, транзакт пройдет к следующему за GATE блоку. Если ОКУ **Parent** занято, транзакт перейдет в строку, помеченную меткой **Prov**.

### **Задание 10.1**

*Смоделировать работу двух ОКУ с временами обслуживания 10-20 ЕМВ и 20-30 ЕМВ, соответственно. Ко второму ОКУ транзакт направляется, если занято первое. Если же заняты оба, транзакт удаляется. Определить процент необслуженных транзактов, если время их поступления в модель составляет 5-15 ЕМВ а время моделирования ограничено 10000 ЕМВ.*

### **10.2. Проверка МКУ**

Состояние МКУ и работа блока GATE в этом случае аналогична п.7.1 за исключением того, что условный оператор X может принимать следующие значения:

SE – МКУ пусто;  
SF – МКУ заполнено;  
SNE – МКУ не пусто;  
SNF – МКУ не заполнено;  
SNV – МКУ недоступно;  
SV – доступно.

### **Пример 10.3**

...  
**GATE SF Dream**  
**ASSIGN ParamZ,"Занято"**  
**TRANSFER ,Contr**  
...

Транзакт ожидает полного заполнения МКУ **Dream**, после этого в его параметр **ParamZ** вписывается слово «Занято» и он переходит в строку с меткой **Contr**.

### Пример 10.4

```
Auto STORAGE      2
GENERATE 10
GATE SF   Auto,GoF
ASSIGN   ParamZ,"Занято"
TRANSFER ,Con
GoF ASSIGN   ParamZ,"Есть места"

Con TERMINATE     1
START 100
```

Если МКУ **Auto** заполнено, в параметр **ParamZ** вписывается слово «Занято» и транзакт переходит в строку с меткой **Con**. Если в МКУ **Auto** есть свободные каналы, транзакт переходит в строку с меткой **GoF**.

### Задание 10.2

*На первом складе два канала разгрузки автобусов, на втором – три. Водители автобусов при подъезде к границе города по рации запрашивают сведения о наличии свободных мест на складах и выбирают тот, на котором имеются свободные места. Если на обоих складах нет свободных мест под разгрузку, выбирается склад с меньшей очередью. Поток автобусов экспоненциальный со средним интервалом 10 ЕМВ, время досмотра 40-60 ЕМВ.*

*В приведенной модели расставить блоки проверки, блоки пересылки и метки. Прокомментировать вставленные строки. Определить коэффициенты загрузки постов и средние длины очередей в течение 100 000 ЕМВ.*

```
;Сегмент описаний
Post1 STORAGE      2
Post2 STORAGE      3

;Сегмент имитации
GENERATE (Exponential(1,0,5))
QUEUE   Q_Post1
ENTER   Post1
DEPART  Q_Post1
ADVANCE      50,10
```

```

LEAVE      Post1

QUEUE      Q_Post2
ENTER      Post2
DEPART     Q_Post2
ADVANCE    50,10
LEAVE      Post2

TERMINATE

```

### **Задание 10.3**

*С целью увеличения дальности беспосадочного полета производится дозаправка самолетов горючим в воздухе, если остаток горючего у них менее чем на час полета в режиме ожидания. В районе дозаправки постоянно дежурит 4 самолета-дозаправщика. Время дозаправки самолетов - случайная величина, равномерно распределенная на интервале [10 , 16] мин. Временные промежутки между прибытиями самолетов в район дозаправки также случайны, независимы и распределены равномерно на интервале [2,4] мин. Остаток топлива у самолетов – на время от 3 до 60 минут.*

*Если все дозаправщики заняты, но очередь не превышает три самолета, то очередной самолет присоединяется к очереди, в противном случае он запрашивает посадку на первом запасном аэродроме. Если этот аэродром не может принять самолет, он уходит на второй запасной аэродром. Первый аэродром закрывается каждые 15-30 минут на время 3 минуты.*

*Если у очередного самолета остаток горючего менее, чем на десять минут, он получает первоочередное обслуживание.*

*Подсчитать количество дозаправленных самолетов, самолетов, севших на первый и на второй запасные аэродромы в течение суток.*

### **10.3. Синхронизация транзактов**

Блок **SPLIT** предназначен для моделирования семейств транзактов,

порождаемых одним (порождающим) транзактом. В момент входа порождающего транзакта в блок **SPLIT A,B,C** создается несколько копий этого транзакта. Число копий задается в поле А. Исходный (порождающий) транзакт переходит в блок, определенный в поле В. Все копии переходят к блоку, следующему за **SPLIT**. Если поле С блока **SPLIT** пустое, то все копии идентичны породившему их транзакту. Например, при входе транзакта в блок

**SPLIT 4,NEXT**

порождаются четыре транзакта-копии, идентичные вошедшему, все они переходят в блок, помеченный меткой **NEXT**, а исходный транзакт переходит в следующий по тексту блок. Всего из этого блока **SPLIT** выходит пять транзактов.

Если поле С не пустое, то его значение интерпретируется как имя (номер) параметра транзакта, в который записывается порядковый номер транзакта в данном семействе. Пусть N - значение этого параметра в момент входа транзакта в блок **SPLIT**. Тогда в момент выхода из **SPLIT** данный параметр у исходного транзакта будет иметь значение N+1 , а у копий соответственно N+2, N+3, ... , N+K , где K - общее число вышедших из блока **SPLIT** транзактов. Например, если транзакт, имеющий ноль в параметре с именем Number, войдет в блок

**SPLIT 2,MetkaE,Number**

то параметр P\$Number у этого транзакта приобретет значение 1, а у копий соответственно 2 и 3.

Транзакты - копии могут двигаться в модели независимо друг от друга. Копии могут проходить блоки **SPLIT** и породить новые копии.

Множество, состоящее из исходного транзакта и всех его копий, называется семейством транзактов. Копия члена семейства является членом того же семейства. Любой транзакт - член одного и только одного

семейства.

Блок **ASSEMBLE** используется для моделирования одновременного завершения нескольких процессов. В нем происходит группировка транзактов одного семейства. Он собирает заданное в поле *A* число транзактов одного семейства и превращает их в один транзакт. Первый из транзактов какого-либо семейства, вошедший в блок, задерживается до тех пор, пока в этом блоке не накопится заданное число транзактов того же семейства. После этого первый транзакт выходит из блока **ASSEMBLE**, а остальные транзакты этого семейства уничтожаются.

В одном блоке **ASSEMBLE** могут одновременно проходить сборку транзакты, принадлежащие разным семействам. Например, если в блок

**ASSEMBLE**      4

поступают транзакты разных семейств, то транзакты каждого семейства собираются по четыре и каждая четверка превращается в один транзакт.

Аналогично применяется и блок **GATHER**. Отличие состоит лишь в том, что транзакты в этом блоке не уничтожаются. Когда в нем накапливается заданное в поле *A* число транзактов одного семейства, все они переходят к следующему блоку.

Пара блоков

**MetkaA**    **MATCH**    **MetkaB**

...

**MetkaB**    **MATCH**    **MetkaA**

синхронизирует продвижение двух транзактов, принадлежащих к одному семейству и двигающихся в модели по различным путям. В поле *MetkaB* блока **MATCH** задается имя другого блока, который называется сопряженным блоком **MATCH**. Два транзакта, принадлежащие к одному семейству и поступающие в пустые сопряженные блоки **MATCH** в моменты времени  $T_1$  и  $T_2 > T_1$ , выходят из этих блоков одновременно в момент  $T_2$ .

### Пример 10.5

Нужно промоделировать сборку изделий рабочими «А», «В» и «С». Изделия в разобранном виде поступают каждые  $300 \pm 100$  мин. Каждое из них разделяется между рабочими «А» и «В», которые параллельно готовят свою часть изделия к сборке. Подготовка изделия состоит из двух фаз, причем после окончания первой фазы производится сверка с одновременным участием обоих рабочих «А» и «В». После сверки рабочие независимо друг от друга выполняют вторую фазу. На выполнение первой фазы рабочий «А» тратит  $100 \pm 20$  мин, а рабочий «В»  $80 \pm 20$  мин.

Вторую фазу рабочий «А» выполняет  $50 \pm 5$  мин, а рабочий «В»  $70 \pm 10$  мин.

После окончания работы рабочими «А» и «В» рабочий «С» выполняет сборку изделия. Сборка занимает  $50 \pm 5$  мин. и не может быть начата до тех пор, пока рабочие «А» и «В» не закончили выполнение своих операций.

