

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

Кафедра информационных технологий автоматизированных систем

С.С. Смородинский, Н.В. Батин

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

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

В 2-х частях

Часть 2

Минск 2005

УДК 681.3 + 007 (075.8)
ББК 32.97 я 73
С 51

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

Сморodinский С.С.

С 51 Оптимизация решений на основе компьютерных имитационных методов и моделей: Учеб. пособие по курсу “Моделирование систем” для студ. спец. “Автоматизированные системы обработки информации” дневн. и дистанц. форм обуч. В 2 ч. Ч.2/ С.С. Смородинский, Н.В. Батин. - Мн.: БГУИР, 2005.- 105 с.
ISBN 985-444-890-8 (ч.2)

В пособии приводится теоретический и практический материал, связанный с анализом и оптимизацией решений на основе имитационного моделирования. Во второй части пособия рассматриваются процедуры имитации технических, информационных и экономических систем на основе системы имитационного моделирования GPSS World.

Пособие рекомендуется использовать при изучении курсов, связанных с применением методов компьютерного моделирования и поддержки принятия решений, в курсовом и дипломном проектировании. Пособие представляет интерес для специалистов, практическая деятельность которых связана с решением задач компьютерного моделирования в различных сферах науки, техники, экономики.

УДК 681.3 + 007 (075.8)
ББК 32.97 я 73

ISBN 985-444-890-8 (ч.2)
ISBN 985-444-891-6

© Смородинский С.С., Батин Н.В., 2005
© БГУИР, 2005

“В тот момент, когда вы решили использовать GPSS World, вы присоединились к многотысячному обществу специалистов, практикующих имитационное моделирование по всему миру. GPSS – это язык, глубоко уходящий своими корнями в начало 1960-х, когда имитационное моделирование только зарождалось. Ни один другой язык моделирования не стал таким значимым.”

*Спрингер Кокс, Президент компании
Minuteman Software, 2002*

<http://www.minutemansoftware.com/>

“Система GPSS World работает на персональных компьютерах под управлением ОС Windows и максимально ориентирована на использование современных технологий, обеспечивающих высокую интерактивность и визуальное представление информации. Удобная многооконная архитектура, система понятных диалоговых окон, простой и наглядный интерфейс пользователя делают GPSS World чрезвычайно удобным и мощным инструментом.”

*В.В. Девятков, руководитель работ
по локализации GPSS World
на русский язык, 2002*

<http://www.elina-computer.ru/>

ВВЕДЕНИЕ

Во второй части учебного пособия по дисциплине “Моделирование систем” рассматриваются особенности моделирования и оптимизации решений в среде новой перспективной системы имитационного моделирования GPSS World, разработанной компанией Minuteman Software (США).

GPSS World – это мощная среда компьютерного моделирования общего назначения, разработанная для профессионалов в области моделирования. Это комплексный моделирующий инструмент, охватывающий области как дискретного, так и непрерывного компьютерного моделирования, обладающий высоким уровнем интерактивности и визуального представления информации. GPSS World представляет собой наиболее удачную разработку из множества клонов GPSS, появившихся в последние годы за рубежом. Данная система является развитием широко известной системы GPSS/PC, причем основа языка GPSS World практически полностью соответствует “классической” версии языка GPSS [12].

Компьютерное моделирование, или имитация (Computer Simulation, или Simulation) – это в сущности компьютерный эксперимент, причем единственное отличие подобного эксперимента от реального состоит в том, что он проводится с моделью исследуемой системы, а не с самой системой. Имитация представляет собой практически единственный реалистический метод исследования

структурно-сложных систем. Метод имитационного моделирования входит в научный инструментарий системного анализа и является мощным средством анализа и оптимизации систем, операций и процессов. Имитационное моделирование применяется на практике для исследовательских целей (решение научных, технических, экономических или других задач), для целей обучения (игровые компьютерные программы и тренажеры), для целей замещения (реальный объект замещается имитатором в виде компьютерной программы).

К имитации прибегают обычно в следующих типичных ситуациях: 1) когда объектом исследования является перспективная система, которой в реальности еще не существует (в развитых странах ни один крупный проект не реализуется без его проверки на имитационной модели); 2) когда эксперименты с дорогостоящей и уникальной системой трудно осуществимы, требуют больших материальных затрат или нецелесообразны по каким-либо причинам (моделирование заменяет натурные испытания в реальных условиях, которые экономически не оправданы); 3) когда исследуются системы или процессы, которые удалены в пространстве или во времени, а также ненаблюдаемы вследствие их размеров или длительности существования и развития (анализ характеристик пространственно-распределенной компьютерной сети, надежности технологического модуля, быстропротекающих динамических процессов и т.д.); 4) когда исследуются свойства, закономерности и особенности систем или процессов с целью решения практических оптимизационных задач на микро- и макроуровнях; 5) когда исследуется влияние тех или иных факторов, а также управленческих решений на процессы в социально-экономических системах.

Основные технологические операции в системе имитационного моделирования GPSS World: 1) подготовка исходного текста имитационной программы (модели) на языке GPSS World; 2) трансляция исходного текста имитационной программы в компьютерный объектный код; 3) выполнение имитационной программы и оценка выходных характеристик модели; 4) формирование отчетов о состоянии модели и результатах имитации.

Во второй части учебного пособия рассматривается методология и технология имитационного моделирования в среде GPSS World, приводятся примеры имитационных компьютерных программ. Материал пособия служит для поддержки дисциплины “Моделирование систем” (цикла лабораторных работ и курсового проектирования). В приложениях приводится справочный материал по системе имитационного моделирования GPSS World. Рекомендуемая литература включает современные источники по системе имитационного моделирования GPSS World [1-4] и языку моделирования GPSS [10-12], источники по алгоритмическим основам методов Монте-Карло и имитационного моделирования [4-5, 6-9]

1. СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD: КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ И ПРИНЦИПЫ РАБОТЫ

1.1. Назначение и принципы работы системы моделирования GPSS World

Язык имитационного моделирования GPSS представляет собой набор команд и операторов для описания объекта моделирования (моделируемой системы). Под объектом моделирования будем понимать любой объект, который требуется исследовать с помощью модели. Таким объектом может быть производственное предприятие, транспортная система, торговая организация и т.д.

На основе языка имитационного моделирования GPSS разработан ряд систем имитационного моделирования. Система имитационного моделирования – это программное средство, обеспечивающее все этапы работы с моделью: подготовку и редактирование текста модели, отладку модели, моделирование, получение результатов моделирования. В данном пособии рассматривается одна из таких систем – GPSS World. Структура системы имитационного моделирования GPSS World приведена в прил.1.

Основное назначение языка моделирования GPSS и разработанной на его основе системы GPSS World - имитационное моделирование *дискретных динамических* систем.

Под *динамической* системой понимается любая система, состояние которой изменяется во времени. *Дискретная* динамическая система - это любая система, состояние которой изменяется *дискретно* (скачкообразно) в *дискретные* (отстоящие друг от друга) *моменты времени*, когда происходят какие-либо события. Типичный пример дискретной динамической системы - любой объект, который может быть представлен в виде системы массового обслуживания (СМО). Состояние СМО (например, количество заявок, находящихся на обслуживании или в очереди) изменяется в моменты времени, когда очередная заявка поступает на обслуживание или завершает его; эти моменты разделены интервалами времени (случайными или постоянными), т.е. являются дискретными. Изменения состояний СМО также являются дискретными: например, количество заявок в СМО может изменяться только на целые числа (1, 2 и т.д.), но не на промежуточные величины.

Язык имитационного моделирования GPSS в наибольшей степени подходит для моделирования систем, которые могут быть представлены в виде СМО. В языке GPSS имеются специальные средства для моделирования потоков заявок, одноканальных и многоканальных узлов СМО, очередей и т.д. Язык GPSS позволяет моделировать практически любые СМО: одноканальные и многоканальные, с неограниченными очередями, с отказами, с ограничениями на очередь, разомкнутые и замкнутые и т.д. Многие основные характеристики СМО (коэффициенты загрузки узлов, длины очередей и т.д.) автоматически определяются в процессе моделирования и выводятся в составе выходных данных модели. В то же время с помощью языка GPSS могут решаться задачи моделирования систем, для которых обычно не используется описание в виде СМО.

Работа языка GPSS основана на использовании метода Монте-Карло, рассмотренного в первой части данного пособия. В большинстве случаев операции метода Монте-Карло (обращения к генераторам случайных чисел, проверки условий и т.д.) выполняются в языке GPSS автоматически, т.е. они скрыты от пользователя. Однако при необходимости пользователь сам имеет возможность реализовать необходимые операции метода Монте-Карло.

1.2. Работа с системой моделирования GPSS World.

Пример GPSS-модели

Сеанс моделирования с использованием системы GPSS World включает следующие основные этапы:

- вызов системы GPSS World;
- подготовка текста модели или загрузка имеющейся модели (файл с расширением .GPS);
- компиляция модели и создание файла модели в машинных кодах (файл с расширением .SIM);
- установка начального значения счетчика завершений (величины, с помощью которой задается условие окончания моделирования) и запуск модели на выполнение;
- получение файла-отчета, т.е. файла результатов моделирования (файл с расширением .GPR);
- сохранение файла-отчета (если содержащиеся в нем результаты представляют интерес);
- печать текста модели и файла-отчета или их копирование в окно Word для обработки и подготовки к печати (если это требуется);
- закрытие окон файла-отчета (.GPR) и файла модели в машинных кодах (.SIM).

Рассмотрим основные из этих действий более подробно.

Вызов системы моделирования GPSS World выполняется обычным образом (с помощью кнопки **Пуск**). После этого, если требуется начать разработку новой модели, то следует выбрать из меню команду **File - New**. Если требуется загрузить уже имеющуюся модель, то следует выбрать команду **File - Open**.

Подготовка текста модели

Пусть требуется разработать новую модель. После выбора команды **File - New** на экран выводится окно **Новый документ**. В нем следует выбрать элемент **Model** и нажать кнопку **ОК**. После этого на экран выводится окно для создания новой модели.

Модель в системе моделирования GPSS World (далее - GPSS-модель) представляет собой последовательность команд и операторов (блоков) языка GPSS, описывающих объект моделирования. Полный перечень этих команд и операторов приведен в прил.2 и 3.

Рассмотрим пример GPSS-модели для решения следующей задачи.

Пример 1.1. На станок поступают детали для обработки. Интервалы между моментами поступления деталей составляют от 5 до 10 мин. Обработка детали на станке занимает от 6 до 8 мин. Требуется разработать GPSS-модель для имитации обработки 100 деталей на станке.

GPSS-модель для этой задачи имеет следующий вид.

```
GENERATE 7.5,2.5
QUEUE OCH
SEIZE STANOK
DEPART OCH
ADVANCE 7,1
RELEASE STANOK
TERMINATE 1
```

Здесь оператор `GENERATE 7.5,2.5` имитирует поступление заявок (в данном примере - деталей) через интервалы, составляющие $7,5 \pm 2,5$ единицы времени, т.е. от 5 до 10 мин. Оператор `QUEUE OCH` имитирует поступление заявки в очередь, имеющую имя `ОСН` (конечно, можно использовать любое другое имя очереди). По смыслу задачи эта очередь представляет собой накопитель перед станком, предназначенный для деталей, ожидающих обработки. Оператор `SEIZE STANOK` имитирует поступление заявки в устройство с именем `STANOK` (имя устройства также может быть любым). Оператор `DEPART OCH` имитирует выход детали из очереди (т.е. из накопителя перед станком). Оператор `ADVANCE 7,1` имитирует задержку детали на 7 ± 1 единицу времени, т.е. на время от 6 до 8 мин (обработка на станке). Оператор `RELEASE STANOK` имитирует выход детали со станка. Оператор `TERMINATE 1` имитирует окончание моделирования детали (ее удаление из модели); при этом из *счетчика завершений* вычитается единица (т.е. величина, указанная в команде `TERMINATE`).

Счетчик завершений - величина, задаваемая пользователем в начале моделирования. В ходе моделирования счетчик завершений должен уменьшаться (его уменьшение должно быть предусмотрено в тексте модели). Когда счетчик завершений становится равным нулю, моделирование завершается. Более подробно использование счетчика завершений рассматривается ниже.

Примечание. В системе моделирования GPSS вместо термина "заявка", "деталь" и т.д. используется термин "транзакт".

В ходе подготовки модели необходимо время от времени выполнять ее сохранение с помощью команды меню **File - Save**. Для имени файла GPSS-модели по умолчанию используется расширение `.GPS`.

Компиляция модели

По окончании подготовки текста модели необходимо выполнить его компиляцию, т.е. преобразование в машинные коды. Для этого используется команда меню **Command - Create Simulation**. Создается файл в машинных кодах. Его имя образуется автоматически на основе имени исходного файла (т.е. файла GPSS-модели), расширение - `.SIM`.

Если компиляция или моделирование прерывается из-за ошибок в модели, то следует по выведенному сообщению определить допущенную ошибку, закрыть окно созданного .SIM-файла, перейти в окно модели, внести необходимые исправления и снова выполнить компиляцию модели.

Установка начального значения счетчика завершений и запуск модели

Для запуска модели необходимо выбрать команду меню **Command - Start**. На экран выводится окно **Start Command**, в котором указывается команда **START** и начальное значение счетчика завершений. По умолчанию предлагается команда **START 1**, т.е. начальное значение счетчика завершений предлагается принять равным единице. Для данного примера начальное значение счетчика завершений должно быть равно 100. Поэтому требуется указать новое значение счетчика завершений, т.е. ввести команду **START 100**. Затем можно начать процесс моделирования, нажав кнопку **ОК**.

В процессе моделирования при *каждом* выполнении оператора **TERMINATE 1**, имитирующего окончание обработки детали, счетчик завершений *уменьшается на единицу*. После выхода 100 транзактов (т.е. после имитации обработки 100 деталей) счетчик завершений оказывается равным нулю. На этом моделирование завершается.

Получение файла-отчета

По окончании моделирования создается файл-отчет с результатами моделирования. Его имя образуется автоматически на основе имени файла GPSS-модели; расширение - .GPR. Обычно следует сохранить этот файл (командой меню **File - Save**), а также скопировать его содержимое в окно текстового редактора Word для обработки и последующей печати. Чтобы скопировать содержимое отчета, следует выделить его (с помощью мыши или нажатием клавиш **Ctrl-A**), нажать на кнопку **Копировать** (или выбрать команду меню **Edit - Copy**), затем перейти в окно редактора Word (т.е. в файл с расширением .DOC) и нажать кнопку **Вставить**, или выбрать команду **Правка - Вставить**. Аналогично можно скопировать в окно Word текст модели.

После этого обычно следует закрыть файл-отчет, а также файл в машинных кодах (.SIM). Сохранять файл в машинных кодах обычно не требуется.

Примечание. Следует обратить внимание, что по окончании моделирования в среде GPSS World имеются *три* открытых окна: текст модели (файл с расширением .GPS), модель в машинных кодах (.SIM) и файл-отчет (.GPR). При выполнении любых операций (сохранение, копирование в Word, закрытие и т.д.) необходимо внимательно следить за тем, чтобы текущим было именно то окно, для которого требуется выполняемая операция.

1.3. Интерпретация результатов моделирования

Файл-отчет, созданный системой моделирования GPSS World, содержит информацию о различных объектах GPSS-модели (устройствах, очередях и т.д.). Кроме того, в файле-отчете содержатся некоторые внутренние данные о

работе системы моделирования, обычно не представляющие практического интереса; их можно удалить из отчета.

Ниже приведены данные из отчета для примера 1.1, представляющие интерес для анализа результатов моделирования.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	QUEUE	100	0	0
	3	SEIZE	100	0	0
	4	DEPART	100	0	0
	5	ADVANCE	100	0	0
	6	RELEASE	100	0	0
	7	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
STANOK	100	0.910	6.997	1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
OCH	1	0	100	46	0.119	0.912	1.689	0

Раздел отчета, включающий колонки LABEL, LOCK, BLOCK TYPE, ENTRY COUNT, CURRENT COUNT и RETRY, представляет собой **информацию об операторах модели**:

- LABEL - метка оператора (в примере 1.1 метки не использовались);
- LOC - порядковый номер оператора;
- BLOCK TYPE - имя оператора;
- ENTRY COUNT - количество транзактов, поступивших на оператор;
- CURRENT COUNT - количество транзактов, находящихся в операторе в конце моделирования;
- RETRY - внутренние данные о работе системы моделирования GPSS World, обычно не представляющие интереса для анализа результатов моделирования.

Информация об операторах может представлять интерес для анализа объекта моделирования. Например, значение колонки ENTRY COUNT для оператора GENERATE представляет собой количество транзактов, поступивших на обслуживание; значение этой же колонки для оператора SEIZE - количество транзактов, вошедших в устройство для обслуживания. Значение колонки CURRENT COUNT для оператора QUEUE указывает на количество транзактов, оставшихся не обслуженными к моменту окончания моделирования; большое значение этой величины может указывать на перегрузку устройства.

В данном примере, как и следовало ожидать, обработано 100 транзактов (деталей).

Информация об операторах может также представлять интерес для отладки GPSS-модели. Например, нулевое значение колонки ENTRY COUNT для какого-либо оператора может указывать на ошибку в модели: транзакты по какой-то причине не поступают в этот оператор. Большая величина в колонке CURRENT

COUNT также может указывать на ошибку: транзакты почему-то не могут выйти из этого оператора и перейти в следующий.

Информация об устройствах и очередях представляет собой набор характеристик объекта моделирования. На основе этой информации могут рассчитываться другие характеристики (например, стоимостные).

Раздел отчета FACILITY содержит **информацию об устройствах**:

- FACILITY - имя устройства;
- ENTRIES - количество транзактов, входивших в устройство;
- UTIL. - коэффициент загрузки устройства (т.е. доля времени, в течение которого в устройстве находились транзакты);
- AVE.TIME - среднее время пребывания транзакта в устройстве;
- OWNER - номер транзакта, находящегося в устройстве в момент окончания моделирования (если устройство свободно, то указывается ноль);
- AVAIL, PEND, INTER, RETRY, DELAY - внутренние данные о работе системы моделирования GPSS, обычно не представляющие интереса для анализа результатов моделирования.