Модель описанного процесса может быть написана с применением блоков синхронизации транзактов:

#### **;МОДЕЛЬ СБОРКИ ИЗДЕЛИЙ**

	<b>GENERATE 300,100</b>		<b>;ПОСТУПЛЕНИЕ ИЗДЕЛИЙ</b>
	<b>SPLIT 1,ManB</b>		<b>;РАЗДЕЛЕНИЕ ИЗДЕЛИЙ</b>
	<b>SEIZE Master_A</b>		<b>;ЗАНЯТЬ РАБОЧЕГО «А»</b>
	<b>ADVANCE 100,20</b>		<b>;ПЕРВАЯ ФАЗА</b>
<b>Sverka_A</b>	<b>MATCH Sverka_B</b>		<b>;ЖДАТЬ РАБОЧЕГО «В»</b>
	<b>ADVANCE 50,5</b>		<b>;ВТОРАЯ ФАЗА</b>
	<b>RELEASE Master_A</b>		<b>;ОСВОБОДИТЬ РАБОЧЕГО «А»</b>
	<b>TRANSFER ,Sborka</b>		<b>;ПЕРЕДАЧА ИЗДЕЛИЯ В СБОРКУ</b>
<b>ManB</b>	<b>SEIZE Master_B</b>		<b>;ЗАНЯТЬ РАБОЧЕГО «В»</b>
	<b>ADVANCE 90,20</b>		<b>;ВРЕМЯ ПЕРВОЙ ФАЗЫ</b>
<b>Sverka_B</b>	<b>MATCH Sverka_A</b>		<b>;ЖДАТЬ РАБОЧЕГО «А»</b>
	<b>ADVANCE 70,10</b>		<b>;ВТОРАЯ ФАЗА</b>
	<b>RELEASE Master_B</b>		<b>;ОСВОБОДИТЬ РАБОЧЕГО «В»</b>
<b>Sborka</b>	<b>ASSEMBLE 2</b>		<b>;ЖДАТЬ ОБЕ ЧАСТИ ИЗДЕЛИЯ</b>
	<b>SEIZE Master_C</b>		<b>;ЗАНЯТЬ РАБОЧЕГО «С»</b>
	<b>ADVANCE 50,5</b>		<b>;СБОРКА ИЗДЕЛИЯ</b>

**RELEASE Master\_C ;ОСВОБОДИТЬ РАБОЧЕГО «С»**

**TERMINATE 1 ;СБОРКА ЗАВЕРШЕНА**  
**START 1000 ;МОДЕЛИРОВАТЬ 1000 СБОРОК**

#### ***Задание 10.4***

*На склад прибывают машины с 20-30 палетами с товаром в среднем каждые 30 минут, причем с 10 утра интенсивность потока возрастает до пиковой в 18.00, когда становится внятеро больше, затем к 24.00 спадает до исходной. Разгрузка осуществляется на трех терминалах авопогрузчиками, снятие и перевозка одной палеты занимает от 3 до 15 минут. Оформление документов на каждую машину производится после разгрузки и выполняется двумя специалистами, оформление занимает 10-16 минут.*

*Определить необходимое количество автопогрузчиков, чтобы среднее время ожидания на разгрузку не превышало одного часа. Определить среднее время нахождения машин на складе от прибытия до завершения оформления товара.*

#### ***Задание 10.5***

*На российско-финскую границу идет поток легковых автомобилей с интервалом 5+-3 минуты.*

*Количество пассажиров в легковом автомобиле составляет:*

*20% - нет пассажиров, один водитель;*

*40% - один пассажир;*

*25% - два пассажира;*

*10% - три пассажира;*

*5% - четыре пассажира.*

*Кроме этого с интервалом 45+-10 минут подходят туристские автобусы со случайным, равномерно распределенным количеством туристов.*

*В каждом автобусе от 35 до 55 туристов.*

*Таможенную декларацию заполняет только водитель, досмотр проходят только автомобили (водители).*

*Паспортный контроль проходят все. Автомобили и автобусы пересекают границу только после сбора всех своих пассажиров.*

*На российской границе паспортный контроль и проверка декларации производятся одним человеком (ОКУ), таможенный досмотр – двумя таможенниками. На финской границе паспортный контроль – 4 пограничника, досмотр – по два таможенника.*

*Время паспортного контроля 30+-6 секунд*

*Время заполнения и проверки таможенной декларации 4+-1 минута*

*Время таможенного досмотра 3+-1 минута*

*Проанализировать длины очередей на паспортный контроль и таможенный досмотр на российской границе.*

## Заключение

Компьютерное моделирование является мощным инструментом исследования сложных систем. Метод имитационного моделирования позволяет создавать адекватные модели организационных, производственных и экономических структур и «проигрывать» на них реальные ситуации, в том числе и те, которые могут привести к выходу системы из строя или ее полному разрушению, чего нельзя допускать в реальности.

Учебное пособие направлено на изучение теоретических основ и формирование практических навыков анализа сложных структур с использованием современных методов имитационного моделирования систем для решения задач научно-исследовательского, управленческого и экономического характера.

Изучение материала, изложенного в учебном пособии, способствует развитию следующих навыков и компетенций:

- освоению методов анализа сложных систем, построения логических схем их функционирования, выявления проблемных элементов;
- овладению методами формирования управленческих решений в условиях определенности, неопределенности и риска;
- освоению принципов построения имитационных моделей и их реализации в среде моделирования GPSS;
- формированию устойчивых навыков в отладке компьютерных моделей для решения практических задач;
- освоению компьютерных технологий обработки данных и результатов моделирования;
- приобретению навыков постановки модельных экспериментов.

В конечном итоге это позволяет принимать обоснованные решения по организации работы в условиях воздействия множества трудно формализуемых факторов.

## СПИСОК ЛИТЕРАТУРЫ

1. Акопов А.С. Имитационное моделирование: : учебник и практикум для академического бакалавриата / М.: Юрайт, 2014. – 389 с.
2. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World. - СПб.: БХВ-Петербург, 2012. - 368 с.
3. Бражник А.Н. Имитационное моделирование: возможности GPSS World. - СПб.: Реноме. 2006. - 439 с.
4. Девятков В. В. Единое исследовательское пространство имитационного моделирования в среде GPSS Studio / В. В. Девятков, М. В. Федотов, Т. В. Девятков // Прикладная информатика: сб. статей. – 2019 - т.14 № 5(83). – с. 98-110.
5. Девятков В.В. Методология и технология имитационных исследований сложных систем: современное состояние и перспективы развития [Текст]: монография / В.В. Девятков — М.: Вузовский учебник: ИНФРА-М, 2013. - 448 с.
6. Емельянов А.А., Власова Е.А., Дума Р.В. Имитационное моделирование экономических процессов. - М.: Финансы и статистика. 2006. - 416 с.
7. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. - М.: ДМК Пресс. 2004. - 320 с.
8. Рыжиков Ю.И. Имитационное моделирование: Теория и технологии. СПб.: КОРОНА принт , 2004. - 384 с.
9. Томашевский В.Н., Жданова Е.Т. Имитационное моделирование в среде GPSS. - М.: Бестселлер, 2003.
10. Арсеньев Б. П., Яковлев С. А. Интеграция распределенных баз данных (монография). – СПб.: Лань, 2001. – 464 с.
11. Бусленко Н. П. Моделирование сложных систем. Москва, 1961 г.
12. Вавилов А. А. и др. Имитационное моделирование производственных систем. М.: Техника, 1983 г.

13. Советов Б. Я., Яковлев С. А. Моделирование систем (учебник). - М.: Высшая школа; (3-е изд.) 2001. - 343 с.
14. Советов Б. Я., Яковлев С. А. Моделирование систем. Учебник для ВУЗов. М.: Высшая школа, 1999 г. - 224 с.
15. Советов Б. Я., Яковлев С. А. Моделирование систем - Лабораторный практикум (учебное пособие с грифом Минвуза). - М.: Высшая школа, 1979. - 141 с.
16. Сосновиков Г.К. Компьютерное моделирование [Текст]: практикум по работе в среде GPSS World / Г.К. Сосновиков, В.А. Воробейников. — М.: Форум, 2015. — 108 с.
17. Шеннон Р. Дж. Имитационное моделирование систем - искусство и наука. М.: Мир, 1978 г. - 418 с.
18. Шрайбер Т. Дж. Моделирование на GPSS. М.: Машиностроение, 1980 г. - 592 с.

## Числовые атрибуты GPSS World

## 1. Системные числовые атрибуты

СЧА	Расшифровка
RNj	Случайное равномерно распределенное число в диапазоне от 0 до 0,999; генерируемое j-м генератором случайных чисел
C1	Текущее значение условного времени. Автоматически изменяется программой и устанавливается в 0 управляющими операторами CLEAR или RESET.
AC1	Текущее значение абсолютного времени. Автоматически изменяется программой. Эта величина не меняется под действием управляющего оператора RESET и устанавливается в 0 лишь под действием оператора CLEAR.
TG1	Число, равное текущему значению счетчика завершений. Транзакты, вошедшие в блоки TERMINATE с ненулевым операндом А, уменьшают значение этого счетчика на число, равное значению операнда А.
M1	Время пребывания в модели транзакта, обрабатываемого программой в данный момент. Эта величина может изменяться блоком MARK. Время пребывания вычисляется следующим образом: M1 равно разнице текущего значения абсолютного времени и отметки времени обрабатываемого транзакта.

## 2. Стандартные числовые атрибуты

## СЧА транзактов

СЧА	Расшифровка
XN1	Номер активного транзакта.
Pj P\$<имя>	значение параметра j текущего транзакта или значение параметра с именем <имя> текущего транзакта.
M1	Время пребывания в модели активного транзакта., либо время от прохождения последнего блока MARK без операнда А.
MPj MP\$<имя>	Значение времени, равное разности абсолютного модельного времени и содержимого j-го параметра текущего транзакта.
MBj MB\$<имя>	Флаг синхронизации: 1, если транзакт в блоке j принадлежит тому же семейству, что и текущий транзакт; 0 - в противном случае.
PR	Приоритет активного транзакта

## СЧА блоков

СЧА	Расшифровка
Nj N\$<имя>	Общее число транзактов, вошедших в j-й блок. Подсчет ведется программой автоматически. Например, N\$Radio - счетчик числа входов в блок Radio.
Wj W\$<имя>	Текущее число транзактов, которые находятся в блоке j. Значение этого счетчика подсчитывается автоматически. Например, W\$SNOW - счетчик текущего числа транзактов в блоке SNOW.

СЧА	Расшифровка
Fj F\$<имя>	Флаг занятости устройства. Эта величина равна 0, если устройство свободно, и 1 - во всех остальных случаях. Этот атрибут изменяется блоками SEIZE, RELEASE, PREEMPT (захват), PREEMPT (прерывание) и RETURN.
Fij Fi\$<имя>	Флаг прерывания устройства: $i=1$ , если устройство находится в состоянии захват в режиме прерывания (блок PREEMPT (захват)), $i=0$ - в противном случае.
FRj FR\$<имя>	Коэффициент использования устройства FRj считается в обычном виде (например 0.709), а не в тысячных долях (например 709).
FCj FC\$<имя>	Общее число входов в устройство j.
FTj FT\$<имя>	Среднее время использования устройства одним транзактом.
FVj FV\$<имя>	Флаг доступности устройства: 1 - если доступно, 0 - недоступно. После начала моделирования устройство доступно. Устройство переводится в доступное блоком FAVAIL.

#### СЧА многоканальных устройств (памятей, МКУ)

СЧА	Расшифровка
Sj S\$<имя>	Текущее содержимое многоканального устройства j. Содержимое многоканального устройства может изменяться блоками ENTER и LEAVE.
Rj R\$<имя>	Число свободных единиц многоканального устройства j. Эта величина может изменяться блоками ENTER и LEAVE.
SRj SR\$<имя>	Коэффициент использования многоканального устройства SRj в обычном виде (например 0.651), а не в тысячных долях (например 651).
SAj SA\$<имя>	Среднее содержимое многоканального устройства j (целая часть).
SMj SM\$<имя>	Максимальное содержимое многоканального устройства j.
SCj SC\$<имя>	Общее число входов в многоканальное устройство j.
STj ST\$<имя>	Среднее время пребывания транзактов в многоканальном устройстве j.
SEj SE\$<имя>	Флаг незанятости многоканального устройства j: 1 - свободно, 0 - занято.
SFj SF\$<имя>	Флаг заполненности многоканального устройства j: 1 - заполнено, 0 - не заполнено.
SVj SV\$<имя>	Флаг готовности многоканального устройства j: 1 - готово, 0 - не готово.

#### СЧА очередей

СЧА	Расшифровка
Qj	Длина соответствующей очереди j. Эта величина изменяется блоками

Q\$<имя>	QUEUE и DEPART.
QAj QA\$<имя>	Средняя длина очереди j.
QMj QM\$<имя>	Максимальная длина очереди j.
QCj QC\$<имя>	Общее число входов в очередь j.
QZj QZ\$<имя>	Число нулевых входов в очередь j.
QTj QT\$<имя>	Среднее время пребывания транзакта в очереди j (включая нулевые входы).
QXj QX\$<имя>	Среднее время пребывания транзакта в очереди j (без нулевых входов).

#### СЧА таблиц

СЧА	Расшифровка
TBj TB\$<имя>	Вычисленное среднее таблицы j. Для занесения в таблицу используется блок TABULATE.
TCj TC\$<имя>	Общее число включений в таблицу j.
TDj TD\$<имя>	Вычисленное среднеквадратичное отклонение для таблицы.

#### СЧА сохраняемых величин

СЧА	Расшифровка
Xj X\$<имя>	Содержимое хранимой величины j.
XHj XH\$<имя>	Содержимое хранимой величины j. СЧА XHj возвращает такое же значение, что и Xj.

#### СЧА матриц

СЧА	Расшифровка
MXj(a,b) MX\$<имя> (a,b)	Содержимое элемента матрицы ячеек j, расположенного в строке a, столбце b.

#### СЧА функций

СЧА	Расшифровка
FNj FN\$<имя>	Вычисленное значение функции j. От значения функции берется целая часть, за исключением тех случаев, когда это значение используется в качестве модификатора в блоках GENERATE, ADVANCE или ASSIGN или в качестве аргумента другой функции.

#### СЧА переменных

СЧА	Расшифровка
-----	-------------

Vj V\$<имя>	Вычисленное значение переменной j. При вычислении значения переменной с фиксированной точкой получается целое число. При вычислении значения переменной с плавающей точкой дробная часть конечного результата отбрасывается.
BVj BV\$<имя>	Вычисленное значение (1 или 0) булевой переменной.

#### СЧА числовых групп

СЧА	Расшифровка
GNj GN\$<имя>	Текущее число элементов в числовой группе с номером j.

#### СЧА групп транзактов

СЧА	Расшифровка
GTj GT\$<имя>	Текущее число элементов в группе транзактов с номером j.

#### СЧА списков пользователя

СЧА	Расшифровка
CHj CH\$<имя>	Текущее число транзактов в j-том списке пользователя. Если СЧА обращается к несуществующему списку пользователя, то возвращается 0 и создаётся данный список пользователя.
CAj CA\$<имя>	Среднее число транзактов в j-том списке пользователя. Если СЧА обращается к несуществующему списку пользователя, то возвращается 0 и создаётся данный список пользователя.
CMj CM\$<имя>	Максимальное число транзактов в j-м списке пользователя. Если СЧА обращается к несуществующему списку пользователя, то возвращается 0 и создаётся данный список пользователя.
CCj CC\$<имя>	Число входов в j-й список пользователя. Если СЧА обращается к несуществующему списку пользователя, то возвращается 0 и создаётся данный список пользователя.
CTj CT\$<имя>	Среднее время пребывания транзакта в j-том списке пользователя. Если СЧА обращается к несуществующему списку пользователя, то возвращается 0 и создаётся данный список пользователя.

#### СЧА логических переключателей

СЧА	Расшифровка
LSj LS\$<имя>	Возвращает состояние логического переключателя j: 1 - установлен, 0 - не установлен.
LRj LR\$<имя>	Возвращает инверсное состояние логического переключателя j: 0 - установлен, 1 - не установлен.

#### Стандартные логические атрибуты устройств

СЛА ОКУ	Расшифровка
FIj	устройство j захвачено,
FNIj	устройство j не захвачено,

FUj	устройство j занято,
FNUj	устройство j свободно,
Fj	устройство j не свободно,
FVj	устройство j доступно,
FNVj.	устройство j недоступно
<b>СЛА МКУ</b>	
SFj	МКУ j заполнено,
SNFj	МКУ j не заполнено,
SEj	МКУ j пусто,
SNEj	МКУ j не пусто,
SVj	МКУ j доступно,
SNVj	МКУ j недоступно.
<b>СЛА ключей</b>	
LSj	логический ключ j включен,
LRj	логический ключ j выключен.