Раздел отчета QUEUE содержит **информацию об очередях**:

- QUEUE - имя очереди;
- MAX - максимальная длина очереди (т.е. максимальное количество транзактов в ней) за время моделирования;
- CONT. - длина очереди в момент окончания моделирования;
- ENTRIES - количество транзактов, входивших в очередь;
- ENTRIES (0) - количество транзактов, которым не потребовалось ждать в данной очереди (*нулевые входы*);
- AVE.CONT. - средняя длина очереди (среднее количество транзактов в очереди);
- AVE.TIME - среднее время пребывания транзактов в очереди;
- AVE.(−0) - среднее время пребывания транзактов в очереди без учета нулевых входов, т.е. среднее время пребывания в очереди для тех транзактов, которым потребовалось *ждать* в ней *хотя бы некоторое время*;
- RETRY - внутренние данные о работе системы моделирования GPSS World, обычно не представляющие интереса для анализа результатов моделирования.

В данном примере, как и было задано в постановке задачи, обработано 100 деталей. Коэффициент загрузки станка составил 0,91, или 91%. Это означает, что станок работает в течение 91% времени, а 9% времени - простаивает. Среднее время ожидания детали в накопителе перед станком (т.е. в очереди) составило 0,912 мин. В накопителе никогда не оказывалось более одной детали (максимальная длина очереди - 1), а в среднем в накопителе находилось 0,119 детали. Из 100 обработанных деталей для 46 не потребовалось ожидания в накопителе (в момент их поступления станок оказывался свободным). Таким об-

разом, ожидание в накопителе потребовалось для 64 деталей; для них среднее время ожидания в накопителе составило 1,689 мин. Среднее время технологического цикла (т.е. время от поступления детали в накопитель до окончания ее обработки на станке) составило $6,997 + 0,912 = 7,909$ мин.

Таким образом, недостатком моделируемой системы является некоторая перегрузка станка: его коэффициент загрузки составил 91%, а желательное значение этого коэффициента - от 75 до 85%. Для устранения этого недостатка можно предложить уменьшить поток деталей (т.е. увеличить интервалы времени между моментами их поступления), использовать более совершенный станок (обрабатывающий детали быстрее) или использовать два станка. Однако необходимо учитывать, что первый из предложенных вариантов приводит к сокращению производства деталей, а второй и третий - требуют дополнительных затрат.

2. СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD: ОСНОВНЫЕ ВОЗМОЖНОСТИ

2.1. Моделирование потоков событий. Транзакты

Как отмечено выше, для всех операций моделирования в GPSS-моделях используются *транзакты*. Транзакт - объект GPSS-модели, который может в ходе моделирования создаваться, перемещаться (переходить из оператора на оператор), задерживаться на заданное время, уничтожаться (удаляться из модели). Транзакты имитируют различные реальные объекты или явления: поступающие на обслуживание заявки (детали, задачи, сигналы, транспортные средства и т.д.), сбои и отказы элементов моделируемой системы, моменты наступления различных событий (например, окончание рабочего дня) и т.д. Так как транзакты используются во всех GPSS-моделях, рассмотрим оператор, применяемый для их имитации, более подробно.

Транзакты создаются (генерируются) следующим оператором:

GENERATE A,B,C,D,E

где A - средний интервал времени между моментами создания транзактов;

B - отклонение от среднего интервала между транзактами;

C - интервал времени между началом моделирования и моментом создания первого транзакта;

D - общее количество создаваемых транзактов;

E - приоритет создаваемых транзактов.

Если в качестве операндов A и B указаны конкретные числа (как в примерах, рассмотренных выше), то интервал времени между транзактами будет представлять собой величину $A \pm B$ (т.е. случайную величину, распределенную по равномерному закону в диапазоне от $A-B$ до $A+B$). Если в качестве операндов A и B указаны ссылки на переменные или функции, то интервал времени между транзактами вычисляется как $A \cdot B$.

Примечание. Имитация более сложных интервалов времени (распределенных по законам, отличным от равномерного) рассматривается ниже.

Если операнд В не указан, то интервал времени между транзактами *в точности равен* величине, указанной в операнде А.

Операнд С используется редко. Если он не указан, то интервал между началом моделирования и созданием первого транзакта определяется по значениям операндов А и В.

Операнд D используется, если требуется смоделировать ограниченное количество каких-либо объектов или явлений. Если операнд D не указан, то создание транзактов прекращается только по окончании моделирования.

Операнд Е задает приоритет транзакта (целое число). По умолчанию приоритет равен 0. Большее значение соответствует более высокому приоритету. Приоритеты обычно используются при имитации СМО с заявками различной важности.

В одной GPSS-модели может быть несколько операторов GENERATE. Во время моделирования они имитируют потоки транзактов независимо друг от друга.

2.2. Структура GPSS-модели

GPSS-модель состоит из двух основных разделов: команд и операторов (блоков). Раздел команд имеется в модели не всегда, раздел операторов – всегда. Перечень команд языка GPSS приведен в прил.2, перечень операторов – в прил.3.

Раздел команд приводится в начале модели. В этом разделе обычно содержится *объявление* элементов GPSS-модели: многоканальных устройств, функций, переменных и т.д. В примере 1.1 нет раздела команд, так как используемые в модели элементы языка GPSS (одноканальное устройство и очередь) не требуют объявления.

Примечание. В некоторых случаях команды GPSS могут указываться также в конце модели. Кроме того, некоторые команды могут как указываться в тексте модели, так и выбираться из меню. Такие случаи обычно возникают в достаточно сложных задачах, поэтому они не рассматриваются в данном пособии.

Раздел операторов практически всегда начинается с оператора GENERATE.

Общая структура команд и операторов языка GPSS следующая:

МЕТКА КЛЮЧЕВОЕ_СЛОВО ОПЕРАНДЫ

Метки команд – это обычно имена объявляемых элементов модели. Метки операторов используются для ссылок на них в тексте модели.

Ключевые слова – это имена команд и операторов. Операнды – имена элементов модели (устройств, очередей и т.д.), а также числовые величины. Например, в операторе GENERATE 7.5,2.5 слово GENERATE – ключевое слово, а 7.5 и 2.5 – операнды.

Метка, ключевое слово и операнды отделяются друг от друга произвольным количеством пробелов. Если операндов несколько, то они разделяются запятыми. Между операндами *не должно быть* пробелов.

В любом месте в тексте GPSS-модели могут располагаться комментарии. Строка комментария должна начинаться со звездочки (*) или точки с запятой (;).

Комментарии могут также содержаться в строках команд и операторов языка GPSS. Текст комментария помещается после последнего операнда и отделяется от него точкой с запятой.

В тексте модели (командах, операторах и комментариях) могут использоваться как строчные, так и заглавные буквы.

2.3. Моделирование многоканальных СМО

Пример 2.1. Пусть в условиях, указанных в примере 1.1, используются два одинаковых станка. Перед станками имеется общий накопитель. Деталь, поступившая на обработку, направляется на станок, оказавшийся в этот момент свободным. Если оба станка заняты, то деталь направляется в накопитель. Как только какой-либо из станков освобождается (т.е. заканчивает обработку предыдущей детали), деталь из накопителя направляется на этот станок.

Такая система представляет собой двухканальную СМО. GPSS-модель для ее имитации имеет следующий вид.

```
STANOK STORAGE      2
      GENERATE      7.5,2.5
      QUEUE         OCH
      ENTER         STANOK
      DEPART        OCH
      ADVANCE       7,1
      LEAVE         STANOK
      TERMINATE     1
```

Как и в примере 1.1, эта модель запускается на выполнение командой **START 100**.

Здесь **STORAGE** - команда объявления многоканального устройства (в данном примере это устройство имеет имя **STANOK**). После слова **STORAGE** указывается количество каналов. Команда **STORAGE** (как и другие команды) должна указываться в начале модели, до первого оператора **GENERATE**. Вход транзакта (заявки) в многоканальное устройство имитируется оператором **ENTER**, а выход - оператором **LEAVE**.

Примечание. Операторы **ENTER** и **LEAVE** могут иметь второй операнд, обозначающий количество каналов, занимаемых транзактом в многоканальном устройстве. Например, если бы требовалось указать, что для выполнения некоторой работы необходимо использовать *оба* станка, то использовались бы операторы **ENTER STANOK, 2** и **LEAVE STANOK, 2**.

В файле-отчете, создаваемом по окончании моделирования, автоматически выводятся характеристики многоканального устройства. Для данного примера они будут иметь следующий вид:

	STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STANOK	2	2	0	2	100	1	0.910	0.455	0	0	

В разделе STORAGE содержится следующая информация о многоканальных устройствах:

- STORAGE - имя многоканального устройства;
- CAP. - количество каналов;
- REM. - количество свободных каналов в момент окончания моделирования;
- MIN. - минимальное количество занятых каналов за время моделирования;
- MAX. - максимальное количество занятых каналов за время моделирования;
- ENTRIES - количество транзактов, входивших в многоканальное устройство;
- AVE.C. - среднее количество занятых каналов;
- UTIL. - коэффициент загрузки многоканального устройства;
- AVL, RETRY, DELAY - внутренние данные о работе системы моделирования GPSS World, обычно не представляющие интереса для анализа результатов моделирования.

В данном примере коэффициент загрузки составил всего 45,5%. Таким образом, группа из двух станков явно недогружена: имеются значительные простои. Этот же вывод следует из того, что среднее количество занятых каналов составляет 0,910 (т.е. в среднем занято менее одного станка из двух имеющихся). Таким образом, из двух вариантов производственной системы, рассмотренных в примерах 1.1 и 2.1, более предпочтительным является первый вариант (его недостатком является лишь незначительная перегрузка станка).

2.4. Моделирование работы объекта за заданное время

Пример 2.2. Пусть в условиях примера 2.1 требуется выполнить моделирование работы станков за восемь часов. В этом случае модель будет иметь следующий вид:

```

STANOK STORAGE      2
      GENERATE      7.5, 2.5
      QUEUE         OCH
      ENTER         STANOK
      DEPART        OCH
      ADVANCE       7, 1
      LEAVE         STANOK
      TERMINATE
      GENERATE      480
      TERMINATE     1

```

Модель запускается на выполнение командой **START 1**.

При запуске модели на выполнение командой **START 1** счетчик завершенных становится равным единице. Для удаления транзактов из модели (т.е.

для имитации окончания обработки деталей) используется оператор TERMINATE. Так как у него нет операндов, счетчик завершений при удалении транзактов из модели не изменяется. Оператор GENERATE 480 имитирует поступление транзактов с интервалом ровно 480 единиц времени; таким образом, этот оператор имитирует поступление первого транзакта через 480 единиц времени (в данном случае - 480 минут) после начала моделирования. Этот транзакт сразу же переходит на оператор TERMINATE 1, т.е. удаляется из модели; при этом из счетчика завершений вычитается единица. Так как при запуске модели счетчик завершений был задан равным единице, он становится равным нулю. Моделирование при этом завершается, и создается файл-отчет.

Важно понимать, что операторы GENERATE 480 и TERMINATE 1 *не представляют собой* какую-либо специальную конструкцию языка GPSS для указания времени моделирования. Рассмотрим, например, работу приведенной модели в случае, если она будет запущена на выполнение командой START 2 (а не START 1). По истечении 480 минут оператор GENERATE 480 имитирует поступление транзакта, который сразу же удаляется из модели (оператором TERMINATE 1). Счетчик завершений при этом уменьшается на единицу, т.е. становится равным *единице*. Через 480 минут оператор GENERATE 480 имитирует поступление еще одного транзакта; он также удаляется из модели, и счетчик завершений становится равным нулю. Таким образом, будет смоделирована работа станка в течение 16 часов. Те же результаты будут получены, если использовать в модели оператор GENERATE 960 (вместо GENERATE 480) и запустить ее на выполнение командой START 1.

2.5. Моделирование систем с заявками разных типов.

Управление движением транзактов. Параметры транзактов

Один из способов моделирования СМО с заявками разных типов - использование параметров транзактов (оператор ASSIGN) и управление движением транзактов в модели с помощью операторов TEST и TRANSFER.

Параметр транзакта представляет собой числовую величину, отражающую некоторую характеристику объекта, имитируемого транзактом. Изменение значения параметра выполняется следующим оператором:

ASSIGN A,B

где A - номер параметра (1, 2, 3 и т.д.) и (необязательно) знак + или -;

B - числовая величина, на которую изменяется параметр (число или выражение).

Если после операнда A (т.е. номера параметра) указан знак "плюс" или "минус", то величина, указанная в операнде B, *прибавляется* к указанному параметру (или *вычитается* из него). Если знак операции не указан, то значение операнда B *присваивается* указанному параметру; если при этом параметр *уже имел* какое-либо значение, то оно *теряется*.

Простейший оператор для управления перемещением транзактов в модели - оператор TRANSFER в следующей форме (режим безусловного перехода):

TRANSFER ,A

Здесь в качестве операнда A указывается метка одного из операторов модели. Транзакт, поступивший на такой оператор TRANSFER, передается на оператор с меткой A. Следует обратить внимание, что перед меткой в операторе TRANSFER необходимо указывать *запятую*.

Примечание. Оператор TRANSFER может использоваться и в других режимах. Они рассматриваются в подразд.2.6 и 9.1.

Для проверки условий в GPSS-моделях (и управления перемещением транзактов в зависимости от проверяемых условий) используется оператор TEST. Его структура следующая:

TEST операция A,B,C

где операция - обозначение операции сравнения: L - меньше, LE - меньше или равно, E - равно, NE - не равно, G - больше, GE - больше или равно;

A,B - некоторые величины, для которых выполняется заданная операция сравнения (величина A сравнивается с величиной B);

C - метка оператора, на который передается транзакт, если результат операции сравнения величин A и B - "ложь". Если результат сравнения - "истина", то транзакт передается на оператор, следующий за оператором TEST.

Если операнд C не указан, то в случае невыполнения условия, заданного операцией сравнения, транзакт *задерживается* в операторе TEST до тех пор, пока условие не будет выполняться.

Между словом TEST и обозначением операции должен находиться *в точности один* пробел.

Пример 2.3. На двух одинаковых станках обрабатываются детали двух типов (деталь любого типа может обрабатываться на любом из станков). Детали типа 1 поступают с интервалами от 5 до 10 мин, типа 2 - с интервалами от 8 до 12 мин. Обработка детали типа 1 занимает от 6 до 8 мин, детали типа 2 - ровно 6 мин. Требуется разработать GPSS-модель для анализа работы станков в течение восьми часов.

STANOK	STORAGE	2
	GENERATE	7.5,2.5
	ASSIGN	1,1
	TRANSFER	,OBR
	GENERATE	10,2
	ASSIGN	1,2
OBR	QUEUE	OCH
	ENTER	STANOK
	DEPART	OCH


```

TEST E      P1,1,TIP2
ADVANCE     7,1
TRANSFER    ,KONOB
TIP2        ADVANCE 6
KONOB       LEAVE    STANOK
TERMINATE
GENERATE     480
TERMINATE    1

```

Здесь оператор GENERATE 7.5,2.5 имитирует поступление деталей типа 1. Оператор ASSIGN 1,1 присваивает первому параметру транзакта значение 1. Оператор TRANSFER ,OBR передает транзакт на метку OBR (т.е. на оператор QUEUE); таким образом имитируется поступление деталей типа 1 в накопитель перед станками.

Оператор GENERATE 10,2 имитирует поступление деталей типа 2. Оператор ASSIGN 1,2 присваивает первому параметру транзакта значение 2. Затем эти транзакты поступают на оператор QUEUE, т.е. имитируется поступление деталей типа 2 в накопитель перед станками.

Таким образом, в результате использования операторов ASSIGN 1,1 и ASSIGN 1,2 для всех транзактов, имитирующих детали типа 1, первому параметру присваивается значение 1, а для всех транзактов, имитирующих детали типа 2 - значение 2.

Примечание. В данном примере параметр транзакта используется для того, чтобы “различать” транзакты, имитирующие детали типа 1 и типа 2. Такое применение параметров транзактов – не единственно возможное. Многие другие примеры будут приведены ниже.

В операторе TEST E P1,1,TIP2 выполняется сравнение значения параметра P1 (т.е. *первого параметра*) с числом 1. Если P1=1, то транзакт пропускается в следующий оператор, т.е. на оператор ADVANCE 7,1. Как показано выше, условие P1=1 выполняется для транзактов, имитирующих детали типа 1. Оператор ADVANCE 7,1 имитирует обработку деталей типа 1 в течение 7 ± 1 мин (т.е. от 6 до 8 мин). Оператор TRANSFER ,KONOB передает транзакты на метку KONOB, т.е. на оператор LEAVE, имитирующий окончание обработки детали на станке.

Если условие P1=1 *не выполняется*, то транзакт направляется на метку TIP2, т.е. на оператор ADVANCE 6. Таким образом имитируется обработка деталей типа 2, занимающая ровно 6 минут. После этого имитируется выход детали типа 2 со станка (оператор LEAVE STANOK).

Следует еще раз обратить внимание, что условие, указываемое в операторе TEST – это условие пропускания транзакта на оператор, *следующий* за оператором TEST. Если условие *не выполняется*, то транзакт переходит на оператор, метка которого указана в операнде C, а если этот операнд не указан, то транзакт задерживается и пропускается в следующий оператор только тогда, когда заданное условие будет выполняться.

2.6. Моделирование систем с заданным соотношением заявок двух типов

Во многих практических задачах требуется имитировать системы, в которых некоторое событие происходит в $X\%$ случаев и не происходит в $100-X\%$ случаев. Для имитации таких явлений используется оператор TRANSFER в следующей форме (статистический режим):

TRANSFER A,B,C

где A - трехзначное число (NNN);

B - метка оператора, на который передаются $100-NN,N$ процентов транзактов;

C - метка оператора, на который передаются NN,N процентов транзактов.

Например, оператор TRANSFER 300,МЕТ1,МЕТ5 означает, что в 70% случаев транзакт должен передаваться на оператор с меткой МЕТ1, а в 30% случаев - на оператор с меткой МЕТ5.

Если операнд B не указан, то $100-NN,N\%$ транзактов передаются на оператор, следующий за оператором TRANSFER, а $NN,N\%$ - на оператор с меткой C.

Пример 2.4. На двух однотипных станках обрабатываются детали двух типов (деталь любого типа может обрабатываться на любом из станков). Детали типа 1 поступают с интервалами от 5 до 10 мин, типа 2 - с интервалами от 8 до 12 мин, как и в примере 4.

Обработка детали на станке состоит из двух операций. Первая операция для деталей типа 1 занимает от 6 до 8 мин, для деталей типа 2 - ровно 6 мин. Вторая операция занимает от 2 до 4 мин, независимо от типа детали. При первой операции в 5% случаев допускается брак; в этих случаях вторая операция не требуется.

Все детали, кроме бракованных, поступают на упаковочную машину. Упаковка детали любого типа занимает ровно 3 мин.

Требуется разработать GPSS-модель для анализа работы участка (т.е. станков и упаковочной машины) в течение восьми часов.

Ниже приводится текст GPSS-модели.

```
STANOK STORAGE      2
      GENERATE      7.5,2.5
      ASSIGN        1,1
      TRANSFER      ,OBR
      GENERATE      10,2
      ASSIGN        1,2
OBR   QUEUE         OCH
      ENTER         STANOK
      DEPART        OCH
      TEST E        P1,1,TIP2
      ADVANCE       7,1
      TRANSFER      ,OPER2
TIP2  ADVANCE       6
```

```

OPER2  TRANSFER  050,,BRAK
        ADVANCE  3,1
        ASSIGN   2,0
        TRANSFER ,KONOB
BRAK    ASSIGN   2,1
KONOB  LEAVE     STANOK
        TEST E   P2,0,KON
        QUEUE    OCH2
        SEIZE    UPAK
        DEPART   OCH2
        ADVANCE  3
        RELEASE  UPAK
KON     TERMINATE
        GENERATE 480
        TERMINATE 1

```

Начало модели аналогично показанному в примере 2.3. Следующая группа операторов

```

        TEST E   P1,1,TIP2
        ADVANCE  7,1
        TRANSFER ,OPER2
TIP2    ADVANCE  6

```

имитирует выполнение первой операции для деталей типа 1 и 2.

Оператор TRANSFER 050,,BRAK имитирует разделение деталей на годные и бракованные. Транзакты, имитирующие годные детали (95%), поступают на следующий оператор ADVANCE 3,1, т.е. на имитацию второй операции. В операторе ASSIGN 2,0 второму параметру этих транзактов присваивается значение 0. Затем с помощью оператора TRANSFER ,KONOB они передаются на оператор LEAVE STANOK, т.е. имитируется окончание обработки детали.

Пять процентов транзактов из оператора TRANSFER 050,,BRAK передаются на метку BRAK. Эти транзакты имитируют бракованные детали, поэтому они *не передаются* на оператор ADVANCE 3,1, имитирующий вторую операцию. В операторе ASSIGN 2,1 второму параметру этих транзактов присваивается значение 1. Таким образом, у всех транзактов, имитирующих годные детали, *второй* параметр равен нулю, а у транзактов, имитирующих бракованные детали - единице.

В операторе TEST E P2,0,KON проверяется, чему равен второй параметр транзакта. Если P2=0 (т.е. поступил транзакт, имитирующий годную деталь), то он пропускается на следующий оператор. Группа операторов QUEUE OCH2 ... RELEASE UPAK имитирует работу упаковочной машины, т.е. упаковку детали.

Если P2≠0 (т.е. поступил транзакт, имитирующий бракованную деталь), то из оператора TEST E P2,0,KON этот транзакт передается на метку KON. Таким образом, для него имитация упаковки не выполняется.

2.7. Простейшие вычисления в GPSS-модели. Сохраняемые величины

Простейший способ вычислений в GPSS-моделях - использование *сохраняемых величин* (ячеек). Для этого применяется следующий оператор:

```
SAVEVALUE      A,B
```

где A - имя сохраняемой величины и (необязательно) знак + или -;

B - величина, на которую изменяется параметр (число или некоторое выражение).

Если после операнда A (т.е. имени сохраняемой величины) указан знак "плюс" или "минус", то число, указанное в операнде B, прибавляется к указанной сохраняемой величине (или вычитается из нее). Если знак операции не указан, то значение операнда B присваивается указанной сохраняемой величине (если сохраняемая величина уже имела какое-либо значение, то оно теряется).

Пример 2.5. Пусть в модели, разработанной для примера 2.4, требуется предусмотреть подсчет количества выпущенных годных деталей каждого типа, а также общую выручку от выпуска всех деталей. Годные детали типа 1 продаются по 8 ден.ед., типа 2 - по 10 ден.ед., бракованные - по 2 ден.ед. (как металлолом, независимо от типа).

```
STANOK STORAGE      2
... CM. ПРИМЕР 2.4 ...
KONOBK LEAVE         STANOK
      TEST E         P2,0,METALL
      QUEUE          OCH2
      SEIZE          UPAK
      DEPART         OCH2
      ADVANCE        3
      RELEASE        UPAK
      TEST E         P1,1,IZD2
      SAVEVALUE      VIPUSK1+,1
      SAVEVALUE      VIRUCHKA+,8
      TRANSFER       ,KON
IZD2   SAVEVALUE      VIPUSK2+,1
      SAVEVALUE      VIRUCHKA+,10
      TRANSFER       ,KON
METALL SAVEVALUE      BRAKOVANO+,1
      SAVEVALUE      VIRUCHKA+,2
KON    TERMINATE
      GENERATE       480
      TERMINATE      1
```

Начало модели ничем не отличается от примера 2.4. В операторе TEST E P1,1,IZD2 определяется тип выпущенной годной детали. Если первый параметр (P1) равен единице, то выпущена деталь типа 1, в противном случае - деталь типа 2. Если P1=1, то транзакт пропускается в следующий оператор. Оператор SAVEVALUE VIPUSK1+,1 предназначен для подсчета количества выпу-

щенных деталей типа 1, а оператор SAVEVALUE VIRUCHKA+,8 - для подсчета выручки от их выпуска. Операторы SAVEVALUE VIPUSK2+,1 и SAVEVALUE VIRUCHKA+,10 выполняют аналогичные подсчеты для деталей типа 2, а операторы SAVEVALUE BRAKOVANO+,1 и SAVEVALUE VIRUCHKA+,2 - для бракованных деталей.

Ниже приведены основные результаты моделирования:

FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
UPAK	91	0.569	3.000		1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
OCH	10	10	111	5	4.412	19.079	19.979	0
OCH2	1	0	91	65	0.078	0.412	1.442	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STANOK	2	0	0	2	101	1	1.947	0.973	0	10

SAVEVALUE	RETRY	VALUE
VIPUSK1	0	52.000
VIRUCHKA	0	822.000
VIPUSK2	0	39.000
BRAKOVANO	0	8.000

Из этих результатов видно, что станки (имитировавшие как многоканальное устройство с именем STANOK) явно перегружены: их коэффициент загрузки составляет 0,973. На это же указывает длина очереди перед станками, достигавшая 10 (колонка MAX в разделе QUEUE), причем в момент окончания моделирования в очереди перед станками находилось 10 деталей (колонка CONT). Среднее количество деталей в очереди к станкам (т.е. среднее количество деталей в накопителе) составило 4,412 детали, а среднее время ожидания обработки на станках - 19,079 мин. Только 5 деталей поступили на обработку на станки сразу (без ожидания в накопителе).

Упаковочная машина загружена недостаточно (коэффициент загрузки - 0,569). Очередь деталей перед упаковочной машиной практически не образовывалась: для 65 деталей (из 91), обработанных на упаковочной машине, ожидание упаковки вообще не потребовалось: при поступлении этих деталей упаковочная машина была свободна. Среднее количество деталей в накопителе перед упаковочной машиной составило 0,078 детали, а среднее время ожидания в накопителе - 0,412 мин.

Всего за 8 часов было выпущено 52 детали типа 1 и 39 деталей типа 2. Было забраковано 8 деталей. Выручка составила 822 ден.ед.

Для годных деталей найдем среднее время технологического цикла, т.е. среднее время обработки детали на участке (от поступления на участок до окончания упаковки). Это время складывается из времени пребывания в накопителе перед станками (19,079 мин), времени обработки на одном из станков (не указано в результатах моделирования), времени пребывания в накопителе перед упаковочной машиной (0,412 мин) и времени упаковки (3 мин).

Время обработки на станке не указано в модели. Его необходимо вычислить (с учетом того, что для деталей типа 1 и типа 2 оно было разным).

Найдем соотношение деталей типа 1 и 2. Было выпущено 52 детали типа 1 и 39 деталей типа 2 (всего - 91 деталь). Найдем доли деталей каждого типа: $52/91 = 0,57$, $39/91 = 0,43$.

Среднее время первой операции для деталей типа 1 составило 7 мин, для деталей второго типа - 6 мин. Среднее время выполнения второй операции на станке составляло 3 мин для деталей всех типов. Таким образом, среднее время обработки детали на станке можно найти следующим образом: $0,57 \cdot 7 + 0,43 \cdot 6 + 3 = 9,57$ мин.

Среднее время технологического цикла: $19,079 + 9,57 + 0,412 + 3 = 32,061$ мин, т.е. примерно 32 мин.

Примечание. В данном примере не учитывается, что при выполнении первой операции часть деталей бракуется (т.е. для них не требуется вторая операция, а также упаковка), так как требуется найти среднее время технологического цикла только для *годных* деталей.

По результатам моделирования можно предложить следующие рекомендации по улучшению работы участка:

- использовать еще один станок для обработки деталей;
- использовать упаковочную машину для выполнения каких-либо дополнительных работ (например, для упаковки изделий, поступающих с других участков).

2.8. Циклы

Для организации цикла (т.е. повторяющегося несколько раз прохождения транзакта через группу операторов модели) используется оператор LOOP. Оператор LOOP всегда указывается *последним* в группе операторов, образующих цикл. Структура оператора LOOP следующая:

LOOP A,B

где A - номер параметра транзакта, в котором указано количество повторений цикла;

B - метка оператора, с которого *начинается* цикл.

Оператор LOOP работает следующим образом. При входе транзакта в этот оператор значение параметра, указанного в операнде A, уменьшается на единицу. Если значение параметра (после вычитания единицы) превышает ноль, то транзакт переходит на метку, указанную в операнде B, т.е. цикл повторяется. Когда параметр оказывается равным нулю, цикл завершается: транзакт переходит на оператор, следующий за LOOP.

Следует обратить внимание, что по окончании цикла параметр, использованный в качестве счетчика цикла, оказывается равным нулю.

Пример 2.6. На участке сборки некоторых электронных изделий работают 10 сборщиков. Изготовление изделия заключается в установке восьми деталей

на печатную плату. Установка одной детали занимает от 1 до 3 мин. После того как на плате установлены все детали, выполняется измерение электрических характеристик изделия, занимающее от 3 до 5 мин. Комплекты для сборки, включающие печатную плату и восемь деталей, поступают на участок с интервалом от 2 до 4 мин.

Требуется разработать GPSS-модель для анализа работы участка сборки в течение восьми часов. Предусмотреть подсчет собранных изделий.

В данном случае изготовление каждого изделия включает восемь операций установки деталей на плату и одну операцию измерения. Для имитации установки деталей на плату удобно использовать цикл.

SBORSHCHIK	STORAGE	10
	GENERATE	3, 1
	QUEUE	OCH_SBOR
	ENTER	SBORSHCHIK
	DEPART	OCH_SBOR
	ASSIGN	1, 8
SBORKA	ADVANCE	2, 1
	LOOP	1, SBORKA
	ADVANCE	4, 1
	LEAVE	SBORSHCHIK
	SAVEVALUE	SOBRANO+, 1
	TERMINATE	
	GENERATE	480
	TERMINATE	1

Здесь в операторе ASSIGN 1,8 первому параметру транзакта присваивается значение 8. Первый параметр транзакта будет затем использован в качестве счетчика циклов.

Операторы

SBORKA	ADVANCE	2, 1
	LOOP	1, SBORKA

представляют собой цикл, имитирующий установку деталей на плату. Цикл повторяется восемь раз (столько, сколько указано в первом параметре). Следует еще раз обратить внимание, что по окончании цикла (т.е. после оператора LOOP) значение первого параметра равно нулю.

Оператор ADVANCE 4,1 имитирует измерение электрических характеристик изделия.

Остальные операторы, использованные в данной модели, рассмотрены в предыдущих примерах.

3. ИМИТАЦИЯ СЛУЧАЙНЫХ СОБЫТИЙ И ВЕЛИЧИН В GPSS-МОДЕЛЯХ

3.1. Имитация случайных величин в GPSS World

В примерах, приводившихся ранее, рассматривалось моделирование систем, в которых все величины были распределены по равномерному закону или представляли собой детерминированные величины. Например, если время обработки изделия на станке может составлять от 5 до 10 мин, то его можно считать равномерной случайной величиной; если оно всегда составляет, например, ровно 5 мин, то оно представляет собой детерминированную величину.

В системе моделирования GPSS World имеются стандартные средства для имитации других случайных величин, необходимых для имитации реальных явлений. Функции GPSS World для имитации случайных величин, наиболее часто применяемых в практических задачах, приведены в прил.4. Полный список функций для имитации случайных величин имеется в документации по системе моделирования GPSS World и в системе подсказок.

Пример 3.1. В организации, выполняющей заказы на переводы с иностранных языков, работают 7 переводчиков. В среднем поступает 5 заказов в день; поток заказов можно считать пуассоновским. Размер перевода – случайная величина, распределенная по экспоненциальному закону; средний размер перевода - 8 листов. Переводчик переводит в среднем 6 листов в день. Прибыль организации от каждого переведенного листа составляет 7,5 ден.ед. Требуется разработать GPSS-модель для анализа работы организации в течение года (300 рабочих дней). Определить прибыль от переводов, выполненных за это время.

Напомним, что в пуассоновском потоке заявок интервалы времени между моментами поступления заявок представляют собой случайные величины, распределенные по экспоненциальному закону (экспоненциальные случайные величины). Такая случайная величина может принимать как очень малые, так и очень большие значения (более строгое определение пуассоновского потока и экспоненциальной случайной величины приводится в курсе теории вероятностей). В данном примере это означает, что интервалы времени между моментами поступления заказов могут быть как очень короткими, так и очень длительными. Размер заказа - также экспоненциальная случайная величина. Это означает, что размеры заказов могут быть как очень малыми (например, если заказчику требуется перевести на иностранный язык чек на покупку какого-либо товара), так и очень большими (например, если требуется перевод книги).

```
PEREV STORAGE      7
      GENERATE      (EXPONENTIAL(1,0,0.2))
      ASSIGN        1,(EXPONENTIAL(2,0,8))
      QUEUE         OCH
      ENTER          PEREV
      DEPART        OCH
```


ADVANCE	(P1/6)
LEAVE	PEREV
SAVEVALUE	PRIB+, (P1#7.5)
TERMINATE	
GENERATE	300
TERMINATE	1

Для имитации случайных величин, распределенных по экспоненциальному закону, здесь используется функция `EXPONENTIAL`.

Первый аргумент этой функции - номер датчика (генератора) случайных чисел, т.е. номер встроенной процедуры языка GPSS, имитирующей случайные равномерно распределенные числа (СРРЧ). Этот номер может представлять собой любое число от 1 до 999. Случайные числа, распределенные по заданному закону (в данном случае - по экспоненциальному), имитируются с использованием СРРЧ на основе метода Монте-Карло. Если в модели имитируется несколько случайных величин, то рекомендуется использовать для их имитации генераторы СРРЧ с разными номерами, чтобы обеспечить независимость имитируемых величин.

Второй аргумент функции `EXPONENTIAL` - смещение имитируемых случайных величин. Этот аргумент в большинстве случаев равен нулю. Третий аргумент - среднее значение имитируемой величины.

В данном примере оператор `GENERATE (EXPONENTIAL(1,0,0.2))` имитирует поток заявок (заказов на переводы) с интервалами, распределенными по экспоненциальному закону. Средний интервал между заявками составляет 0,2 дня, так как поступает в среднем 5 заказов в день.

В операторе `ASSIGN 1,(EXPONENTIAL(2,0,8))` первому параметру транзакта, имитирующего заказ, присваивается значение, представляющее собой экспоненциальную случайную величину со средним значением, равным 8. Эта величина имитирует размер заказа.

Примечание. Важно понимать, что величины, присваиваемые первому параметру транзакта в операторе `ASSIGN 1,(EXPONENTIAL(2,0,8))`, являются *разными* для каждого из транзактов. *Среднее* значение этих величин равно 8.

Операторы `QUEUE OCH ... LEAVE PEREV` имитируют работу переводчиков. В операторе `ADVANCE (P1/6)` имитируется время выполнения заказа переводчиком (в днях). Это время определяется как отношение значения первого параметра (т.е. размера перевода в листах) к количеству листов, переводимых переводчиком за день.

В операторе `SAVEVALUE PRIB+, (P1#7.5)` выполняется подсчет прибыли. Прибыль от выполнения перевода вычисляется как произведение размера перевода (первого параметра) на прибыль от одного листа (7,5). Эта величина прибавляется к сохраняемой величине `PRIB`. Знак `#` в языке GPSS обозначает операцию *умножения*.

Пример 3.2. Ремонтная служба предприятия выполняет профилактический осмотр и ремонт некоторых приборов. Каждый прибор состоит из 20 одностип-

ных узлов. За время между ремонтами примерно 30% узлов выходит из строя. В ремонтной службе работают три ремонтника. Поток приборов, поступающих для осмотра и ремонта, представляет собой поток Эрланга второго порядка; средний интервал между моментами поступления приборов - 15 минут. Ремонтник выполняет осмотр прибора и замену отказавших узлов. Осмотр прибора занимает от 5 до 15 мин. Время, необходимое для замены одного узла - гауссовская (нормальная) случайная величина со средним значением 3 мин и стандартным отклонением 0,5 мин. Требуется разработать GPSS-модель для анализа работы ремонтной службы в течение 100 часов. Предусмотреть подсчет количества замененных узлов.