*\* Здесь и далее под индексом "j" понимается или номер СЧА, СЛА или \$Имя.*

## Алфавитный указатель основных блоков GPSS World

1	<b>ADVANCE</b>	задержка транзакта, с включением его в список будущих событий
2	<b>ASSIGN</b>	запись данных в параметр транзакта
3	<b>DEPART</b>	вывод транзактов из очереди
4	<b>ENTER</b>	вход транзакта в многоканальное устройство
5	<b>FAVAIL</b>	смена состояния устройства на "доступно (разблокировано)"
6	<b>FUNAVAIL</b>	смена состояние устройства на "недоступно (заблокировано)"
7	<b>GATE</b>	проверка состояния объекта модели и изменение направления движения потока транзактов
8	<b>GATHER</b>	накопление нескольких связанных друг с другом транзактов (семейство транзактов)
9	<b>GENERATE</b>	создание транзакта и размещение его в списке будущих событий
10	<b>INDEX</b>	увеличение параметра транзакта
11	<b>LEAVE</b>	выход из многоканального устройства
12	<b>LINK</b>	внесение транзакта в список пользователя
13	<b>LOGIC</b>	модификация логического ключа
14	<b>LOOP</b>	организация цикла с уменьшением значения параметра-транзакта (счетчика)
15	<b>MARK</b>	сохранение значения системного времени в параметре транзакта
16	<b>MATCH</b>	ожидание транзактом связанных с ним транзактов в других блоках модели (синхронизация транзактов)
17	<b>MSAVEVALUE</b>	присваивание значений элементам матриц
18	<b>PREEMPT</b>	прерывание обработки транзакта и захват устройства активным транзактом в приоритетном режиме
19	<b>PRIORITY</b>	изменение приоритета транзакта
20	<b>QUEUE</b>	включение транзакта в очередь
21	<b>RELEASE</b>	освобождение одноканального устройства
22	<b>REMOVE</b>	удаление члена из числовой группы или группы транзактов
23	<b>RETURN</b>	освобождение устройства транзактом, захватившем его в блоке <b>PREEMPT</b>
24	<b>SAVAIL</b>	изменение состояния многоканального устройства на "доступно (разблокировано)"
25	<b>SAVEVALUE</b>	присваивание значений ячейки сохраняемых величин
26	<b>SEIZE</b>	занятие одноканального устройства или ожидание его освобождения для последующего занятия
27	<b>SPLIT</b>	создание копий транзакта с адресацией ее в некоторый блок
28	<b>SUNAVAIL</b>	изменение состояния многоканального устройства на

		"недоступно (заблокировано)"
29	<b>TABULATE</b>	обновление данных таблицы
30	<b>TERMINATE</b>	уничтожение транзакта, уменьшение счетчика завершенных транзактов
31	<b>TEST</b>	проверка арифметических условий и изменение направления движения потока транзактов
32	<b>TRANSFER</b>	пересылка транзакта на указанный блок

Предметный указатель основных блоков GPSS World

**1. Блоки, связанные с транзактами**

**GENERATE** A,B,C,D,E

Вводит транзакты в модель.

A – среднее значение интервала времени (необязательный операнд);

B – разброс или модификатор среднего значения (необязательный операнд);

C – время появления первого транзакта (необязательный операнд);

D – общее число генерируемых транзактов (необязательный операнд);

E – уровень приоритета каждого транзакта (необязательный операнд).

**TERMINATE** A

Удаляет активный транзакт из процесса моделирования.

A – величина уменьшения счетчика завершения (необязательный операнд).

**ADVANCE** A,B

Задерживает транзакт.

A – среднее время задержки;

B – разброс или модификатор среднего значения (необязательный операнд).

**2. Блоки, изменяющие значения параметров транзактов**

**ASSIGN** A,B,C

Изменяет значение параметра транзакта.

A – имя (номер) изменяемого параметра;

B – новое значение параметра;

C – номер функции (необязательный операнд).

## **INDEX**    A,B

Обновляет параметр активного транзакта.

A – имя (номер) параметра;

B – числовое значение, которое должно быть добавлено к содержимому параметра.

## **MARK** A

Записывает в параметр активного транзакта значение абсолютного модельного времени.

A – имя (номер) параметра (необязательный операнд).

## **PRIORITY**    A,B

Устанавливает приоритет активного транзакта.

A – новое значение приоритета;

B – BU (необязательный операнд).

### **3. Блоки, управляющие движением транзактов**

## **GATE** X    A,B

Изменяет маршрут движения транзактов в зависимости от состояния некоторого объекта.

X – условие в виде стандартного логического атрибута (FNV, FV, I, LS, LR, M, NI, NM, NU, SE, SF, SNE, SNF, SNV, SV, U);

A – имя (номер) проверяемого объекта;

B – метка строки перемещения транзакта в случае невыполнения условия (необязательный операнд).

## **LOOP**    A,B

Уменьшает параметр-счетчик транзакта на единицу и указывает метку строки – начала цикла.

A – параметр-счетчик активного транзакта, содержащий число

повторений цикла;

В – метка строки – начала цикла (необязательный операнд).

### **TEST X** A,B,C

Сравнивает числовые значения (обычно СЧА) и управляет местом назначения активного транзакта, основываясь на результате сравнения.

X – оператор отношения (E, G, GE, L, LE, NE);

A – проверяемое значение;

B – контрольное значение;

C – метка строки назначения (необязательный операнд).

### **TRANSFER** A,B,C,D

Обеспечивает переход активного транзакта к новому блоку.

A – режим (BOTH, ALL, PICK, FN, P, SBR,SIM) (необязательный операнд);

B – метка строки (необязательный операнд);

C – альтернативная метка строки (необязательный операнд);

D – приращение номера блока для режима ALL (необязательный операнд).

## **4. Блоки, работающие с транзактами в списке пользователя**

### **LINK**A,B,C

Управляет размещением активного транзакта в списке пользователя.

A – имя (номер) списка пользователя;

B – упорядочивание списка;

C – метка строки выхода транзактов (необязательный операнд).

### **UNLINK X** A,B,C,D,E,F

Управляет удалением активного транзакта из списка пользователя.

X – оператор отношения (E, G, GE, L, LE, NE);

А – имя (номер) списка пользователя;  
В – метка строки назначения для удаленных из списка транзактов;  
С – максимальное количество транзактов, которое можно удалить  
(необязательный операнд);  
D – проверяемое значение (необязательный операнд);  
Е – контрольное значение (необязательный операнд);  
F – метка (альтернативное место назначения для входящего транзакта)  
(необязательный операнд).

### **5. Блоки, работающие с копиями транзактов**

**ASSEMBLE**     A

Ожидает и уничтожает транзакты.

А – счетчик транзактов.

**GATHER**    A

Накапливает транзакты, являющиеся членами семейства.

А – счетчик транзактов, которые должны быть накоплены.

**MATCH**     A

Пара сопряженных блоков MATCH заставляет транзакты ожидать друг друга.

А – метка строки, в которой проверяется наличие транзакта.

**SPLIT**     A,B,C

Создает транзакты того же семейства, что и активный транзакт.

А – количество создаваемых транзактов;

В – метка строки, на которую отправляются транзакты-копии;

С – номер параметра (необязательный операнд).

### **6. Блоки, связанные с ОКУ**

**SEIZE A**

Занимает устройство.

A - имя (номер) устройства.

**RELEASE A**

Освобождает устройство.

A – имя устройства (числовое или символьное имя освобождаемого устройства).

**PREEMPT A**

Переводит устройство в прерванное состояние.

A – имя прерываемого устройства.

**RETURN A**

Удаляет транзакт из прерванного устройства.

A - имя (номер) устройства (числовое или символьное имя освобождаемого устройства).

**FAVAIL A**

Обеспечивает перевод устройства с именем A в доступное состояние

**FUNAVAIL A,B,C,D,E,F,G,H**

Обеспечивает перевод устройства в недоступное для входа транзактов состояние.

Проверка состояния устройства может быть выполнена с помощью блока **GATE**.

**7. Многоканальные устройства**

МКУ служат для моделирования объектов, обладающих определенной емкостью. Емкость МКУ задают с помощью команды

STORAGE в сегменте описаний. Транзакт может занимать и освобождать определённую часть памяти.

**ENTER** A,B

Вход транзакта в МКУ

A – имя МКУ символическое или числовое;

B – число занимаемых каналов МКУ (необязательный операнд).

**LEAVE** A,B

Выводит транзакт из МКУ.

A – имя МКУ символическое или числовое;

B – число освобождаемых каналов МКУ (необязательный операнд).

**SAVAIL** A

Обеспечивает доступное состояние МКУ, имя или номер которого указаны в операнде A.

**SUNAVAIL** A

Обеспечивает недоступное состояние МКУ с именем, обозначенным операндом A.