В данном примере количество неисправных (требующих замены) узлов в каждом приборе представляет собой случайную величину, которая может принимать (с различными вероятностями) значения от 0 до 20. Каждый из 20 узлов, входящих в прибор, может с вероятностью 0,3 (30%) оказаться неисправным, и с вероятностью $1-0,3=0,7$ - исправным. Таким образом, количество неисправных узлов - это случайная величина, представляющая собой количество событий в N испытаниях, где в каждом испытании событие может произойти с вероятностью P или не произойти с вероятностью $1-P$ (в данном примере $N=20$, $P=0,3$, под событием понимается неисправность узла). Такая случайная величина называется *биномиальной* (распределенной по биномиальному закону); значения N и P – параметры биномиального распределения.

```

REMONТ  STORAGE      3
          generate    (GAMMA(1,0,7.5,2))
          ASSIGN      1,(BINOMIAL(2,20,0.3))
          ASSIGN      2,P1
          QUEUE        1
          ENTER        REMONТ
          DEPART       1
          ADVANCE      (UNIFORM(3,5,15))
          TEST G       P1,0,ISPRAV
ZAMENA   ADVANCE      (NORMAL(4,3,0.5))
          LOOP         1,ZAMENA
ISPRAV   LEAVE        REMONТ
          SAVEVALUE    ZAMENENO+,P2
          TERMINATE
          GENERATE      6000
          TERMINATE    1

```

Здесь оператор GENERATE имитирует поток приборов, поступающих в ремонтную службу. Для имитации интервалов времени между приборами, распределенных по закону Эрланга второго порядка, используется функция GAMMA (см. прил.4, примечание к описанию функции GAMMA).

В операторе ASSIGN 1,(BINOMIAL(2,20,0.3)) с помощью функции BINOMIAL имитируется количество неисправных узлов, представляющее собой биномиальную случайную величину. Эта величина присваивается первому параметру транзакта, имитирующего прибор. В операторе ASSIGN 2,P1 значе-

ние первого параметра (P1) присваивается второму параметру. Это необходимо из-за того, что параметр P1 будет использован в качестве счетчика циклов (в операторе LOOP), и по окончании цикла он окажется равным нулю.

Операторы QUEUE 1 ... LEAVE REMONT имитируют работу ремонтников. В операторе ADVANCE (UNIFORM(3,5,15)) имитируется осмотр прибора, занимающий от 5 до 15 мин. В операторе TEST G P1,0,ISPRAV выполняется проверка: имеются ли неисправные узлы. Если такие узлы есть ($P1 > 0$), то имитируется их замена. Для этого используется оператор цикла LOOP. Для имитации времени замены каждого из узлов, представляющего собой гауссовскую случайную величину, используется функция NORMAL.

Описания функций BINOMIAL, UNIFORM и NORMAL см. в прил.4.

В операторе SAVEVALUE ZAMENENO+,P2 выполняется подсчет количества замененных узлов: сохраняемая величина ZAMENENO увеличивается на значение *второго параметра* транзакта, которому присвоено количество заменяемых узлов.

Примечание. Вместо оператора ADVANCE (UNIFORM(3,5,15)) можно использовать оператор ADVANCE 10,5.

3.2. Имитация случайных событий в GPSS World. Функции. Моделирование систем с заявками нескольких типов

Функции в GPSS предназначены для получения некоторой величины (значения функции Y), значение которой зависит от другой величины (аргумента функции X). Функции в GPSS задаются перечислением пар "аргумент, значение". Функции объявляются в разделе команд (т.е. в начале модели, до первого оператора GENERATE) следующим образом:

```
ИМЯ_ФУНКЦИИ    FUNCTION    A,B  
X1,Y1/X2,Y2/.../XN,YN
```

где A - аргумент функции;

B - обозначение, имеющее вид DN (для дискретных функций) или CN (для непрерывных функций), где N - количество пар "аргумент,значение";

X₁,Y₁/X₂,Y₂/.../X_N,Y_N - пары "аргумент,значение" (обычно - числа).

Значения аргумента функции должны перечисляться по возрастанию, т.е. должно выполняться условие $X_1 < X_2 < \dots < X_N$.

Дискретная функция принимает *только* значения Y₁, Y₂,...,Y_N. Значение дискретной функции определяется в соответствии со значением аргумента X следующим образом: если $X_{i-1} \leq X \leq X_i$, то функция принимает значение Y_i. Например, если $X \leq X_1$, то функция принимает значение Y₁; если $X_1 < X \leq X_2$, то функция принимает значение Y₂, и т.д.

Непрерывная функция принимает *любые* значения от Y₁ до Y_N. Если аргумент функции X принимает значение на отрезке $[X_{i-1}, X_i]$, то функция прини-

мает значение на отрезке $[Y_{i-1}, Y_i]$. На каждом из таких отрезков зависимость величины Y от X является линейной. Например, если аргумент X принимает значение, равное середине отрезка $[X_{i-1}, X_i]$, то функция принимает значение, равное середине отрезка $[Y_{i-1}, Y_i]$.

Важное (но не единственное) применение дискретных функций в GPSS - моделирование систем с заявками нескольких (трех и более) типов.

Примечание. Для моделирования заявок двух типов в GPSS обычно достаточно использовать оператор TRANSFER, рассмотренный в подразд.2.5 и 2.6.

Пример 3.3. Требуется разработать модель для анализа работы организации, выполняющей переводы с иностранных языков (см. пример 3.1). Известно, то из всех заказов на переводы примерно 40% составляют заказы на технические переводы, 30% - экономические, 20% - юридические, 10% - прочие. Прибыль организации от одного листа технического перевода - 6 ден.ед., экономического - 7,5, юридического - 8,5, прочих - 10 ден.ед.

По результатам моделирования требуется определить прибыль, а также количество выполненных переводов каждого типа.

```
PEREV STORAGE 7
TIP FUNCTION RN1,D4
0.4,1/0.7,2/0.9,3/1,4
PRIBYL FUNCTION P2,D4
1,6/2,7.5/3,8.5/4,10
GENERATE (EXPONENTIAL(1,0,0.2))
ASSIGN 1,(EXPONENTIAL(2,0,8))
ASSIGN 2, FN$TIP
QUEUE OCH
ENTER PEREV
DEPART OCH
ADVANCE (P1/6)
LEAVE PEREV
SAVEVALUE PRIB+, (P1#FN$PRIB)
SAVEVALUE P2+, 1
TERMINATE
GENERATE 300
TERMINATE 1
```

Здесь для имитации заявок четырех типов (технические, экономические, юридические и прочие переводы) используется дискретная функция TIP. Она может принимать одно из четырех значений: 1, 2, 3 или 4. Аргументом этой функции является CРРЧ, разыгрываемое генератором случайных чисел RN1.

Функция TIP используется для имитации типа заявки (перевода) в операторе ASSIGN 2, FN\$TIP. Значение функции TIP при выполнении этого оператора определяется следующим образом. Разыгрывается CРРЧ RN1, т.е. некоторое число из диапазона от нуля до единицы. Если $RN1 \leq 0,4$, то функция принимает значение 1 ($TIP=1$); если $0,4 < RN1 \leq 0,7$, то $TIP=2$; если $0,7 < RN1 \leq 0,9$, то $TIP=3$; если $0,9 < RN1 \leq 1$, то $TIP=4$. В операторе ASSIGN 2, FN\$TIP значение функции присваивается второму параметру транзакта. Таким образом, пример-

но в 40% случаев этот параметр будет иметь значение 1, в 30% случаев - значение 2, в 20% - значение 3, в 10% - значение 4. Транзакты, у которых P2=1, используются для имитации заявок на технические переводы. Транзакты, у которых второй параметр равен 2, 3 или 4, используются для имитации экономических, юридических и прочих переводов соответственно.

Примечание. Используемый в этом примере алгоритм имитации дискретной случайной величины рассмотрен в первой части данного пособия.

Функция PRIBYL используется для имитации прибыли от одного листа перевода. Эта прибыль зависит от типа перевода. Аргументом функции PRIBYL является значение второго параметра транзакта (P2), обозначающего тип перевода.

В операторе SAVEVALUE PRIB+, (P1#FN\$PRIBYL) значение параметра P1 (т.е. размер перевода) умножается на значение функции PRIBYL, определяемое в зависимости от значения P2. Если P2=1 (т.е. имитируется технический перевод), то функция PRIBYL принимает значение 6; если P2=2 (экономический перевод), то PRIBYL=7,5; если P2=3 (юридический перевод), то PRIBYL=8,5; если P2=4 (другой тип перевода), то PRIBYL=10. Таким образом, величина P1#FN\$PRIBYL - это прибыль от выполнения перевода. Она прибавляется к сумме значений прибыли (сохраняемая величина PRIB).

В операторе SAVEVALUE P2+,1 на единицу увеличивается сохраняемая величина, номер которой указан в параметре P2. Например, если через этот оператор проходит транзакт, у которого P2=3 (т.е. транзакт, имитирующий юридический перевод), то сохраняемая величина с номером 3 увеличивается на единицу.

Приведем фрагмент результатов моделирования:

SAVEVALUE	VALUE
1	589.000
2	460.000
3	323.000
4	133.000
PRIB	86783.333

Это означает, что за время моделирования (300 дней) было выполнено 589 технических переводов, 460 экономических, 323 юридических и 133 прочих. Прибыль организации составила 86 783,33 ден.ед.

4. СТАНДАРТНЫЕ ЧИСЛОВЫЕ АТТРИБУТЫ

4.1. Понятие стандартных числовых атрибутов

Стандартные числовые атрибуты (СЧА) - это числовые характеристики различных объектов GPSS-модели (очереди, устройств и т.д., а также модели в целом), автоматически вычисляемые в процессе моделирования. На СЧА можно ссылаться в операторах GPSS-модели.

Чаще всего используются следующие СЧА:

F - признак занятости одноканального устройства: 0, если устройство свободно; 1, если устройство занято, т.е. в нем есть транзакт;

S - текущее количество занятых каналов в многоканальном устройстве;

R - текущее количество свободных каналов в многоканальном устройстве;

SF - признак заполненного состояния многоканального устройства: 1, если все каналы в устройстве заняты; 0, если в устройстве есть хотя бы один свободный канал;

SE - признак свободного состояния многоканального устройства: 1, если все каналы в устройстве свободны; 0, если в устройстве есть хотя бы один транзакт;

Q - текущая длина очереди.

Некоторые другие СЧА будут рассмотрены ниже. Полный список СЧА приводится в прил.5.

Ссылки на СЧА в тексте GPSS-модели записываются следующим образом: СЧА\$имя_объекта. Если объект модели (например, очередь) имеет не символьное, а числовое имя (т.е. номер), то знак \$ не указывается.

Приведем несколько простых примеров использования СЧА.

Оператор TEST E F\$STANOK,0,VINOD действует следующим образом: если одноканальное устройство с именем STANOK свободно, то транзакт пропускается дальше (на оператор, следующий за оператором TEST); если устройство занято, то транзакт направляется на метку VINOD.

Оператор TEST L Q1,5,OTKAZ действует следующим образом: если длина очереди с номером (числовым именем) 1 составляет менее 5, то транзакт переходит в следующий оператор; если длина очереди больше или равна 5, то транзакт переходит на оператор с меткой OTKAZ.

4.2. Примеры применения СЧА: моделирование СМО с ограничениями на очередь

Пример 4.1 (модель СМО с ограничением на количество заявок в очереди). Разработать модель для анализа работы организации, описанной в примере 3.3, при следующем дополнительном условии: организация не принимает заказы на переводы, если уже имеется 10 заказов, ожидающих выполнения. Определить вероятность отказа (в процентах).

```

PEREV    STORAGE      7
TIP      FUNCTION     RN1,D4
0.4,1/0.7,2/0.9,3/1,4
PRIBYL   FUNCTION     P2,D4
1,6/2,7.5/3,8.5/4,10
          GENERATE     (EXPONENTIAL(1,0,0.2))
          SAVEVALUE    VSEGO+,1
          TEST L       Q$OCH,10,OTKAZ
          ASSIGN       1,(EXPONENTIAL(2,0,8))
          ASSIGN       2, FN$TIP
          QUEUE        OCH
          ENTER        PEREV
          DEPART       OCH
          ADVANCE      (P1/6)
          LEAVE        PEREV
          SAVEVALUE    PRIB+, (P1#FN$PRIBYL)
          SAVEVALUE    P2+,1
          TRANSFER     ,KON
OTKAZ    SAVEVALUE    OTK+,1
KON      TERMINATE
          GENERATE     300
          SAVEVALUE    PROCOTK, ((X$OTK/X$VSEGO)#100)
          TERMINATE    1

```

В операторе TEST L Q\$OCH,10,OTKAZ текущее значение длины очереди OCH (СЧА Q) сравнивается с числом 10. Если длина очереди меньше 10, то транзакт переходит в следующий оператор; таким образом имитируется прием заказа в случае, если количество заказов, ожидающих выполнения, меньше 10. Если длина очереди равна 10 (т.е. *не меньше* 10), то транзакт переходит на метку OTKAZ.

По окончании моделирования значение сохраняемой величины ОТК равно количеству отказов, а значение сохраняемой величины VSEGO - общему количеству заказчиков, обращавшихся в организацию (как обслуженных, так и получивших отказ). Оператор SAVEVALUE PROCOTK, ((X\$OTK/X\$VSEGO)#100) используется для вычисления процента отказов. Этот оператор выполняется после того, как в операторе GENERATE 300 имитируется появление транзакта, т.е. непосредственно перед окончанием моделирования. Обозначение X\$ИМЯ - это ссылка на значение сохраняемой величины.

Пример 4.2 (модель СМО без очереди). В справочной службе имеется 10 телефонов. Поток запросов (телефонных звонков) в службу можно считать пуассоновским; в среднем поступает три звонка в минуту. Длительность разговора - случайная величина, распределенная по гауссовскому (нормальному) закону со средним значением 2,5 мин и стандартным отклонением 15 с. Если в момент звонка все телефоны оказываются занятыми, то клиент обычно звонит еще раз в среднем через 5 мин (интервал до второго звонка - случайная величина, распределенная по экспоненциальному закону). Если все телефоны снова оказываются занятыми, то больше клиент обычно не звонит.

Требуется разработать GPSS-модель для анализа работы справочной службы в течение одного часа. Вычислить вероятность отказа (т.е. вероятность того, что клиент не дозвонится до справочной службы).

```
TELEFON STORAGE      10
                GENERATE (EXPONENTIAL(1,0,1/3))
                ASSIGN   1,0
                SAVEVALUE VSEGO+,1
ZVONOK  TEST NE      SF$TELEFON,1,ZAN
                ENTER    TELEFON
                ADVANCE  (NORMAL(2,2.5,0.25))
                LEAVE    TELEFON
                TRANSFER ,KON
ZAN      TEST E       P1,0,OTKAZ
                ASSIGN   1,1
                ADVANCE  (EXPONENTIAL(3,0,5))
                TRANSFER ,ZVONOK
OTKAZ    SAVEVALUE    NEDOZVON+,1
KON      TERMINATE
                GENERATE 60
                SAVEVALUE PROC_OTK,(X$NEDOZVON/X$VSEGO)
                TERMINATE 1
```

В операторе TEST NE SF\$TELEFON,1,ZAN проверяется значение стандартного числового атрибута SF для многоканального устройства TELEFON. Как указано выше, СЧА SF равен нулю, если в многоканальном устройстве имеется хотя бы один свободный канал; СЧА SF равен единице, если все каналы заняты. Если для устройства TELEFON выполняется условие $SF \neq 1$, это значит, что хотя бы один телефон свободен. В этом случае транзакт пропускается в следующий оператор для имитации разговора. В противном случае происходит переход на метку ZAN.

Группа операторов

```
ZAN TEST E      P1,0,OTKAZ
      ASSIGN     1,1
      ADVANCE    (EXPONENTIAL(3,0,5))
      TRANSFER   ,ZVONOK
```

имитирует повторный звонок в среднем через 5 минут после первого, а также невозможность еще одного (третьего) звонка.

4.3. Примеры применения СЧА: выбор устройства для обслуживания заявки

Во многих случаях в GPSS-модели оказывается необходимым выбрать одно из возможных направлений движения транзакта. Например, при имитации обслуживания заявок может потребоваться выбрать свободное устройство, самую короткую очередь и т.д. Для этих целей применяется оператор SELECT. Формат этого оператора следующий:

```
SELECT условие   A,B,C,D,E,F
```


Оператор SELECT находит элемент модели, удовлетворяющий заданному условию, и записывает номер этого элемента модели в заданный параметр транзакта, вошедшего в оператор SELECT.

Приведем некоторые из условий, которые могут применяться в операторе SELECT (полный список условий имеется в документации по GPSS World):

- состояния устройств: NU - устройство свободно, U - устройство занято;
- состояния многоканальных устройств: SE - многоканальное устройство пусто (т.е. в нем нет транзактов), SNE - многоканальное устройство не пусто, SF - многоканальное устройство заполнено (все каналы заняты), SNF - многоканальное устройство не заполнено (есть хотя бы один свободный канал);
- операторы отношения: MAX - максимальное значение СЧА, MIN - минимальное значение СЧА, L - меньше, LE - меньше или равно, E - равно; NE - не равно, G - больше, GE - больше или равно.

Между словом SELECT и обозначением условия должен быть ровно один пробел.

Назначение операндов оператора SELECT следующее:

A - номер параметра, в который записывается *номер объекта*, удовлетворяющего заданному условию;

B,C - нижняя и верхняя граница номеров объектов, для которых проверяется заданное условие;

D - величина, с которой сравнивается заданный СЧА объекта. Сам СЧА указывается в операнде E. Операнд D требуется указывать только в случае, если в качестве "условия" указан оператор отношения;

E - имя СЧА, для которого проверяется заданное условие;

F - метка оператора, на который должен перейти транзакт, если заданное условие не выполняется ни для одного из объектов модели.

Если заданное условие выполняется для какого-либо объекта модели, то транзакт переходит на оператор, следующий за оператором SELECT. Если операнд F не указан, то транзакт *в любом случае* переходит на оператор, следующий за оператором SELECT. Если в качестве условия указаны операторы отношения MIN или MAX, то указывать операнд F не требуется, так как эти условия выполняются всегда.

Примечание. Если при разработке GPSS-модели предполагается выбирать какой-либо элемент модели (устройство, очередь, многоканальное устройство) с использованием оператора SELECT, то этому элементу модели необходимо присвоить номер, а не символьное имя.

Пример 4.3. Приведем пример оператора SELECT, в котором используются все операнды. Пусть в некоторой GPSS-модели имеются многоканальные устройства с номерами от 1 до 5. Требуется выбрать из них устройство, имеющее не менее двух свободных каналов, и записать номер этого устройства в третий параметр транзакта. Если такого устройства не окажется, то транзакт

должен перейти на оператор с меткой KON. Эти действия реализуются следующим оператором SELECT:

```
SELECT GE 3,1,5,2,R,KON
```

Пример 4.4. На предприятие поступают заказы на ремонт приборов трех видов, из них 30% - приборы первого вида, 10% - второго, 60% - третьего. Поток заказов можно считать пуассоновским; в среднем поступает один прибор в три часа. На предприятии работают четыре ремонтника. Прибор направляется к свободному ремонтнику, а если все ремонтники заняты - то к тому из них, у которого меньше приборов, ожидающих ремонта. Время ремонта прибора - гауссовская случайная величина со средним значением 5 часов и стандартным отклонением 30 мин. После ремонта для приборов второго и третьего типа требуется регулировка на специальном стенде, занимающая от 40 мин до часа.

Разработать модель для анализа работы предприятия в течение 1000 часов. Подсчитать количество приборов, отремонтированных каждым ремонтником.

```
TIP FUNCTION      RN1,D3
0.3,1/0.4,2/1,3
      GENERATE      (EXPONENTIAL(1,0,180))
      ASSIGN        1, FN$TIP
      SELECT NU     2,1,4,,, VYB_OCH
      TRANSFER      , OBR
VYB_OCH SELECT MIN  2,1,4,,, Q
OBR      QUEUE      P2
      SEIZE         P2
      DEPART        P2
      ADVANCE       (NORMAL(2,300,30))
      RELEASE       P2
      SAVEVALUE     P2+,1
      TEST G        P1,1,KON
      QUEUE         OCH_STEND
      SEIZE         STEND
      DEPART        OCH_STEND
      ADVANCE       (UNIFORM(3,40,60))
      RELEASE       STEND
KON      TERMINATE
      GENERATE      60000
      TERMINATE     1
```

Оператор GENERATE (EXPONENTIAL(1,0,180)) имитирует пуассоновский поток приборов, поступающих на ремонт, со средним интервалом 3 часа (180 мин). В операторе ASSIGN 1, FN\$TIP первому параметру транзакта присваивается значение функции TIP, т.е. 1 (в 30% случаев), 2 (в 10%) или 3 (в 60% случаев). Таким образом имитируется поступление приборов трех типов.

В операторе SELECT NU 2,1,4,,, VYB_OCH из четырех одноканальных устройств с номерами (числовыми именами) 1, 2, 3 и 4 выбирается свободное устройство; для выбора свободного устройства используется условие NU. Эти четыре устройства имитируют четырех ремонтников. Номер выбранного устрой-

ства записывается во второй параметр транзакта, имитирующего поступивший прибор. Затем транзакт пропускается в следующий оператор (TRANSFER ,OBR) и передается на метку OBR, т.е. на вход в очередь. Если ни одно из устройств 1,2,3,4 не оказалось свободным, то транзакт передается на метку VYB_OCH.

В операторе SELECT MIN 2,1,4,,Q из очередей с номерами 1, 2, 3, 4 (очереди к ремонтникам) выбирается очередь минимальной длины; здесь Q – СЧА, обозначающий длину очереди. Номер этой очереди записывается во второй параметр. Затем транзакт поступает в следующий оператор, т.е. входит в очередь.

Примечание. Если несколько очередей одновременно имеют одинаковую (минимальную по сравнению с другими очередями) длину, то автоматически выбирается первая из них.

Оператор QUEUE P2 имитирует вход транзакта (прибора) в очередь, *номер которой* записан во втором параметре, т.е. выбран в одном из операторов SELECT. Оператор SEIZE P2 имитирует поступление прибора в одноканальное устройство, указанное в параметре P2. Оператор DEPART P2 имитирует выход из очереди, оператор ADVANCE (NORMAL (2,300,30)) - ремонт прибора, оператор RELEASE P2 - выход из устройства, т.е. окончание ремонта.

Примечание. Если транзакт поступает на оператор QUEUE P2 из оператора TRANSFER ,OBR (т.е. после того, как в операторе SELECT NU было выбрано свободное устройство), то он не задерживается в очереди, а сразу же поступает в свободное устройство, номер которого записан во втором параметре.

Следует еще раз обратить внимание, что в данном примере операторы QUEUE P2 и DEPART P2 имитируют не одну, а *четыре* очереди (с номерами 1,2,3,4). Аналогично, операторы SEIZE P2 и RELEASE P2 имитируют работу *четырех* одноканальных устройств (ремонтников).

В операторе SAVEVALUE P2+,1 выполняется подсчет приборов, отремонтированных *каждым* ремонтником. Когда в этот оператор поступает транзакт, увеличивается на единицу сохраняемая величина, указанная во втором параметре транзакта, т.е. номер устройства, в котором он обслуживался (и номер очереди к нему).

Операторы TEST G P1,1,KON ... RELEASE STEND имитируют регулировку на стенде для приборов второго и третьего типа.

Ниже приведены основные результаты моделирования.

FACILITY	ENTRIES	UTIL.	AVE. TIME				
1	127	0.632	298.469				
2	95	0.481	303.524				
3	63	0.313	297.816				
4	35	0.171	293.843				
STEND	235	0.195	49.854				
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)
1	1	0	127	112	0.039	18.457	156.269
2	1	0	95	89	0.017	10.961	173.555
3	1	0	63	59	0.009	8.707	137.137
4	1	0	35	33	0.004	7.267	127.164
OCH_STEND	2	1	236	184	0.025	6.421	29.141

SAVEVALUE	VALUE
1	127.000
2	95.000
3	63.000
4	35.000

Из полученных результатов видно, что имеются значительные простои ремонтников и стенда. Коэффициент загрузки второго ремонтника составляет всего 0,481, третьего - 0,313, четвертого - 0,171.

Очереди к ремонтникам (т.е. очереди из приборов, ожидающих ремонта) практически не возникали: они никогда не превышали одного прибора. Кроме того, из колонок ENTRY и ENTRY(0) видно, что большинство приборов поступало на ремонт сразу же (без ожидания в очереди).

Первым ремонтником было отремонтировано 127 приборов, вторым - 95, третьим - 63, четвертым - 35.

Выполним расчет среднего времени пребывания прибора в ремонтной службе. Найдем сначала среднее время пребывания прибора в очереди. Из результатов моделирования видно, что в очереди к первому ремонтнику за время моделирования находилось 127 приборов, ко второму – 95, к третьему – 63, к четвертому – 35 (колонок ENTRY). Известно также среднее время пребывания в каждой из очередей (колонок AVE.TIME). На основе этих данных вычислим среднее время пребывания прибора в очереди: $(127 \cdot 18,457 + 95 \cdot 10,961 + 63 \cdot 8,707 + 35 \cdot 7,267) / (127 + 95 + 63 + 35) = 13,09$ мин.

Среднее время ремонта прибора – 5 часов (300 минут). Кроме того, для приборов второго и третьего типа, составляющих 70% от всех приборов, требовалась регулировка на стенде. Среднее время пребывания в очереди к стенду составило 6,421 мин, среднее время регулировки на стенде – 50 мин.

Таким образом, среднее время пребывания прибора в ремонтной службе составило $13,09 + 300 + 0,7 \cdot (6,421 + 50) \approx 353$ мин.

Исходя из результатов моделирования, можно рекомендовать сократить количество ремонтников до трех или до двух (если нет возможности увеличения потока заказов). Чтобы выполнить моделирование нового варианта ремонтной службы, например, с тремя ремонтниками, достаточно изменить операторы SELECT (заменить в них цифру 4 на 3).

5. МОДЕЛИРОВАНИЕ ВЗАИМОСВЯЗАННЫХ ПРОЦЕССОВ

5.1. Проверка нескольких условий. Логические переменные

Во многих случаях в процессе моделирования требуется проверка некоторых условий. Если требуется проверить только одно условие, то обычно применяется оператор TEST, рассмотренный выше, или операторы LOGIC и GATE, рассматриваемые в подразд. 5.4 и 5.5. Для одновременной проверки нескольких (двух и более) условий удобно использовать логические переменные. Каждая логическая переменная описывается в начале модели (до первого оператора GENERATE) следующей командой:

ИМЯ_ПЕРЕМЕННОЙ BVARIABLE ЛОГИЧЕСКОЕ_ВЫРАЖЕНИЕ

где имя_переменной – имя логической переменной;

логическое_выражение – выражение, принимающее значение “истина” или “ложь”.

В логическом выражении обычно используются операции сравнения стандартных числовых атрибутов с некоторыми заданными величинами или друг с другом. Операции сравнения записываются обычным образом, например, знаком = обозначается “сравнение на равенство”, знаком >= - проверка условия “больше или равно”, и т.д. Кроме того, в логическом выражении обычно применяются логические операции: ‘NOT’ – отрицание, ‘AND’ – логическое И, ‘OR’ – логическое ИЛИ. Следует обратить внимание, что обозначения логических операций заключаются в кавычки.

Ссылка на логическую переменную указывается в модели в том месте, где требуется проверка условия, заданного этой переменной. Обычно значение логической переменной сравнивается с нулем или единицей в операторе TEST. Ссылка на логическую переменную имеет вид: BV\$имя_переменной.

Пример 5.1. В ремонтную службу предприятия поступают приборы для ремонта. Каждый прибор может с одинаковой вероятностью содержать от трех до семи неисправных деталей. Поток приборов - пуассоновский; средний интервал между моментами поступления приборов - полчаса. Ремонтная служба работает круглосуточно.

В ремонтной службе работают два ремонтника. Ремонт прибора включает следующие операции: осмотр прибора и замена неисправных деталей. Осмотр прибора занимает от 10 до 20 мин. Время замены *одной* неисправной детали - гауссовская случайная величина со средним значением 5 мин и стандартным отклонением 30 с.

В начале работы в ремонтной службе имеется запас деталей, используемых для замены: 250 деталей. Каждые 24 часа этот запас пополняется до 250 деталей.

Требуется разработать GPSS-модель для анализа работы ремонтной службы в течение 10 суток.

Здесь для начала ремонта прибора необходимо выполнение двух условий: во-первых, наличие свободного ремонтника, во-вторых, наличие деталей в количестве, достаточном для ремонта данного прибора.

```

REMONТNIK  STORAGE      2
              INITIAL    X$ZAPAS,250
GOTOV      BVARIABLE    'NOT' (SF$REMONТNIK=1) 'AND' (X$ZAPAS>=P1)
              GENERATE    (EXPONENTIAL(1,0,30))
              ASSIGN      1, (DUNIFORM(2,3,7))
              QUEUE       OCH
              TEST E      BV$GOTOV,1
              ENTER       REMONТNIK
              DEPART      OCH
              SAVEVALUE   ZAPAS-,P1
              ADVANCE     15,5
ZAMENA      ADVANCE     (NORMAL(3,5,0.5))
              LOOP        1,ZAMENA
              LEAVE       REMONТNIK
              TERMINATE
              GENERATE    1440
              SAVEVALUE   ZAPAS,250
              TERMINATE
              GENERATE    14400
              TERMINATE  1

```

Здесь в команде INITIAL X\$ZAPAS,250 устанавливается начальное значение сохраняемой величины ZAPAS, равное 250, т.е. начальный запас деталей. Напомним, что обозначение X\$ИМЯ представляет собой ссылку на сохраняемую величину.

Примечание. Команда INITIAL выполняется перед началом процесса моделирования, т.е. до того как начнется имитация поступления транзактов и их обработки.

Логическая переменная GOTOV принимает значение “истина”, когда имеется возможность начала работы с прибором. Для этого требуется одновременное выполнение двух условий: наличие хотя бы одного свободного ремонтника и достаточный запас деталей. Первое из этих условий задается выражением 'NOT' (SF\$REMONТNIK=1): стандартный числовой атрибут SF для многоканального устройства REMONТNIK не равен единице, т.е. устройство не заполнено. Второе условие задается выражением (X\$ZAPAS>=P1): значение сохраняемой величины ZAPAS должно быть не меньше, чем значение первого параметра транзакта. В первый параметр транзакта будет записываться количество неисправных деталей в приборе. Так как для работы с прибором требуется выполнение обоих условий, они связаны логической операцией ‘AND’.

Примечание. Логическое выражение для переменной GOTOV можно было записать и по-другому: (SF\$REMONТNIK=0) 'AND' (X\$ZAPAS>=P1) .

Оператор GENERATE (EXPONENTIAL(1,0,30)) имитирует поступление приборов на ремонт. В операторе ASSIGN 1,(DUNIFORM(2,3,7)) в первый

параметр транзакта записывается количество деталей, требующих замены, т.е. число 3, 4, 5, 6 или 7 (с одинаковой вероятностью). Для этого используется функция `DUNIFORM(2,3,7)`, имитирующая дискретную равномерную случайную величину в диапазоне от 3 до 7. Здесь 2 – номер генератора `СРРЧ` (конечно, мог использоваться и другой номер генератора); 3 и 7 – границы возможных значений имитируемой величины.

Оператор `QUEUE OCH` имитирует поступление приборов в очередь к ремонтникам. В операторе `TEST E BV$GOTOV,1` проверяется значение логической переменной `GOTOV`. Если эта переменная равна единице (т.е. имеет значение “истина”), это значит, что выполняются условия, необходимые для ремонта прибора. В этом случае транзакт пропускается дальше, т.е. имитируется поступление прибора к ремонтнику. Если переменная `GOTOV` не равна единице, это означает отсутствие свободного ремонтника или нехватку деталей для ремонта. Транзакт задерживается в операторе `TEST` до момента, когда переменная `GOTOV` примет значение “истина”.

Операторы `ENTER REMONTNIK` и `DEPART OCH` имитируют поступление приборов к ремонтнику и их выход из очереди.

В операторе `SAVEVALUE ZAPAS-,P1` имитируется уменьшение запаса деталей на величину параметра `P1`, т.е. на такое количество деталей, которое необходимо для ремонта.

Оператор `ADVANCE 15,5` имитирует осмотр прибора. Операторы

```
ZAMENA ADVANCE (NORMAL(3,5,0.5))
      LOOP 1,ZAMENA
```

имитируют замену деталей. Операция замены повторяется столько раз, сколько деталей требуется заменить; количество заменяемых деталей указано в первом параметре транзакта.

Операторы `LEAVE REMONTNIK` и `TERMINATE` имитируют окончание обслуживания прибора.

Группа операторов

```
GENERATE      1440
SAVEVALUE     ZAPAS,250
TERMINATE
```

имитирует пополнение запаса деталей через каждые 24 часа (1440 минут). Таким образом, каждые 24 часа запас деталей, имитируемый сохраняемой величиной `ZAPAS`, становится равным 250.

Приведем результаты моделирования:

QUEUE										
OCH	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY		
	20	20	511	135	2.790	78.635	106.869	0		
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
REMONTNIK	2	2	0	2	491	1	1.356	0.678	20	0

Видно, что в работе ремонтной службы имеется недостаток: значительное время ожидания ремонта приборов. Среднее время ожидания ремонта (т.е. пре-

бывания прибора в очереди) составило 78,635 мин. Количество приборов, ожидающих ремонта, достигало 20. При этом загрузка ремонтников невелика: коэффициент загрузки равен 0,678, или 67,8%. По-видимому, длительное ожидание ремонта связано не с загруженностью ремонтников, а с нехваткой деталей, необходимых для ремонта. Анализ данного недостатка и его устранение будут подробно рассмотрены в подразд.8.2.

5.2. Создание ансамблей транзактов

В GPSS имеется возможность создания любого количества копий транзакта, имеющегося в модели. Транзакты, созданные путем копирования некоторого транзакта, называются транзактами одного ансамбля (семейства). Для создания ансамблей транзактов используется следующий оператор:

SPLIT A,B

где A - количество создаваемых копий транзакта;

В - метка оператора, на который передаются все созданные копии.

Исходный транзакт (т.е. транзакт, вошедший на оператор SPLIT) передается в следующий оператор. Если операнд В не указан, то все созданные копии также передаются в следующий оператор.

Примечание. Таким образом, после каждого входа транзакта на оператор SPLIT из этого оператора выходит A+1 транзакт (исходный транзакт и A копий).

Пример 5.2. Оператор GENERATE 12,2 (создание транзактов через каждые 12±2 минуты) можно заменить следующей группой операторов:

GENERATE	,, , 1
MET	ADVANCE 12, 2
	SPLIT 1, MET

Здесь оператор GENERATE создает *один* транзакт. Он задерживается на 12±2 минуты и попадает на оператор SPLIT. Оттуда этот транзакт передается в следующий оператор модели, а его копия - на метку MET (задержка на 12±2 минуты). После этого копия первого транзакта снова попадает на оператор SPLIT, создается новая копия, и т.д.

Пример 5.3. Имитируется поступление деталей на обработку. Детали поступают партиями по 30 штук и обрабатываются по одной. Интервалы времени между партиями можно считать экспоненциальными случайными величинами; в среднем интервал между партиями составляет 40 мин.

Фрагмент модели, имитирующий поступление партий, может иметь следующий вид:

```
GENERATE    (EXPONENTIAL(1, 0, 40))
SPLIT       29
... имитация обработки деталей...
```


Здесь оператор `GENERATE (EXPONENTIAL (1,0,40))` имитирует поступление *партий* деталей. Оператор `SPLIT 29` имитирует поступление *30 деталей* на дальнейшую обработку.

Транзакты одного ансамбля, созданные оператором `SPLIT`, могут обрабатываться точно так же, как и созданные оператором `GENERATE`, т.е. входить в очереди, устройства и многоканальные устройства, учитываться в операторах `SAVEVALUE` и т.д. Кроме того, с транзактами, созданными с помощью оператора `SPLIT`, можно выполнять дополнительные операции: объединение нескольких транзактов в один (оператор `ASSEMBLE`), накопление нескольких транзактов и их одновременное пропускание в следующий оператор (оператор `GATHER`), согласование продвижения транзактов по модели (оператор `MATCH`). Для транзактов, созданных оператором `GENERATE`, такие операции *невозможны*.

Примечания.

1. Если транзакт, вошедший на оператор `SPLIT`, имел некоторые значения параметров (заданные оператором `ASSIGN`), то все его копии будут иметь те же значения параметров. Их можно затем изменять оператором `ASSIGN`.