## 8. Блоки регистрации транзактов в очереди

**QUEUE** A,B

Помещает транзакт в конец очереди.

A – числовое или символьное имя очереди;

B – число добавляемых к очереди элементов (необязательный операнд).

**DEPART** A,B

Удаляет транзакт из очереди.

A – имя (номер) очереди;

B – число удаляемых из очереди элементов (необязательный операнд).

## 9. Блоки, связанные с таблицами

**TABULATE** A,B

Заносит значение в таблицу.

A – имя или номер таблицы;

B – вес, указывающий сколько раз значение должно быть занесено в таблицу (необязательный операнд).

*Таблицы должны быть объявлены в сегменте описаний командами:*

*Name* **TABLE** A,B,C,D

*Определяет таблицу статистики параметра и гистограмму его плотности распределения*

*Name* – имя таблицы;

*A* – аргумент таблицы (табулируемый параметр): элемент данных, плотность распределения которого будет заноситься в таблицу;

*B* – верхняя граница нижнего интервала гистограммы;

*C* – ширина интервалов;

*D* – число интервалов (частотных классов);

Команды GPSS World

**BVARIABLE** – Определяет булеву переменную.

**CLEAR** – Сбрасывает статистику и удаляет транзакты.

**CONDUCT** – Выполняет зарегистрированный PLUS-эксперимент.

**CONTINUE** – Продолжает процесс моделирования после остановки.

**EQU** – Присваивает значение переменной пользователю.

**EXIT** – Завершает сеанс работы с GPSS World.

**FUNCTION** - Определяет функцию.

**FVARIABLE** – Определяет переменную с плавающей точкой.

**HALT** – Останавливает процесс моделирования и сбрасывает очередь команд.

**INCLUDE** – Читывает и транслирует дополнительные файлы модели.

**INITIAL** – Инициализирует или изменяет логический ключ, ячейку или матрицу.

**INTEGRATE** – Автоматически вычисляет интеграл для переменной пользователя.

**MATRIX** – Определяет матрицу.

**QTABLE** – Определяет таблицу статистики очереди (Q-таблицу).

**REPORT** – Устанавливает имя файла отчета или запрашивает немедленный отчет.

**RESET** – Сбрасывает статистику процесса моделирования.

**RMULT** – Устанавливает начальные числа первых семи генераторов случайных чисел.

**SHOW** – Вычисляет выражение и отображает результат.

**START** – Устанавливает счетчик завершения и запускает процесс моделирования.

**STEP** – Устанавливает ограниченное количество входов транзактов в блоки, после которого процесс моделирования переходит в состояние останова.

**STOP** – Устанавливает условие остановки, основанное на количестве попыток входа в блок.

**STORAGE** – Определяет многоканальное устройство.

**TABLE** – Определяет таблицу.

**VARIABLE** – Определяет переменную.

## Элементы стандартного отчета

Строка заголовка стандартного отчета содержит имя файла модели, который создал отчет. Также в нее включена дата и время прогона модели.

**1. Общая информация о результатах работы модели**

**STARTTIME** — начальное время. Абсолютное модельное время в момент начала моделирования. Устанавливается равным абсолютному модельному времени с помощью оператора RESET или CLEAR;

**END TIME** — конечное время. Абсолютное модельное время, когда счетчик завершения принимает значение 0;

**BLOCKS** — количество блоков, использованных в текущей модели, к моменту завершения моделирования;

**FACILITIES** — количество устройств, использованных в модели, к моменту завершения моделирования;

**STORAGES** — количество многоканальных устройств, использованных в текущей модели к моменту завершения моделирования.

**2. Информация об именах**

Файл статистики содержит информацию об именах, которые просматривает GPSS в ходе моделирования.

**NAME** — перечень заданных пользователем имен, содержащихся в модели;

**VALUE** — числовое значение, присваиваемое имени. Система начинает отсчет с 10 000;

**3. Информация о блоках**

**LABEL** — метка, алфавитно-цифровое имя данного блока (если оно задано);

**LOC** — числовой номер позиции данного блока в модели;  
**BLOCKTYPE** — тип блока GPSS;

ENTRY COUNT — количество транзактов, вошедших в данный блок, с начала работы программы или после последнего выполнения оператора RESET или CLEAR;

CURRENT COUNT — количество транзактов, находящихся в данном блоке к моменту завершения моделирования;

RETRY — количество транзактов, ожидающих специального условия, зависящего от состояния данного блока.

#### ***4. Информация об объектах типа «устройство»***

Элементы статистики, представленные в данном разделе, имеют следующее содержание:

FACILITY — имя или номер устройства;

ENTRIES — количество раз, когда устройство было занято или занято с прерыванием с начала моделирования или после последнего выполнения оператора RESET или CLEAR;

UTIL. — коэффициент использования, доля времени моделирования, в течение которого устройство было занято;

AVE. TIME - среднее время занятия устройства одним транзактом в течение времени моделирования с начала моделирования или после выполнения оператора RESET или CLEAR;

AVAIL . — состояние устройства в конце моделирования (равно 1, если устройство доступно; 0 — если недоступно);

OWNER — номер транзакта, который занимает устройство (0 — устройство не занято);

PEND — количество транзактов, вошедших в блоки PREEMPT в режиме прерывания;

INTER — количество транзактов, прерванных на данный момент транзактов в списке прерываний);

RETRY — количество транзактов, ожидающих выполнения

специального условия, зависящего от состояния данного устройства;

DELAY — количество транзактов, ожидающих занятия устройства (входят также транзакты, ожидающие занятия устройства в режиме прерывания с помощью блоков PREEMPT).

### **5. Информация об объектах типа «очередь»**

QUEUE — имя или номер очереди;

MAX — максимальное содержимое очереди в течение времени моделирования;

CONT. — текущее содержимое очереди в конце процесса моделирования; ENTRY — общее количество входов транзактов в очередь в течение времени моделирования.

ENTRY(O) — общее количество входов транзактов в очередь с нулевым временем ожидания;

AVE.CONT. — среднее значение содержимого очереди в течение времени моделирования;

AVE.TIME — среднее время пребывания одного транзакта в очереди с учетом всех входов в очередь;

AVE.(-O) — среднее время пребывания одного транзакта в очереди без учета «нулевых» входов в очередь;

RETRY — количество транзактов, ожидающих выполнения специального условия, зависящего от состояния очереди.

### **6. Информация об объектах типа «многоканальное устройство»**

STORAGE — имя или номер МКУ;  
CAP. — емкость МКУ, заданная оператором STORAGE;  
REM. — число единиц МКУ, свободных в конце моделирования;

MIN. — минимальное число единиц памяти, использовавшихся за период моделирования;

MAX. — максимальное число каналов МКУ, использовавшихся за

период моделирования;

ENTRIES — количество входов в МКУ за период моделирования;

AVL. — состояние МКУ в конце моделирования (1 — доступно; 0 — недоступно);

AVE.C — среднее значение занятой емкости за период моделирования;

UTIL. — коэффициент использования МКУ;

RETRY — количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данного МКУ;

DELAY — количество транзактов, ожидающих в блоках ENTER.

### ***7. Информация о таблицах***

TABLE — имя или номер таблицы или Q-таблицы;

MEAN — средневзвешенное значение табулируемого аргумента;

STD.DEV. — взвешенное среднеквадратическое отклонение:

$STD. DEV. = \sqrt{SOS / (COUNT - 1) - (SUM2 / (COUNT * (COUNT - 1)))}$ ,

SOS — накопленная сумма квадратов;

RANGE — нижний и верхний пределы частотного класса:

\* при попадании табулируемого аргумента в интервал, который имеет значение больше или меньше нижней границы частотного класса или равное верхней границе, изменяется значение частоты (FREQUENCY);

\* операнд В (весовой коэффициент) блока TABULATE может быть использован для определения величины, которая добавляется в частотный класс при табулируемого значения в этот частотный класс;

\* частотные классы, суммарное значение которых равно 0, в файл статистики не выводятся;

\* RETRY — количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной таблицы;

\* FREQUENCY — суммарная величина, которая формируется при

попадании табулируемого аргумента в указанные границы (значения операнда В суммируются блоком TABULATE);

\* CUM.% — величина частоты в процентах к общему табулируемого аргумента.

### ***Информация о списках пользователя***

USER CHAIN — имя или номер списка

SIZE — количество транзактов в списке пользователя;

RETRY — количество транзактов, ожидающих наступления специального условия, зависящего от состояния данного списка пользователя;

AVE. CONT. — среднее содержимое списка пользователя в течение времени моделирования;

ENTRIES — общее число транзактов, входивших в список в течение времени моделирования;

MAX — максимальное количество транзактов в списке пользователя за время моделирования;

AVE.TIME — среднее время пребывания транзакта в списке.

### ***9. Информация о числовых группах***

NUMERIC GROUP — имя или номер числовой группы  
GROUP SIZE — количество транзактов, содержащихся в числовой группе в конце моделирования;

RETRY — число транзактов, ожидающих выполнения специального условия, зависящего от состояния данной числовой группы.

### ***12. Информация о сохраняемых величинах (ячейках)***

SAVEVALUE — имя или номер ячейки;

VALUE — значение сохраняемой величины в конце моделирования;

RETRY — количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной ячейки.

### ***10. Информация о матрицах***

**MATRIX** – имя или номер матрицы;

**RETRY** – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной матрицы;

**INDICES** – до шести целых чисел, определяющих элемент матрицы

**VALUE** – значение элемента матрицы в конце моделирования (элементы, равные 0, выводятся в отчете группами).

## Индивидуальные задания

**Задание 1**

Клиенты парикмахерской делятся на три типа — клиенты, желающие только постричься, желающие только побриться и желающие постричься и побриться. Клиенты появляются через каждые  $20 \pm 5$  минут. Желающих постричься 60% от общего числа клиентов и из них 20% желают также побриться, желающих только побриться 40% от общего числа клиентов. Некоторые клиенты (20% от общего количества) имеют право на обслуживание вне очереди (по предварительной записи) и длина общей очереди их не волнует. Может образовываться льготная очередь и её длина важна для льготников. Новый клиент оценивает длину «своей» очереди и если она больше 2 человек, уходит, а если меньше — встаёт в «свою» очередь.

Продолжительность стрижки  $35 \pm 10$  минут, продолжительность бритья  $10 \pm 5$  минуты.

Необходима статистика очереди для всех клиентов. Моделируется рабочий день (480 минут). Дополнительным требованием является необходимость сбора информации о количестве клиентов разного типа — льготниках, не льготниках, обслуженных в течение дня, а также об ушедших без обслуживания.

**Задание 2**

В системе передачи данных СЗТУ осуществляется обмен пакетами данных между пунктами А и В по дуплексному (двухканальному) каналу связи. Пакеты поступают в пункты системы от абонентов с интервалами времени между ними  $8 \pm 3$  мс в пункт А и  $6 \pm 2$  мс в пункт В. Передача пакета занимает 10 мс. В обоих пунктах имеются буферные регистры-

накопители, которые могут хранить два пакета (включая передаваемый). В случае прихода пакета в момент занятости регистров пунктам системы предоставляется выход на спутниковую полудуплексную (одноканальную) линию связи, которая осуществляет передачу пакетов данных за  $10 \pm 5$  мс. При занятости спутниковой линии пакет получает отказ.

В 3% случаев при передаче по дуплексному каналу происходит сбой, о котором становится известно после передачи, и передачу пакета приходится повторять, для этого он направляется снова в регистр, но имеет повышенный приоритет. В спутниковой линии сбои происходят только в 2% случаев, пакет в этом случае также направляется в регистр с высоким приоритетом.

Смоделировать обмен информацией в системе передачи данных в течение 10 мин. Определить частоту вызовов спутниковой линии и ее загрузку. Определить процент потерянных пакетов. В случае возможности отказов определить необходимый для безотказной работы системы объем буферных регистров.

### **Задание 3**

На обрабатывающий участок цеха поступают детали в среднем через 50 мин. Первичная обработка деталей производится на одном из двух станков. Первый станок обрабатывает деталь в среднем 40 мин и имеет до 4% брака, второй — соответственно 60 мин и 8% брака. Все бракованные детали возвращаются на повторную обработку на второй станок. Детали, попавшие в разряд бракованных дважды, считаются отходами. Вторичную обработку проводят также два станка в среднем 100 мин каждый. Причем первый станок обрабатывает имеющиеся в накопителе после первичной обработки детали, а второй станок подключается при образовании в накопителе задела больше трех деталей. Все интервалы времени

распределены по экспоненциальному закону.

Смоделировать обработку на участке 500 деталей. Определить загрузку второго станка на вторичной обработке и вероятность появления отходов.

#### **Задание 4**

В аптеке есть два отдела: отдел готовых лекарств и рецептурный отдел. Посетители приходят в аптеку в среднем через каждые 2,5 минуты. Половина посетителей направляется в отдел готовых лекарств, после чего покидают аптеку. Вторая половина посетителей обращается в рецептурный отдел. В 50 % случаев клиент сразу же получает готовое лекарство по рецепту, в остальных случаях готового лекарства нет и клиент должен зайти еще раз за изготовленным лекарством. После рецептурного отдела примерно 20 % посетителей заходит еще и в отдел готовых изделий, остальные просто покидают аптеку. Время обслуживания в отделе готовых изделий равно в среднем 2-м минутам, в рецептурном отделе — 5-ти минутам.

#### **Задание 5**

Половине пациентов, которым назначен электрофорез нужно выполнить один курс, состоящий из пяти процедур, остальным нужно пройти два курса (10 процедур). Время процедуры составляет 8-12 минут, между пациентами нужно делать перерыв на 2-4 минуты. Расписание процедур составлено из расчета одна процедура каждые 15 минут. Каждому пациенту процедуры назначаются на одно время ежедневно, поэтому задержка между процедурами составляет 24 часа минус 15 минут и минус время ожидания в очереди. Впрочем, 20 часов простоя кабинета в сутки можно просто опустить, тогда время повтора процедуры составит 4 часа

минус 15 минут и время ожидания. Процедурный кабинет работает с 9 до 13.00. Определить, сколько нужно закупить аппаратов, чтобы обслуживать 100 пациентов в месяц, а среднее время ожидания не превышало бы 5 минут. Провести эксперимент с изменением количества каналов обслуживания от 1 до 10 и выявить оптимальное количество

### **Задание 6**

Студенты приходят на экзамен и по очереди входят в аудиторию. Количество билетов может быть меньше числа студентов в группе и поэтому может образоваться очередь за билетами. Имеется список студентов, которые входят в аудиторию случайным образом. Также случайным образом они берут билеты. Каждый билет включает два вопроса различной сложности. Затем студенты идут готовиться в помещение, достаточное для того, чтобы вместить всех, получивших билеты. После подготовки, зависящей от сложности билета и подготовленности студента, студенты выстраиваются в очередь к преподавателю. Они садятся поочередно к преподавателю и рассказывают свои вопросы. Продолжительность ответа на каждый вопрос составляет 4-10 минут. Качество ответа оценивается оценкой, которая зависит от подготовленности студента (но с точки зрения наблюдения является величиной случайной). Когда студент сел к преподавателю его билет освобождается и может быть передан очередному студенту.

Преподаватель делает перерыв на 5-10 минут каждые 1,5-2 часа.

Интерес могут представлять различные характеристики процесса: время ожидания студентами билетов (если билетов меньше, чем студентов), общее время экзамена, число студентов, сдавших экзамен на 2, 3, 4 и 5 и др.

### **Задание 7**

Желающие прокатиться на карусели подходят с интервалом 10-20

секунд и занимают очередь, если ее длина не превышает 20 человек, иначе уходят на другие аттракционы. На карусели 15 мест, но часть клиентов катается несколько раз подряд и не освобождают места. Количество таких желающих – равномерная распределенная величина от 0 до 5 человек. То есть после очередного заезда освобождаются не все места на карусели.

Посадка на карусель занимает 20-40 секунд.

Время заезда можно регулировать, но стоит учесть, что наш народ обмана не любит, и при слишком коротком времени заезда вероятность того, что клиент станет в очередь, резко уменьшается. При времени заезда до 100 секунд вероятность составляет не более 0,2; до 200 секунд – 0,4; 300 секунд – 0,8; 400 и более – 1 (задать вероятность можно непрерывной функцией).

Стоимость одного заезда равна 30 рублей, содержание карусели обходится 10000 рублей в день.

Определить оптимальное время заезда по критерию максимума прибыли. Моделировать в течение 30 рабочих дней по 12 часов работы в день.

### **Задание 8**

На склад готовой продукции предприятия каждые 1-2 мин поступают изделия. С интервалом времени 35-50 мин к складу подъезжают автомашины, в каждую из которых нужно погрузить эти изделия по 25 штук на одну машину. Погрузка осуществляется с одного терминала (одновременно под погрузку может стать только одна машина) в течение 12 часов без перерыва на обед и продолжается 5 мин. У склада одновременно могут находиться не более трех автомашин, включая автомашину стоящую под погрузкой и две в очереди. Автомашины, не нашедшие места у склада, уезжают с его территории без груза.