2. Транзакт, созданный оператором `SPLIT`, может затем также использоваться для создания новых транзактов в операторе `SPLIT`. Все транзакты, имеющие один общий исходный транзакт, принадлежат одному ансамблю.

5.3. Объединение нескольких транзактов в один

Для объединения заданного количества транзактов в один транзакт используется следующий оператор:

```
ASSEMBLE      A
```

где *A* - количество объединяемых транзактов.

Пусть, например, указан оператор `ASSEMBLE 3`. Когда в этот оператор поступает транзакт, он не пропускается дальше, а остается в операторе `ASSEMBLE` до тех пор, пока в него не поступят еще два транзакта (т.е. пока в операторе не будет находиться три транзакта). После этого три транзакта объединяются в один (точнее, первый транзакт пропускается в следующий оператор, остальные – уничтожаются).

Все транзакты, объединяемые в операторе `ASSEMBLE`, должны принадлежать *одному ансамблю*, т.е. должны быть созданы оператором `SPLIT` (одним или несколькими) из одного исходного транзакта. Поэтому, например, следующий фрагмент модели является *неправильным*:

```
GENERATE      (EXPONENTIAL (1, 0, 20))  
ASSEMBLE      3
```

Правильно реализовать такой фрагмент модели можно следующим образом:

```

GENERATE    , , , 1
MET ADVANCE  (EXPONENTIAL(1, 0, 20))
SPLIT       1, MET
ASSEMBLE    3

```

Важное (но не единственное) применение оператора ASSEMBLE - имитация процессов сборки изделий и составления комплектов изделий в моделях производственных систем.

Применение оператора ASSEMBLE рассматривается в примере 5.4.

5.4. Логические переключатели

Логические переключатели (называемые также просто переключателями или ключами) - это элементы GPSS-модели, которые могут находиться только в одном из двух состояний: установленном или сброшенном. Логические переключатели обычно используются для разрешения или запрещения движения транзактов через определенную часть модели. Для этого выполняется проверка состояния переключателей, например, с помощью рассматриваемого ниже оператора GATE.

Для управления состоянием логических переключателей используется оператор LOGIC:

```
LOGIC операция      А
```

где операция - обозначение операции, выполняемой с логическим переключателем: S - установка, R - сброс, I - инвертирование (изменение текущего состояния на противоположное);

А - имя переключателя (числовое или символьное), с которым выполняется заданная операция.

Какое-либо объявление логических переключателей не требуется. По умолчанию в начале работы модели все логические переключатели сброшены.

Применение логических переключателей рассматривается в примере 5.4.

5.5. Управление движением транзактов в зависимости от состояния элементов модели

Во многих случаях выбор направления движения транзактов производится в зависимости от состояния устройств, многоканальных устройств или переключателей, используемых в модели. Для этого применяется оператор TEST, рассмотренный ранее, и оператор GATE. Формат оператора GATE следующий.

```
GATE условие      А,В
```

Оператор GATE проверяет, соответствует ли заданный элемент модели указанному условию, и в зависимости от этого выбирает направление дальнейшего движения транзакта (или задерживает этот транзакт на все время, пока заданное условие не выполняется). Некоторые из условий, которые могут задаваться в операторе GATE, следующие (полный список условий имеется в [1]):

- для устройств: NU - устройство свободно, U - устройство занято;
- для многоканальных устройств: SE - многоканальное устройство пусто (т.е. в нем нет транзактов), SNE - многоканальное устройство не пусто, SF - многоканальное устройство заполнено, SNF - многоканальное устройство не заполнено;
- для логических переключателей: LS - переключатель установлен, LR - переключатель сброшен.

Назначение операндов оператора GATE следующее:

А - имя устройства, многоканального устройства или логического переключателя, для которого проверяется заданное условие;

В - метка оператора, на который должен перейти транзакт, если заданное условие *не выполняется* для элемента модели, указанного в операнде А.

Операнд А обязателен, В - необязателен.

Оператор GATE работает следующим образом. Для элемента модели, указанного в операнде А, проверяется заданное условие. Если оно выполняется, то транзакт пропускается на оператор, следующий за GATE. Если условие не выполняется, то транзакт передается на оператор, метка которого указана в операнде В, а если операнд В не указан - то задерживается (т.е. остается в операторе GATE) на все время, пока не выполняется условие.

Пример 5.4. В ремонтную службу предприятия поступают приборы для ремонта. Поток приборов - пуассоновский; в среднем на ремонт поступает три прибора в час.

Каждый прибор состоит из четырех блоков, причем все блоки требуют ремонта. Блоки, входящие в один прибор, могут ремонтироваться независимо друг от друга разными ремонтниками.

В ремонтной службе работают два ремонтника. Время ремонта одного *блока* - экспоненциальная случайная величина со средним значением 30 мин.

После ремонта всех блоков, входящих в прибор, требуется регулировка *прибора* на специальном стенде. Регулировка занимает от 20 до 40 мин.

Приборы поступают в ремонтную службу только в течение рабочего дня (восемь часов). Ремонтная служба работает круглосуточно.

Требуется разработать GPSS-модель для анализа работы ремонтной службы в течение 30 суток.

```

REMONTEK STORAGE 2
GENERATE (EXPONENTIAL(1,0,20))
GATE LS RABOTA,NERAB
SPLIT 3
QUEUE OCH
ENTER REMONTEK
DEPART OCH
ADVANCE (EXPONENTIAL(2,0,30))
LEAVE REMONTEK
ASSEMBLE 4
QUEUE OCH_STEND

```

```

        SEIZE      STEND
        DEPART    OCH_STEND
        ADVANCE   30,10
        RELEASE   STEND
        SAVEVALUE OTREM+,1
NERAB   TERMINATE
;
        GENERATE  ,,1
NACHALO LOGIC S   RABOTA
        ADVANCE   480
        LOGIC R   RABOTA
        ADVANCE   960
        TRANSFER  ,NACHALO
;
        GENERATE  43200
        TERMINATE 1

```

Здесь оператор `GENERATE (EXPONENTIAL(1,0,20))` имитирует поток приборов, поступающих на ремонт.

Для имитации рабочего времени (т.е. восьмичасовых интервалов, когда приборы поступают в ремонтную службу) используется логический переключатель `RABOTA`. Пусть рабочее время имитируется установленным состоянием этого переключателя, а нерабочее – сброшенным состоянием (конечно, можно и наоборот). Состояние переключателя проверяется в операторе `GATE LS RABOTA,NERAB`. Если переключатель установлен, то транзакт пропускается дальше; таким образом имитируется поступление прибора на ремонт. Если переключатель сброшен (т.е. условие, указанное в операторе `GATE`, не выполняется), то транзакт передается на метку `NERAB`. Эта метка указана в операторе `TERMINATE`, т.е. транзакт удаляется из модели без каких-либо действий с ним.

Примечание. Изменение состояния переключателя `RABOTA` (т.е. имитация начала или окончания рабочего дня) рассматривается в данном примере ниже.

В операторе `SPLIT 3` создаются три копии поступившего транзакта. Таким образом, в следующий оператор поступают уже *четыре* транзакта. Так имитируется поступление четырех блоков, из которых состоит прибор.

Операторы `QUEUE OCH...LEAVE REMONTNIK` имитируют работу ремонтников, т.е. ремонт блоков.

В операторе `ASSEMBLE 4` имитируется сборка отремонтированных блоков в прибор.

Операторы `QUEUE OCH_STEND...RELEASE STEND` имитируют работу стенда для регулировки приборов после ремонта. Оператор `SAVEVALUE OTREM+,1` требуется для подсчета количества отремонтированных приборов.

Группа операторов `GENERATE ,,1... TRANSFER ,NACHALO` управляет состоянием логического переключателя `RABOTA`, т.е. имитирует начало и конец рабочего дня.

Оператор GENERATE „,1 имитирует один транзакт, используемый для управления состоянием логического переключателя. В операторе LOGIC S RABOTA (с меткой NACHALO) логический переключатель RABOTA переводится в состояние “установлен”. Затем транзакт задерживается в операторе ADVANCE 480 на 480 минут (8 часов). В течение этого времени состояние переключателя RABOTA не изменяется, т.е. он остается установленным. Таким образом, в это время *в начале модели* выполняется условие, указанное в операторе GATE LS RABOTA,NERAB, т.е. поступающие транзакты пропускаются в модель.

После выхода из оператора ADVANCE 480 транзакт поступает на оператор LOGIC R RABOTA: логический переключатель RABOTA сбрасывается. Таким образом имитируется окончание рабочего дня. Транзакт задерживается в операторе ADVANCE 960 на 960 минут (16 часов); таким образом, переключатель RABOTA на это время остается сброшенным. Поэтому в это время *в начале модели* условие, указанное в операторе GATE LS RABOTA,NERAB, не выполняется; поступающие транзакты передаются на выход из модели (метка NERAB).

После выхода из оператора ADVANCE 960 транзакт поступает на оператор TRANSFER ,NACHALO, т.е. возвращается на оператор LOGIC S RABOTA. Переключатель RABOTA снова переводится в состояние “установлен”. Таким образом имитируется начало нового рабочего дня.

Пример 5.5. Пусть в условиях примера 5.4 приборы на ремонт поступают круглосуточно, а ремонтная служба, наоборот, работает только в течение восьмичасового рабочего дня.

В модель, составленную для примера 5.4, потребуется внести следующие изменения:

```

REMONТNIK STORAGE      2
                   GENERATE (EXPONENTIAL (1, 0, 20) )
                   SPLIT      3
                   QUEUE      OCH
                   GATE LS    RABOTA
                   ENTER      REMONТNIK

```

...см. пример 5.4...

Транзакт, поступивший в оператор GATE LS RABOTA, пропускается дальше (в устройство REMONТNIK), если логический переключатель RABOTA находится в состоянии “установлен”. Если транзакт поступает в то время, когда этот переключатель сброшен, то транзакт *ожидает*, пока переключатель перейдет в состояние “установлен”. Как показано в примере 5.4, логический переключатель RABOTA находится в состоянии “установлен” в течение восьми часов модельного времени, затем – в состоянии “сброшен” в течение 16 часов модельного времени, и т.д.

Примечание. Метка NERAB, имеющаяся в примере 5.4, в примере 5.5 не требуется.

5.6. Синхронизация движения транзактов в модели

Для синхронизации (согласования) движения двух транзактов в модели используется следующий оператор:

метка MATCH A

где A - метка другого оператора MATCH, называемого *сопряженным* с данным оператором.

Метка и операнд A обязательны.

В модели всегда должна быть *пара* операторов MATCH, причем метка одного из них должна совпадать с операндом A другого, и наоборот.

Примечание. В модели может быть несколько пар операторов MATCH.

Операторы MATCH работают следующим образом. Транзакт, поступивший на оператор MATCH, не пропускается дальше до тех пор, пока в другой (сопряженный) оператор MATCH не войдет транзакт, принадлежащий тому же ансамблю. Как только в обоих операторах MATCH оказывается по одному транзакту, оба транзакта (одновременно) пропускаются на операторы, следующие за операторами MATCH.

Пример 5.5. Для сборки некоторого изделия требуется три одинаковых детали, закрепляемых на плате.

Детали поступают в цех с интервалами от 5 до 15 мин. Для деталей требуется обработка, выполняемая на станке. В цехе имеется один станок. Время обработки детали - гауссовская случайная величина со средним значением 7 мин и стандартным отклонением 1,5 мин.

Платы поступают в цех в среднем через каждые полчаса; поток плат - пуассоновский. Для плат никакой обработки не требуется.

Закрепление одной детали на плате занимает от 3 до 7 мин. В цехе имеется одно рабочее место для сборки изделий.

Требуется разработать модель для анализа работы цеха в течение 30 суток.

При разработке модели этой производственной системы необходимо, чтобы в ту часть модели, где будет имитироваться сборка готового изделия, поступали три транзакта, имитирующие детали, и один транзакт, имитирующий плату. Для этого будут использоваться операторы MATCH.

	GENERATE	, , , 1
	SPLIT	1, PLATA
DETAL	ADVANCE	10, 5
	SPLIT	1, DETAL
	QUEUE	OCH_STAN
	SEIZE	STANOK
	DEPART	OCH_STAN
	ADVANCE	(NORMAL (1, 7, 1.5))
	RELEASE	STANOK
	ASSEMBLE	3

```

DET3      MATCH      PLATA1
          TRANSFER    , SBORKA
PLATA     ADVANCE     (EXPONENTIAL (2, 0, 30))
          SPLIT       1, PLATA
PLATA1    MATCH       DET3
SBORKA    ASSEMBLE    2
          ASSIGN      1, 3
          QUEUE       OCH_SBOR
          SEIZE       SBOR
          DEPART      OCH_SBOR
ZAKREPL   ADVANCE     5, 2
          LOOP        1, ZAKREPL
          RELEASE     SBOR
          SAVEVALUE   IZDEL+, 1
          TERMINATE
          GENERATE    14400
          TERMINATE   1

```

В операторе GENERATE „,1 создается один транзакт. Из этого транзакта с помощью операторов SPLIT будут создаваться *все остальные* транзакты, имитирующие детали и платы. Таким образом, все они будут относиться к одному ансамблю. Это необходимо потому, что в модели будут использоваться операторы ASSEMBLE (для имитации сборки) и MATCH (для согласования поступления деталей и плат на сборку), а эти операторы работают только с транзактами из одного ансамбля.

Оператор SPLIT 1,PLATA создает одну копию поступившего транзакта. Исходный транзакт пропускается в следующий оператор (ADVANCE 10,5); этот транзакт будет использоваться *для имитации потока деталей*. Копия этого транзакта, передаваемая на метку PLATA, будет использоваться *для имитации потока плат*.

Операторы

```

DETAL     ADVANCE     10, 5
          SPLIT       1, DETAL

```

имитируют поступление деталей, как показано в подразд.5.2.

Операторы QUEUE OCH_STAN...RELEASE STANOK имитируют обработку деталей на станке.

Оператор ASSEMBLE 3 объединяет три транзакта (имитирующие детали) в один транзакт. Таким образом, имитируется *комплект из трех деталей*. Полученный транзакт поступает на оператор DET3 MATCH PLATA1. Из этого оператора транзакт пропускается дальше (на оператор TRANSFER ,SBORKA) только тогда, когда в другой оператор MATCH (PLATA1 MATCH DET3) также поступает транзакт, имитирующий *плату*.

Операторы

```

PLATA     ADVANCE     (EXPONENTIAL (2, 0, 30))
          SPLIT       1, PLATA

```

имитируют поступление плат, как показано в подразд.5.2.

Транзакты, имитирующие платы, поступают на оператор PLATA1 MATCH DET3. Транзакт, поступивший в этот оператор, пропускается дальше (на оператор TRANSFER ,SBORKA) только тогда, когда в другой оператор MATCH (DET3 MATCH PLATA1) также поступает транзакт, имитирующий комплект из трех деталей.

Оба транзакта из операторов MATCH *одновременно* пропускаются на операторы, следующие за операторами MATCH. Транзакт из оператора DET3 MATCH PLATA1, имитирующий комплект из трех деталей, попадает на оператор TRANSFER ,SBORKA и передается на оператор SBORKA ASSEMBLE 2. Транзакт из оператора PLATA1 MATCH DET3, имитирующий плату, также попадает на оператор SBORKA ASSEMBLE 2. В этом операторе два транзакта объединяются в один. Полученный транзакт имитирует комплект из трех деталей и платы.

В операторе ASSIGN 1,3 первому параметру транзакта присваивается значение 3. Этот параметр будет затем использоваться в качестве счетчика циклов при имитации закрепления деталей на плате.

Операторы QUEUE OCH_SBOR...RELEASE SBOR имитируют работу сборщика готовых изделий. Оператор SAVEVALUE IZDEL+,1 выполняет подсчет выпущенных изделий.

6. СРЕДСТВА АНАЛИЗА ВРЕМЕНИ В GPSS-МОДЕЛЯХ

6.1. Анализ времени пребывания транзакта в модели

С каждым транзактом в GPSS-модели связан стандартный числовой атрибут (СЧА) M1, значение которого представляет собой время с момента создания данного транзакта в операторе GENERATE, т.е. время пребывания транзакта в модели. Этот атрибут, как и любой другой СЧА, может использоваться в качестве операнда в операторах языка GPSS. Таким образом, время пребывания транзакта в модели можно сравнивать с некоторой заданной величиной (в операторе TEST), присваивать сохраняемой величине (оператором SAVEVALUE), записывать в параметр транзакта (оператором ASSIGN) и т.д.

Иногда требуется во время моделирования устанавливать значение M1 равным нулю, чтобы отсчитывать время пребывания транзакта в модели с некоторого заданного момента, а не с момента создания транзакта. Для этого используется следующий оператор:

MARK

При прохождении транзакта через оператор MARK значение атрибута M1 у этого транзакта устанавливается равным нулю.

Примечание. Важно учитывать, что атрибут M1 (если он не был изменен с помощью оператора MARK) содержит время пребывания транзакта в модели *именно* с момента его создания в операторе GENERATE. Если транзакт создается не в операторе GENERATE, а в операторе SPLIT (т.е. представляет собой копию другого транзакта), то в момент его создания

СЧА M1 равен *не нулю*, а значению атрибута M1 исходного транзакта. Чтобы значение M1 (т.е. время пребывания в модели) для такого транзакта отсчитывалось с момента его создания, требуется использовать оператор MARK.

Пример 6.1. Пусть в условиях примера 3.1 качество обслуживания заказчиков считается приемлемым, если заказ выполняется со скоростью не менее четырех листов в день. Другими словами, перевод размером до четырех листов должен быть выполнен не более чем за один день, перевод размером от четырех до восьми листов – не более чем за два дня, и т.д. Если перевод выполняется за более длительный срок, то плата заказчика уменьшается, и прибыль от одного переведенного листа составляет не 7,5, а только 5 ден.ед.

Требуется разработать модель для анализа работы организации в течение года (300 рабочих дней) с учетом этих условий.

При решении этой задачи важно понимать, что время *выполнения заказа* – это время от поступления заказа в организацию, выполняющую переводы, до получения заказчиком готового перевода. Это время включает не только *время работы переводчика над заказом*, но и *время ожидания* начала такой работы (если в момент поступления заказа все переводчики оказались занятыми).

В ходе моделирования для каждого заказа потребуется определять *необходимое* время его выполнения. Эта величина будет сравниваться с *фактическим* временем выполнения заказа. В зависимости от того, выполнен ли заказ своевременно, определяется прибыль организации.

```
PEREV  STORAGE      7
        GENERATE     (EXPONENTIAL(1,0,0.2))
        ASSIGN       1, (EXPONENTIAL(2,0,8))
        ASSIGN       2, (INT(P1/4)+1)
        QUEUE        OCH
        ENTER         PEREV
        DEPART        OCH
        ADVANCE       (P1/6)
        LEAVE         PEREV
        SAVEVALUE     KOL_VYP+, 1
        TEST LE       M1, P2, NESV
        SAVEVALUE     PRIB+, (P1#7.5)
        TRANSFER      , KON
NESV    SAVEVALUE     PRIB+, (P1#5)
        SAVEVALUE     KOL_NESV+, 1
KON     TERMINATE
        GENERATE      300
        TERMINATE     1
```

В операторе ASSIGN 2, (INT(P1/4)+1) определяется необходимое время выполнения перевода (количество дней). Эта величина записывается во второй параметр транзакта, имитирующего заказ. Здесь INT – стандартная функция GPSS, выделяющая целую часть аргумента.

В операторе TEST LE M1, P2, NESV сравнивается фактическое время выполнения заказа (время пребывания транзакта в модели, M1) и необходимое

время его выполнения (второй параметр транзакта). Если $M1 \leq P2$, то вычисляется прибыль при своевременном выполнении заказа, в противном случае – уменьшенная прибыль. Кроме того, в модели предусмотрен подсчет общего количества выполненных заказов (сохраняемая величина KOL_VYP) и количества заказов, выполненных несвоевременно (KOL_NESV).

Приведем фрагмент результатов моделирования:

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY		
OCH	28	9	1502	176	9.345	1.867	2.114	0		
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
PEREV	7	0	0	7	1493	1	6.735	0.962	0	9
SAVEVALUE	RETRY		VALUE							
KOL_VYP	0		1486.000							
PRI ^B	0		76293.880							
KOL_NESV	0		938.000							

Видно, что значительное количество заказов выполняется несвоевременно (938 из 1486). Причина – чрезмерная загрузка переводчиков (коэффициент загрузки – 0,962). На это же указывает значительная очередь заказов, ожидающих выполнения (среднее время ожидания в очереди – 1,867 дня, среднее количество заказов, ожидающих в очереди – 9,345, максимальное – 28).

Для устранения этих недостатков можно попытаться увеличить количество переводчиков. Для этого достаточно изменить операнд команды STORAGE. Будем считать, что увеличение количества переводчиков не связано для организации с дополнительными затратами (это возможно, если переводчикам выплачиваются деньги только за фактически выполненные заказы). В табл.6.1 приведены основные результаты моделирования для 8, 9, 10, 11, 12 переводчиков.

Таблица 6.1

Количество переводчиков	8	9	10	11	12
Коэффициент загрузки переводчиков	0,847	0,753	0,678	0,616	0,565
Общее количество выполненных заказов	1495	1496	1496	1496	1496
Количество заказов, выполненных несвоевременно	213	66	12	2	0
Прибыль, ден.ед.	88 811	90 435	90 898	90 990	91 004

Таким образом, для своевременного выполнения всех заказов и получения максимальной прибыли требуется 12 переводчиков. Однако с учетом необходимости обеспечения достаточной загрузки переводчиков, возможно, более рациональный вариант – использование 10 или 11 переводчиков. При этом достигается приемлемая загрузка переводчиков, а количество заказов, выполненных несвоевременно, достаточно мало.

Другие примеры решения задач, связанных с анализом времени пребывания транзакта в модели, будут рассмотрены в разд.7.

6.2. Время моделирования

С любой GPSS-моделью связаны два СЧА, содержащие информацию о времени моделирования: C1 (относительное время) и AC1 (абсолютное время).

Эти СЧА являются системными, т.е. они связаны не с каким-либо элементом модели (транзактом, устройством и т.д.), а с моделью в целом. Абсолютное время - это время с момента выполнения последней команды CLEAR, а если такая команда не использовалась - то с момента начала моделирования. Относительное время - это время с момента выполнения последней команды RESET; если эта команда не использовалась, то относительное время представляет собой время с момента выполнения последней команды CLEAR, а если и она не использовалась - то с момента начала моделирования. Если ни команда RESET, ни команда CLEAR не использовалась, то значения абсолютного и относительного времени (т.е. СЧА AC1 и C1) совпадают.

Примечание. Команда RESET удаляет из памяти все накопленные за время моделирования статистические данные. Команда CLEAR удаляет все статистические данные, а также удаляет из модели все транзакты. Команды RESET и CLEAR применяются для решения сложных задач, например, связанных с анализом нескольких периодов работы объекта моделирования, когда требуется получить информацию по каждому из таких периодов отдельно. В данном пособии эти команды не рассматриваются. Сведения об этих командах приводятся в [1].

Пример 6.2. Торговое предприятие продает некоторые изделия. Спрос на изделия является сезонным: летом он снижается. Поток покупателей можно считать пуассоновским. Летом на предприятие в среднем обращается шесть покупателей в день, в остальное время – десять покупателей в день. Каждый покупатель приобретает от одного до пяти изделий.

Летом в начале каждого дня запас изделий на складе доводится до 18 изделий, в другие сезоны – до 30 изделий. Имеется начальный запас в размере 30 изделий.

Если на складе имеется меньше изделий, чем требуется покупателю, то покупатель приобретает столько изделий, сколько имеется на складе. Если склад пуст, то покупатель не приобретает ничего.

Требуется разработать GPSS-модель для анализа работы предприятия в течение года. Предусмотреть подсчет общего количества проданных изделий, а также количества случаев, когда спрос покупателя не был удовлетворен (т.е. покупатель купил меньше товара, чем ему требовалось, или не купил ничего).

Для простоты будем считать, что в году 365 дней, причем к летнему периоду относятся дни со 151-го по 240-й.

	INITIAL	X\$SKLAD, 30
LETO	BVARIABLE	(C1>=151) 'AND' (C1<=240)
	GENERATE	(EXPONENTIAL(1, 0, 0.1))
	TEST E	BV\$LETO, 0, NET
	TRANSFER	, ZAKAZ
	;	
	GENERATE	(EXPONENTIAL(2, 0, 0.167))
	TEST E	BV\$LETO, 1, NET
	TRANSFER	, ZAKAZ
	;	

```

ZAKAZ      ASSIGN      1, (DUNIFORM(3,1,5))
           TEST LE      P1,X$SKLAD,NEHVATKA
           SAVEVALUE     SKLAD-,P1
           SAVEVALUE     PRODANO+,P1
           TERMINATE
;
NEHVATKA   SAVEVALUE     NE_HVATILO+,1
           TEST NE      X$SKLAD,0,NET
           SAVEVALUE     PRODANO+,X$SKLAD
           SAVEVALUE     SKLAD,0
NET         TERMINATE
;
           GENERATE      1
           TEST E        BV$LETO,1,NE_LETO
           SAVEVALUE     SKLAD,18
           TERMINATE      1
NE_LETO    SAVEVALUE     SKLAD,30
           TERMINATE      1

```

Модель запускается на выполнение командой **START 365**.

Здесь логическая переменная **LETO** принимает значение “истина”, если выполняется условие $151 \leq C1 \leq 240$ (т.е. время моделирования относится к летнему периоду).

Группа операторов

```

GENERATE      (EXPONENTIAL(1,0,0.1))
TEST E        BV$LETO,0,NET
TRANSFER      ,ZAKAZ

```

имитирует поток покупателей в течение всего года, кроме лета. Оператор **GENERATE (EXPONENTIAL(1,0,0.1))** имитирует поток покупателей с интервалом 0,1 дня (т.е. 10 покупателей в день). В операторе **TEST E BV\$LETO,0,NET** проверяется, когда поступил транзакт, имитирующий покупателя. Если логическая переменная **LETO** равна нулю (т.е. принимает значение “ложь”), это означает, что смоделирован приход покупателя до наступления летнего периода или после его окончания. Транзакт пропускается на следующий оператор, а оттуда - на метку **ZAKAZ**, т.е. на группу операторов, имитирующих обслуживание покупателя. Если логическая переменная **LETO** не равна нулю (т.е. принимает значение “истина”), значит, смоделирован приход покупателя в летний период. Транзакт направляется на метку **NET**, т.е. на оператор **TERMINATE**. Таким образом, транзакты из оператора **GENERATE (EXPONENTIAL(1,0,0.1))** поступают в ту часть модели, где имитируется обслуживание покупателей, только в то время, когда $C1 < 151$ или $C1 > 240$.

Аналогичным образом группа операторов

```

GENERATE      (EXPONENTIAL(2,0,0.167))
TEST E        BV$LETO,1,NET
TRANSFER      ,ZAKAZ

```

имитирует поток покупателей в летний период.

Группа операторов `ZAKAZ ASSIGN 1, (DUNIFORM(3, 1, 5)) ...NET TERMINATE` имитирует обслуживание покупателей (или отказ в обслуживании, если на складе не оказалось нужного количества изделий).

Группа операторов `GENERATE 1...TERMINATE 1` имитирует пополнение склада. Оператор `GENERATE 1` создает транзакты с интервалом в одну единицу времени; таким образом имитируется ежедневное поступление изделий на склад. С помощью оператора `TEST E BV$LETO, 1, NE_LETO` транзакт направляется на оператор `SAVEVALUE SKLAD, 18`, если он поступил в интервале времени $151 \leq C1 \leq 241$, или на оператор `SAVEVALUE SKLAD, 30` (с меткой `NE_LETO`), если он поступил в интервале времени $C1 < 151$ или $C1 > 240$.

Примечание. В рассмотренной модели вместо СЧА $C1$ можно было использовать $AC1$.

7. МОДЕЛИРОВАНИЕ ЗАМКНУТЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Замкнутой называется система массового обслуживания, в которой имеется фиксированное количество заявок. Каждая из этих заявок поступает на обслуживание, обслуживается и через некоторое время снова требует обслуживания. Примеры замкнутых СМО: компьютер, обрабатывающий сигналы от нескольких датчиков (причем очередной сигнал с каждого датчика поступает на компьютер через некоторое время после получения ответа на предыдущий сигнал); наладчик, обслуживающий группу станков, которые периодически требуют наладки и т.д.

Каких-либо *специальных* операторов для моделирования замкнутых систем в языке GPSS *нет*. В GPSS-моделях замкнутых СМО каждая заявка обычно имитируется транзактом, который остается в модели в течение всего времени моделирования, т.е. его не требуется удалять оператором `TERMINATE`. Во многих случаях представляет интерес время пребывания транзакта в модели с момента начала очередного обслуживания.

Пример 7.1. В состав производственного участка входят шесть автоматических станков, работающих под управлением компьютера. Периодически станки направляют на управляющий компьютер сигналы о своем состоянии. Компьютер обрабатывает эти сигналы и направляет на станки управляющие сигналы. Время обработки компьютером сигнала о состоянии, полученного от станка - от 2 до 4 мс. Интервал между получением станком управляющего сигнала и выдачей следующего сигнала о состоянии - случайная величина, распределенная по экспоненциальному закону, со средним значением 10 мс. Требуется, чтобы время реакции компьютера (т.е. время от передачи станком сигнала о состоянии до получения управляющего сигнала) не превышало 15 мс; превышение этого ограничения допускается не более чем в одном случае из тысячи.

Требуется разработать GPSS-модель для анализа работы производственного участка в течение одной минуты.

```

        GENERATE      ,, , 6
STANOK ADVANCE      (EXPONENTIAL(1,0,10))
MARK
QUEUE      OCH_COMP
SEIZE      COMP
DEPART      OCH_COMP
ADVANCE      (UNIFORM(2,1,4))
RELEASE      COMP
SAVEVALUE      SIGNAL+,1
TEST G      M1,15,STANOK
SAVEVALUE      PREV+,1
TRANSFER      ,STANOK
;

        GENERATE      60000
SAVEVALUE      VER_PREV, (X$PREV/X$SIGNAL)
TERMINATE      1

```

Оператор `GENERATE ,, , 6` создает шесть транзактов, имитирующих сигналы, поступающие от станков к компьютеру и обратно. Оператор `MARK` устанавливает значение атрибута `M1` равным нулю; таким образом, имитируется отсчет времени с момента передачи сигнала о состоянии со станка на компьютер.

Группа операторов `QUEUE OCH_COMP... RELEASE COMP` имитирует работу управляющего компьютера. Таким образом, транзакт, выходящий из оператора `RELEASE COMP`, имитирует управляющий сигнал, передаваемый с компьютера на станок. Оператор `SAVEVALUE SIGNAL+,1` подсчитывает общее количество обработанных сигналов.

В операторе `TEST G M1,15,STANOK` проверяется, не превысило ли время реакции предельную величину (15 мс). Если условие `M1>15` выполняется, то транзакт поступает на оператор `SAVEVALUE PREV+,1`, выполняющий подсчет количества случаев, когда время реакции превысило допустимую величину. Затем транзакт передается (оператором `TRANSFER ,STANOK`) на метку `STANOK`, т.е. имитируется передача управляющего сигнала на станок. Если условие `M1>15` не выполняется (время реакции не превысило 15 мс), то транзакт также возвращается на метку `РАБОТА`.

Группа операторов

```

        GENERATE      60000
SAVEVALUE      VER_PREV, (X$PREV/X$SIGNAL)
TERMINATE      1

```

имитирует окончание моделирования по истечении одной минуты (60 000 мс). Как показано выше (см. примеры 2.2, 4.1), эти операторы выполняются за время моделирования один раз (конечно, если модель запускается на выполнение командой **START 1**). В операторе `SAVEVALUE VER_PREV, (X$PREV/X$SIGNAL)` вычисляется вероятность превышения допустимого времени реакции.

В данном примере сохраняемая величина `VER_PREV` приняла значение 0,004. Таким образом, вероятность превышения предельного времени реакции оказалась выше допустимой. Возможно, следует уменьшить количество стан-

ков, работающих под управлением одного компьютера. Чтобы выяснить это, достаточно повторить моделирование, заменив первый оператор модели на GENERATE ,,5 (имитация участка с пятью станками). В этом случае величина VER_PREV принимает значение, близкое к нулю. Таким образом, вариант участка с пятью станками является допустимым.

Пример 7.2. В цехе имеется шесть станков типа А и четыре станка типа В. Для станков периодически требуется наладка. Интервалы времени между наладками (т.е. интервалы от окончания наладки до момента, когда потребуется следующая наладка) представляют собой случайные величины, распределенные по экспоненциальному закону. Для станков типа А интервалы между наладками составляют в среднем 1 ч, для станков типа В – в среднем 2 ч. Время наладки станка – случайная величина, также распределенная по экспоненциальному закону; наладка станка А занимает в среднем 10 мин, станка В – в среднем 20 мин. В цехе работают два наладчика.

Прибыль от одного часа работы станка составляет 12 ден.ед.

Разработать модель работы цеха в течение 30 суток. Предусмотреть подсчет прибыли от работы цеха.

```

STANA      STORAGE      6
STANB      STORAGE      4
NALADCHIK  STORAGE      2
            GENERATE     ,, , 6
            ASSIGN       1, 1
RABOTA1    ENTER        STANA
            ADVANCE      (EXPONENTIAL(1, 0, 60))
            LEAVE        STANA
            TRANSFER     , NALADKA
            GENERATE     ,, , 4
            ASSIGN       1, 2
RABOTA2    ENTER        STANB
            ADVANCE      (EXPONENTIAL(1, 0, 120))
            LEAVE        STANB
NALADKA    SAVEVALUE    VREMRAB+, M1
            QUEUE        OCH_NALAD
            ENTER        NALADCHIK
            DEPART       OCH_NALAD
            TEST E       P1, 1, TIPB
            ADVANCE      (EXPONENTIAL(2, 0, 10))
            TRANSFER     , KONOBR
TIPB       ADVANCE      (EXPONENTIAL(2, 0, 20))
KONOBR     LEAVE        NALADCHIK
            MARK
            TEST E       P1, 1, RABOTA2
            TRANSFER     , RABOTA1
            GENERATE     43200
            SAVEVALUE    PRIB+, ((X$VREMRAB/60)#12)
            TERMINATE    1

```

Здесь многоканальные устройства STANA и STANB имитируют, соответственно, станки А и В. Многоканальное устройство NALADCHIK имитирует наладчиков. Оператор GENERATE „,6 создает шесть транзактов, используемых для имитации работы станков типа А. В операторе ASSIGN 1,1 в первый параметр этих транзактов записывается значение 1; это необходимо, чтобы имитировать время наладки станков с учетом их типа. Операторы

```
RABOTA1  ENTER      STANA
          ADVANCE    (EXPONENTIAL(1,0,60))
          LEAVE      STANA
```

имитируют работу станков типа А в то время, когда наладка для них не требуется. Оператор TRANSFER „NALADKA имитирует *остановку* станка для наладки.

Аналогично операторы GENERATE „,4...LEAVE STANB имитируют работу станков типа В.

В операторе SAVEVALUE VREMRAB+,M1 значение атрибута M1, т.е. время *от начала работы станка до его остановки для наладки*, прибавляется к сохраняемой величине VREMRAB. Таким образом, в этой сохраняемой величине суммируется время *работы* всех станков, т.е. то время, когда они именно *работают*, а не простаивают в ожидании наладки или во время самой наладки.

Операторы QUEUE OCH_NALAD ... KONOBR LEAVE NALADCHIK имитируют работу наладчиков.

В операторе MARK значение СЧА M1 становится равным нулю. Таким образом, начинается отсчет очередного периода работы станка по окончании его наладки.

Операторы TEST E P1,1,RABOTA2 и TRANSFER „RABOTA1 возвращают транзакты на операторы, имитирующие соответствующую группу станков. Таким образом имитируется возобновление работы станков по окончании наладки.

Группа операторов GENERATE 43200...TERMINATE 1 имитирует окончание моделирования по истечении 30 дней (43 200 минут). В операторе SAVEVALUE PRIB+,(X\$VREMRAB/60)#12) вычисляется прибыль от работы станков за 30 дней: время работы станков (сохраняемая величина VREMRAB) переводится в часы путем деления на 60, и полученная величина умножается на 12 (т.е. на прибыль от работы одного станка в течение часа).