Подсчитать число автомашин, уехавших без груза. Определить

среднее и максимальное количество изделий, хранящихся на складе. Выработать рекомендации по постройке второго терминала. Хранение одного изделия на складе обходится 1 рубль в час, расширение площадки ожидания обойдется 100000 рублей. Окупится ли за год расширение площадки ожидания?

### Задание 9

Товары вывозятся с таможенного склада поддонами. В транспортном отделе поддоны загружаются в машины автоматической погрузочной машиной в количестве, заказанном получателями товаров. Размер заказа является дискретной случайной величиной с распределением:

Частота	.10	.25	.30	.15	.12	.05	.03
Размер заказа (поддонов)	6	12	18	24	30	36	48

Если поддонов меньше, чем заказано, машина ожидает доставки следующей партии со склада. К началу отгрузки уже имеется 1000 готовых поддонов. Время между поступлением заказов распределено экспоненциально со средним значением 15 минут. Время погрузки заказа составляет 120 секунд плюс 10 секунд на каждый поддон. Доставка со склада на площадку погрузки осуществляется партиями по 60 поддонов через каждые 455 минут.

Необходимо смоделировать 5 дней работы транспортного отдела, чтобы получить следующую информацию:

1. Среднее количество заказов, ожидающих выполнения в транспортном отделе.

2. Количество поддонов, отправляемых каждый день.

3. Есть ли необходимость увеличить или уменьшить начальный запас поддонов (1000 шт.)?

### **Задание 10**

Экспериментальная автоматическая гибкая производственная система имеет два станка с ЧПУ, зону приемки, промежуточный накопитель (на случай, если занят второй станок) и зону готовых изделий. Заготовки прибывают каждые 150 секунд (экспоненциальное распределение) и последовательно обрабатываются на двух станках. Роботу требуется  $8 \pm 1$  секунд, чтобы захватить или установить на станок заготовки, и 6 секунд, чтобы переместить их из зоны приемки к первому станку. Время обработки на первом станке распределено по нормальному закону со средним в 60 секунд и стандартным отклонением в 10 секунд. На перемещение от первого станка в промежуточный накопитель и из накопителя ко второму станку роботу требуется  $7 \pm 1$  секунд. Время обработки на втором станке составляет 100 секунд (экспоненциальное распределение). Чтобы переместить заготовки от второго станка в зону готовых изделий, роботу требуется  $5 \pm 1$  секунд.

Необходимо смоделировать работу системы для 1000 готовых изделий, а также:

1. Найти коэффициент использования робота и станков.
2. Определить требуемую емкость зоны приемки и промежуточного накопителя, чтобы хватало мест для размещения заготовок.

### **Задание 11**

Расходные материалы, необходимые для работы приборов технического контроля, находятся на региональном и центральном складах.

На региональном складе может храниться до 20 комплектов материалов, потребность в которых возникает через  $60 \pm 10$  часов и составляет один комплект. В случае понижения уровня запасов до  $k=3$  комплектов на протяжении 60 часов формируется запрос на пополнение запасов региональном складе до полного объема в  $n=20$  комплектов. Запрос посылают на центральный склад, где на протяжении  $80 \pm 20$  часов комплектуются материалы и через  $70 \pm 10$  часов доставляются на региональный склад. Следующий запрос на пополнение запасов может подаваться только после выполнения предыдущего.

Хранение одного комплекта на региональном складе требует 10 единиц стоимости за единицу времени. Штраф за задержку поставки комплекта составляет 150 единиц стоимости за единицу времени.

Определить, при каких значениях  $n$  и  $k$  достигается максимальная экономическая эффективность работы складов. Модельное время – один год.

### **Задание 12**

Поток самолетов, требующих посадки в аэропорту, - экспоненциальный с интенсивностью 12 самолетов в час. В аэропорту есть 2 посадочные полосы.

Самолет, совершив посадку на полосу, освобождает ее через 8 минут. Если самолет, требующий посадки, застает все полосы занятыми, то он становится в «очередь» самолетов, ожидая посадки. Каждые 10 минут он запрашивает разрешение на посадку.

Через  $80 \pm 15$  минут после затребования посадки самолет нуждается в дозаправке, что обходится аэропорту в  $55000 \pm 3900$  денежных единиц.

После 120 минут безуспешного ожидания самолет отправляется на посадку в другой аэропорт. За каждый самолет, совершивший посадку без

ожидания, аэропорт получает прибыль 4100 единиц стоимости. За каждый самолет, севший после ожидания -  $1000 \pm 250$  ед. стоимости. Эксплуатация одной посадочной полосы обходится в 5100000 денежных единиц в месяц.

Определить количество  $n$  посадочных полос, при котором достигается максимальная экономическая эффективность.

### **Задание 13**

В аэропорту имеется 3 стойки регистрации. К стойкам регистрации подходят пассажиры с багажом. Количество мест багажа у пассажиров следующее: у 5% пассажиров нет багажа, у 70% – одно место, у 20% – два места, у остальных – три. Только что подошедший пассажир подходит к свободной стойке регистрации для обслуживания, если такая имеется. Если свободных стоек регистрации на момент прихода пассажира нет, то он встает в конец очереди к той стойке регистрации, где очередь является самой короткой на этот момент. Сразу после обслуживания пассажира его багаж поступает на ленту транспортера. Багаж проходит проверку на интроскопах (сканеры). Имеется 2 интроскопа, что значит, что 2 места багажа могут быть проверены одновременно. Время проверки на интроскопе от 10 до 15 секунд

5% багажа не проходят проверку и идут на дополнительную проверку, которая занимает от 20 до 30 секунд. После дополнительной проверки 4.8% багажа, с которым все в порядке, возвращаются обратно на ленту, а 0.2% багажа сходят с ленты на ручной досмотр. Ручной досмотр занимает 5-7 минут. После выполнения всех проверок багаж с ленты распределяется на 4 разные погрузочные тележки. Перемещение до тележек и распределение занимают 3-5 минут.

Вычислите максимально возможное время пребывания пассажира с багажом в очереди на регистрацию, разницу между временем выстраивания

пассажира в очередь и текущим моментом, сколько времени багаж находится в системе.

#### **Задание 14**

Фирма наняла одного декларанта для таможенного оформления продовольственных товаров, бытовой техники и запчастей к автомобилям. Необходимость таможенного оформления продовольственных товаров возникает каждые  $40 \pm 8$  часов, оформление ручное и занимает  $10 \pm 1$  час. Это оформление выполняется в первую очередь. Необходимость в оформлении запчастей возникает каждые  $90 \pm 10$  часов, оформление также ручное и занимает  $15 \pm 5$  часов. Заявки на оформление бытовой техники поступают каждые  $5 \pm 1$  часов, их выполнение автоматизировано и занимает  $120 \pm 30$  минут. Половина заявок по бытовой технике поступает от постоянного клиента, поэтому обрабатываются в первую очередь.

Рабочий день декларанта с 10 до 19.00 часов с перерывом на обед с 13 до 14.00. Оформление заявок приносит доход 1000 рублей за каждую, содержание декларанта обходится в 50000 рублей в месяц.

1. Смоделировать работу в течение 150 дней.
2. Определить, стоит ли нанимать второго декларанта, т.е. есть ли перегруз у первого и принесет ли найм второго декларанта дополнительную прибыль.

#### **Задание 15**

Контейнеры с керамическими изделиями поступают в цех обжига (входной поток пуассоновский с параметром  $L$ ). Каждый контейнер содержит партию из 100 изделий, которые требуют одинакового времени обжига. Время обжига - равномерно распределенная величина в интервале  $A \pm B$ . В цехе находится печь, в которую одновременно загружают три контейнера. Время обжига соответствует наибольшему из времен,

необходимых для обжига изделий из этих трех контейнеров. Прибыль от обжига каждого изделия составляет  $S1$  единиц стоимости. Один час работы печи требует  $S2$  единиц стоимости (учитывается только «чистое» время работы печи).

Сравните экономическую эффективность следующих дисциплин обслуживания:

А. Контейнеры загружаются в печь по три по принципу FIFO. Для поддержки функционирования очереди необходимо  $S3$  единиц стоимости в час.

В. Контейнеры разделяются на две очереди: очередь с большим временем обжига и очередь с меньшим временем обжига изделий в печи. В печь загружаются по три контейнера из каждой очереди выбор осуществляется по принципу FIFO. Для поддержки этих двух очередей необходимо  $k1 \cdot S3$  единиц стоимости.

С. Контейнеры разделяются на три очереди: с «большим», «средним» и «меньшим» временем обжига изделий в печи. В печь загружаются по три контейнера из каждой очереди, выбор осуществляется по принципу FIFO. Для поддержки функционирования этих трех очередей необходимо  $k2 \cdot S3$  единиц стоимости.

Оценить интервалы значений  $k1$  и  $k2$  при которых дисциплины В и С становятся невыгодными.

$$A \pm B - 20 \pm 8$$

$$S1 - 5$$

$$S2 - 20$$

$$S3 - 11$$

$$L - 1/20$$

## **Задание 16**

Транспортный цех объединения, имеющий 8 грузовиков, обслуживает три филиала А, В и С. В момент начала работы грузовики находятся в филиале А. Грузовики перевозят изделия А в В и из В в С, возвращаясь потом в А без груза. Погрузка в А занимает в среднем 20 мин., переезд из А в В длится 30 мин., разгрузка(погрузка) в В-20 мин, переезд в С - 30 мин, разгрузка в С - 20 мин, и переезд в А - 20 мин(всюду распределение экспоненциальное).

Разработать модель работы цеха. Определить Оптимальное число грузовиков для обслуживания филиалов.

### **Задание 17**

На сборочный участок цеха из различных источников поступают детали двух типов – А и В. Детали поступают партиями по три штуки. В обоих случаях интервалы между приходами партий распределены экспоненциально со средним значением 10 минут. На сборку подаются деталь типа А и деталь типа В. Процесс сборки занимает примерно 3 минуты и осуществляется двумя сборщиками, работающими параллельно. Затем изделие АВ поступает на регулировку, продолжающуюся в среднем 4 минуты (время распределено экспоненциально). Регулировкой также занимаются два работника. После этого изделие готово к транспортировке.

Транспортировку осуществляет электрокар, приезжающий за готовыми изделиями каждые 40 минут. На погрузку одного изделия уходит примерно 5 секунд. Максимальное количество изделий, перевозимых за один рейс, - 20 штук. Если к моменту прихода электрокара к отправке готово меньшее количество изделий, он забирает их все и уезжает.

Постройте модель такой системы. Прогон завершите в момент ухода 50-го кара. Получите распределение случайной величины “число изделий, увозимых за один рейс”. При необходимости дайте рекомендации по изменению частоты движения транспорта. При возникновении очередей на

этапах сборки и регулировки укажите причины их появления и возможные пути устранения.

### **Задание 18**

Требуется разработать модель процессов возникновения и устранения неисправностей в некоторой технической системе, состоящей из множества однотипных блоков; в запасе имеется несколько исправных блоков; известны статистические данные:

интенсивность возникновения отказов – в среднем каждые 100 ЕМВ;

длительность операции поиска неисправностей – в среднем 15 ЕМВ;

замена отказавшего блока – 10 +-5 ЕМВ. Влечет за собой уменьшение запаса исправных блоков;

ремонт отказавшего блока – в среднем 50 ЕМВ, влечет пополнение запаса;

Поиск и замену отказавшего блока производит бригада TEAM1, а ремонт замененного блока — бригада TEAM2.

В случае отсутствия запаса процесс затормаживается до поступления отремонтированного блока.

Исходя из общего времени отсутствия запаса определить размер запаса в течение 100000 ЕМВ, при котором простоя не будет.

Начальный запас задать с помощью блока INITIAL Sklad,N, который запишет в ячейку Sklad количество запасных блоков.

## Основные формулы теории массового обслуживания

**Обозначения**

ОКУ – одноканальное устройство;

МКУ – многоканальное устройство;

$\lambda$  – интенсивность потока заявок;  $\lambda = 1/T_{cp}$ , где  $T_{cp}$  – средний интервал времени между поступлениями заявок;

$\mu$  – интенсивность обслуживания;  $\mu = 1/T_{обсл}$ , где  $T_{обсл}$  – среднее время обслуживания одной заявки одним каналом;

$n$  – количество каналов обслуживания МКУ;

$m$  – количество мест в очереди;

$\rho = \lambda / \mu$  – коэффициент загрузки канала (ОКУ);

$\chi = \rho / n$  – коэффициент загрузки МКУ, где  $n$  – количество каналов МКУ

**Примечание:** системы с отказами (очередь отсутствует, при занятом устройстве заявка получает отказ) описываются теми же формулами при  $m = 0$ .

Обозначение	Пояснение	ОКУ с ограниченной длиной очереди	ОКУ с неограниченной длиной очереди	МКУ с ограниченной длиной очереди	МКУ с неограниченной длиной очереди
$P_0$	Вероятность того, что система полностью свободна	$\frac{1 - \rho}{1 - \rho^{m+2}}$	$1 - \rho$	$\left[ \sum_{k=0}^n \frac{\rho^k}{k!} + \frac{\rho^n \chi - \chi^{m+1}}{n! (1 - \chi)} \right]^{-1}$	$\left[ \sum_{k=0}^n \frac{\rho^k}{k!} + \frac{\rho^{n+1}}{n! (n - \rho)} \right]^{-1}$
$P_{отк}$	Вероятность отказа в обслуживании	$\frac{\rho^{m+2}(1 - \rho)}{1 - \rho^{m+2}}$	0	$\frac{\rho^{n+m} P_0}{n^m n!}$	0
$P_{обсл}$	Вероятность обслуживания	$1 - P_{отк}$	1	$1 - P_{отк}$	1
$Q = P_{обсл}$	Относительная пропускная способность	$1 - \frac{\rho^{m+2}(1 - \rho)}{1 - \rho^{m+2}}$	1	$1 - P_{отк}$	1
$A = \lambda Q$	Абсолютная пропускная способность	$\lambda Q$	$\lambda$	$\lambda Q$	$\lambda$
$L_{оч}$	Среднее количество заявок в очереди	$\frac{\rho^2(1 - (m + 1 - m\rho)\rho^m)}{(1 - \rho)(1 - \rho^{m+2})}$	$\frac{\rho^2}{1 - \rho}$	$\frac{\rho^{n+1} P_0}{n n!} \frac{1 - \chi^m(m + 1 - m\chi)}{(1 - \chi)^2}$	$\frac{\rho^{n+1} P_0}{n n!} \frac{1}{(1 - \chi)^2}$

$L_{\text{смо}}$	Среднее количество заявок в системе обслуживания	$L_{\text{оч}} + \frac{\rho - \rho^{m+2}}{1 - \rho^{m+2}}$	$\frac{\rho}{1 - \rho}$	$L_{\text{оч}} + n_3$	$L_{\text{оч}} + n_3$
$T_{\text{ож}}$	Среднее время ожидания в очереди	$\frac{L_{\text{оч}}}{\lambda}$	$\frac{\rho}{\mu(1 - \rho)}$	$\frac{L_{\text{оч}}}{\lambda}$	$\frac{L_{\text{оч}}}{\lambda}$
$T_{\text{обсл}}$	Среднее время обслуживания в устройстве	$\frac{Q}{\mu}$	$\frac{1}{\mu}$	$\frac{Q}{\mu}$	$\frac{1}{\mu}$
$T_{\text{смо}} = T_{\text{ож}} + T_{\text{обсл}}$	Среднее время нахождения заявки в системе	$\frac{L_{\text{оч}}}{\lambda} + \frac{Q}{\mu}$	$\frac{1}{\mu(1 - \rho)}$	$\frac{L_{\text{оч}}}{\lambda} + \frac{Q}{\mu}$	$\frac{L_{\text{оч}}}{\lambda} + \frac{1}{\mu}$
$n_3$	Среднее количество занятых каналов	$\rho$	$\rho$	$\rho \left[ 1 - \frac{\rho^n \chi^m P_0}{n!} \right]$	$\rho$
$K_3$	Коэффициент занятости устройства	$\frac{\rho - \rho^{m+2}}{1 - \rho^{m+2}}$	$c$	$\frac{n_3}{n}$	$\frac{n_3}{n}$

АНАЛИЗ ГЕОСИСТЕМ  
В СРЕДАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Часть 1. Среда GPSS

УЧЕБНОЕ ПОСОБИЕ

Авторы

Т.А. Нигматулин, А.Ю. Сидоренко, М.А. Авдошина

Издано в авторской редакции

Подписано в печать 03.12.2025 г.

Формат 60×84/16.

Усл. печ. л. 12. Тираж – Электронное издание

СПб. ООО «Свое издательство», 191040, Санкт-Петербург,  
Пушкинская ул, д. 10 литера А, помещ. 1-н  
2025