Результаты моделирования оказались следующими:

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
UCH_NALAD	6	2	4542	2497	0.566	5.381	11.951	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STANA	6	2	0	6	3397	1	4.745	0.791	0	0
STANB	4	2	0	4	1151	1	3.329	0.832	0	0
NALADCHIK	2	0	0	2	4540	1	1.360	0.680	0	2

SAVEVALUE	RETRY	VALUE
VREMRAB	0	348082.786
PRIB	0	69616.557

Таким образом, за 30 суток станки работали примерно 348 083 минуты, или около 5800 часов. Прибыль составила примерно 69 617 ден.ед.

Легко подсчитать, что максимально возможное время работы за 30 суток (если бы станкам вообще не требовалась наладка) составляет для каждого станка 43 200 мин, а для всех 10 станков – 432 000 мин. Таким образом, простой станков за 30 суток, связанные с их наладкой, составили $432\,000 - 348\,083 = 83\,917$ мин, или примерно 1398,6 часа. Коэффициент загрузки *всех* станков составил $348\,083 / 432\,000 \approx 0,81$.

Коэффициент загрузки станков типа А составил 0,791, станков типа В – 0,832. Таким образом, простой, связанные с наладкой, составили для станков типа А 20,9% рабочего времени, для станков типа В – 16,8%.

Среднее время ожидания наладки (время от остановки станка до начала его наладки) составило 5,38 мин, т.е. достаточно немного.

Коэффициент загрузки наладчиков составил 0,68.

Таким образом, по всем основным показателям режим работы производственного участка можно считать приемлемым.

8. ОТОБРАЖЕНИЕ ИНФОРМАЦИИ В GPSS-МОДЕЛЯХ В ТАБЛИЧНОЙ И ГРАФИЧЕСКОЙ ФОРМЕ

8.1. Сбор данных о распределении значений характеристик модели. Таблицы

В системе моделирования GPSS World имеется возможность сбора данных о *распределении* значений различных числовых величин, используемых в модели (времени пребывания транзактов в модели, длин очередей и т.д.), по диапазонам. Эти данные выводятся в файл результатов моделирования в табличной форме: указывается *количество случаев* за время моделирования, когда заданная величина принимала значения из диапазонов $[B, B+C]$, $[B+C, B+2C]$, $[B+2C, B+3C]$ и т.д., где В и С - конкретные числа (границы и размеры диапазонов), задаваемые при объявлении таблицы. Более подробно структура таблицы рассматривается ниже.

Каждая таблица объявляется в разделе команд GPSS-модели (до первого оператора GENERATE) следующей командой:

```
имя_таблицы TABLE A,B,C,D
```

где имя_таблицы – произвольное имя, обычно символьное;

А - имя СЧА, распределение которого по диапазонам требуется показать в таблице. Этот СЧА называется также аргументом таблицы или табулируемой величиной;

В - верхняя граница первого диапазона аргумента таблицы;

С - шаг таблицы (разность между верхней и нижней границей каждого диапазона);

Д - количество диапазонов.

В качестве операнда А во многих случаях указывается М1, так как часто требуется получить данные о времени пребывания транзактов в модели. Значение операнда В следует выбирать близким к минимально возможному значению аргумента таблицы, а значение операнда D - близким к максимально возможному значению аргумента таблицы (как правило, эти величины в точности не известны). Значения операнда С выбирается в зависимости от того, с какой точностью требуется получить данные о распределении значений аргумента таблицы.

Как правило, заранее (до выполнения моделирования) не известно, какие значения будет принимать аргумент таблицы. Поэтому значения параметров таблицы (аргументы В,С,D в команде TABLE) сначала указываются приближенно, а затем подбираются путем выполнения нескольких запусков модели. Подбор параметров таблицы будет показан в примере 8.2.

Регистрация значения СЧА, заданного в операнде А при объявлении таблицы, производится при прохождении транзакта через следующий оператор:

```
TABULATE      А
```

где А - имя таблицы.

Оператор TABULATE указывается в тексте GPSS-модели в том месте, где требуется получить данные о значении табулируемой величины.

Пример 8.1. Для примера 7.2 требуется получить таблицу для анализа потерь времени работы станков, связанных с наладкой.

```
STANA      STORAGE      6
STANB      STORAGE      4
NALADCHIK  STORAGE      2
VREMNAL    TABLE      M1, 5, 15, 8
            GENERATE     , , , 6
...см. пример 7.2...
NALADKA    SAVEVALUE    VREMRAB+, M1
            MARK
            QUEUE        OCH_NALAD
            ENTER        NALADCHIK
            DEPART       OCH_NALAD
            TEST E       P1, 1, TIPB
            ADVANCE      (EXPONENTIAL(2, 0, 10))
            TRANSFER     , KONOBR
TIPB        ADVANCE      (EXPONENTIAL(2, 0, 20))
KONOBR      LEAVE        OPER
            TABULATE     VREMNAL
            MARK
            TEST E       P1, 1, RABOTA2
            TRANSFER     , RABOTA1
            GENERATE     43200
            SAVEVALUE    PRIB+, (X$VREMRAB/60) #12)
            TERMINATE    1
```

Здесь в команде VREMNAL TABLE M1,5,15,8 указаны параметры таблицы. Они означают, что требуется определить количество случаев, когда время наладки составляло менее 5 мин, от 5 до 20 мин, от 20 до 35 мин и т.д. Всего предполагается получить результаты по восьми интервалам с шагом 15 мин.

Оператор MARK, добавленный перед оператором QUEUE OCH_NALAD, требуется для того, чтобы отсчитывать время (значение атрибута M1) с момента начала ожидания наладки, т.е. с момента остановки станка.

Для регистрации значений табулируемой величины указан оператор TABULATE VREMNAL. Он указан после оператора LEAVE NALADCHIK, т.е. в той части модели, где имитируется окончание наладки (возобновление работы станка). В этот момент атрибут M1 содержит время пребывания транзакта в очереди к устройству NALADCHIK и в самом устройстве NALADCHIK, т.е. полное время простоя станка, включая время ожидания наладки и саму наладку.

Таблица, полученная по результатам моделирования, будет следующей.

TABLE	MEAN	STD.DEV.	RANGE		RETRY FREQUENCY	CUM. %
VREMNAL	18.303	17.281			0	
			–	5.000	915	20.16
			5.000	– 20.000	2108	66.62
			20.000	– 35.000	886	86.14
			35.000	– 50.000	376	94.42
			50.000	– 65.000	141	97.53
			65.000	– 80.000	66	98.99
			80.000	– 95.000	23	99.49
			95.000	–	23	100.00

Из таблицы видно, что среднее время остановки станка для наладки (величина MEAN) составило 18,303 мин. Стандартное отклонение (величина STD.DEV), т.е. среднее отклонение конкретных значений времени простоя от среднего времени простоя, составило 17,281 мин.

Из колонки FREQUENCY видно, что в 915 случаях время простоя станка составляло менее 5 мин. В 2108 случаях время простоя составило от 5 до 20 мин, в 886 случаях – от 20 до 35 мин, и т.д. В 23 случаях время простоя станка превысило 95 мин.

Из колонки CUM.% видно, что в 20,16% случаев время простоя станка составляло менее 5 мин, в 66,62% случаев – менее 20 мин (т.е. от 0 до 20 мин), в 86,14% случаев – менее 35 мин, и т.д. В 0,51% случаев (100-99,49) время простоя превышало 95 мин.

Анализ таблицы подтверждает вывод о хороших характеристиках работы участка. Видно, что длительные простои станков происходят достаточно редко.

Как отмечено выше, значения параметров таблицы (аргументы B,C,D в команде TABLE) требуется подбирать, выполняя несколько запусков модели. Если полученная таблица оказывается неудачной (например, все значения попадают в какой-либо один диапазон, выходят за границу первого или последнего диапазонов, или из таблицы нельзя получить какую-либо требуемую информа-

цию), то необходимо выполнить моделирование заново, изменив параметры таблицы в команде TABLE.

Пример 8.2. Пусть в условиях примера 8.1 желательно более точно определить *максимальное* время простоя станка, связанное с наладкой. Для этого следует увеличить количество диапазонов таблицы. Увеличив количество диапазонов до 13, т.е. используя команду VREMNAL TABLE M1,5,15,13, получим следующую таблицу:

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM. %
VREMNAL	18.303	17.281		0		
			– 5.000		915	20.16
			5.000 – 20.000		2108	66.62
			20.000 – 35.000		886	86.14
			35.000 – 50.000		376	94.42
			50.000 – 65.000		141	97.53
			65.000 – 80.000		66	98.99
			80.000 – 95.000		23	99.49
			95.000 – 110.000		11	99.74
			110.000 – 125.000		6	99.87
			125.000 – 140.000		4	99.96
			140.000 – 155.000		1	99.98
			155.000 – 170.000		1	100.00

Из этой таблицы видно, что время простоя станка ни разу не превышало 170 мин.

Построение гистограммы

Имеется возможность графического представления табулируемой величины в виде гистограммы. Для этого по окончании моделирования следует выбрать элемент меню Window – Simulation Window – Table Window. В появившемся окне Open Table Window выбрать таблицу, для которой требуется построить гистограмму (если в модели было построено несколько таблиц), и нажать кнопку ОК.

Гистограмма для примера 8.2 приведена на рис.8.1. На ней наглядно показано, что время простоя станка чаще всего составляло от 5 до 20 мин, реже – составляло от 0 до 5 мин или от 20 до 35 мин, еще реже – от 35 до 50 мин и т.д.

Примечание. Во многих случаях требуется поместить гистограмму в файл, подготовленный в текстовом редакторе Word, т.е. в файл с расширением .DOC. Это можно сделать, например, следующим образом:

1) получив гистограмму, нажать клавишу PrintScreen. Все содержимое экрана копируется в буфер;

2) вызвать графический редактор Paint, входящий в набор стандартных программных средств операционной системы Windows. Для этого нажать кнопку Пуск и выбрать команду Стандартные – Paint;

3) вставить содержимое буфера в окно графического редактора Paint (например, командой Правка – Вставить);

4) средствами графического редактора Paint выделить изображение гистограммы. Скопировать его в буфер (например, командой Правка – Копировать);

5) перейти в окно текстового редактора Word. Вставить содержимое буфера.

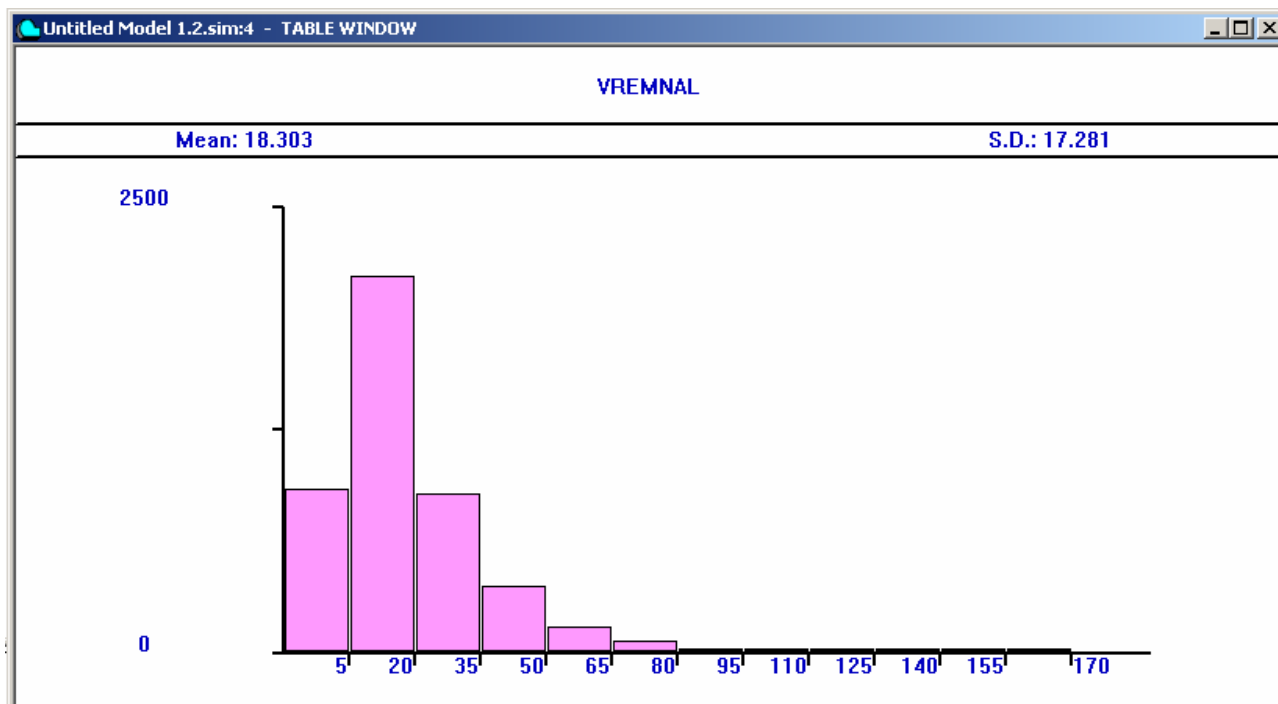


Рис.8.1. Гистограмма времени простоя станков для примера 8.2

8.2. Графическое отображение моделируемых величин во времени

В системе моделирования GPSS World имеется возможность графического отображения моделируемых величин (например, длин очередей, сохраняемых величин и т.д.) в процессе моделирования. Другими словами, в ходе моделирования можно наблюдать на экране графики изменения анализируемых величин во времени. Для этого требуется выполнить следующее:

- 1) подготовить текст модели и откомпилировать ее, используя команду меню **Command – Create Simulation**;
- 2) выбрать команду меню **Window – Simulation Window – Plot Window**;
- 3) в появившемся окне **Edit Plot Window** указать параметры графиков, которые требуется построить (эти действия будут подробно рассмотрены в примере 8.3);
- 4) запустить модель на выполнение, используя команду меню **Command – Start**. В процессе моделирования отображаются заданные графики.

Горизонтальная ось (абсцисса) всегда представляет собой ось времени, вертикальная ось (ордината) – ось моделируемой величины. В ходе моделирования может строиться один или несколько графиков. В одной системе координат (т.е. в одном окне) можно отобразить изменение одной или нескольких моделируемых величин.

Необходимо, однако, учитывать, что построение графиков в процессе моделирования значительно замедляет работу модели.

Пример 8.3. Воспользуемся возможностями графического отображения моделируемых величин для решения задачи анализа и устранения недостатков в работе ремонтной службы предприятия, рассмотренной в примере 5.1. Пусть

требуется в ходе моделирования отображать на экране изменение запаса деталей, потребности в деталях, а также длины очереди (т.е. количества приборов, ожидающих ремонта). Графики изменения запаса деталей и потребности в них следует построить *в одной системе координат*, чтобы выявить случаи, когда потребность в деталях превышала их запас. Таким образом, требуется построить два графика. На одном из них будет отображаться изменение запаса деталей и потребности в них, на другом – изменение длины очереди.

Запас деталей в модели имитируется сохраняемой величиной ZAPAS, длина очереди – стандартным числовым атрибутом Q очереди ОСН. Для отображения потребности в деталях введем еще одну сохраняемую величину. Обозначим ее как SPROS. При поступлении нового транзакта, имитирующего прибор, эта сохраняемая величина будет увеличиваться на величину первого параметра, т.е. на число деталей, необходимых для ремонта прибора. Для этого введем оператор SAVEVALUE SPROS+,P1. При поступлении транзакта в устройство REMONTNIK эта величина должна уменьшаться на величину параметра P1 (SAVEVALUE SPROS-,P1); таким образом имитируется получение необходимого количества деталей из запаса.

Приведем текст модели с внесенными изменениями

```

REMONTNİK    STORAGE      2
              INITIAL      X$ZAPAS,250
GOTOV        BVARIABLE    'NOT' (SF$REMONTNİK=1) 'AND' (X$ZAPAS>=P1)
              GENERATE      (EXPONENTIAL(1,0,30))
              ASSIGN        1,(DUNIFORM(2,3,7))
              SAVEVALUE      SPROS+,P1
              QUEUE          OCH
              TEST E        BV$GOTOV,1
              ENTER          REMONTNIK
              DEPART         OCH
              SAVEVALUE      ZAPAS-,P1
              SAVEVALUE      SPROS-,P1

```

... остальная часть модели – как в примере 5.1 ...

Подготовив текст модели, требуется откомпилировать ее, используя элемент меню **Command – Create Simulation**. Затем выбрать элемент меню **Window – Simulation Window – Plot Window**. В появившемся окне **Edit Plot Window** указать параметры графиков, которые требуется построить.

Введем сначала параметры графика, отображающего запас деталей и потребность в них.

В области **New Expression** в поле **Expression** требуется указать величину, для которой строится график: X\$ZAPAS (т.е. сохраняемую величину ZAPAS). В поле **Label** указывается произвольная подпись для графика указанной величины, например, “Запас деталей”. После заполнения полей **Expression** и **Label** требуется нажать кнопку **Plot**. Введенная величина и подпись отображаются в области **Window Contents**, т.е. в списке величин, для которых строятся графики. После этого можно также нажать кнопку **Memorize**, чтобы введенная вели-

чина была сохранена на время данного сеанса моделирования. Введенная величина и подпись отображаются в области **Memorized Expressions**.

Так как на этом же графике требуется отобразить изменение потребности в деталях (т.е. сохраняемой величины *SPROS*), необходимо в области **New Expression** в поле **Expression** указать величину *x\$SPROS*, а в поле **Label** - подпись к графику этой величины, например, “Потребность в деталях”. Нажать кнопку **Plot**, затем – кнопку **Memorize**. Введенная величина и подпись отображаются в областях **Window Contents** и **Memorized Expressions**.

Примечание. Если какая-либо величина введена в область **Window Contents** по ошибке, то для ее удаления следует выделить ее (в области **Window Contents**) и нажать кнопку **Remove**.

После ввода списка величин требуется указать остальные параметры графика. В поле **Title** вводится общий заголовок графика, например, “Запас и потребность в деталях”. В поле **Time Range** вводится диапазон оси времени. В данном случае время моделирования составляет 14 400 минут, поэтому следует ввести 14 400. В полях **Min Value** и **Max Value** вводится минимальное и максимальное значение по оси ординат (т.е. диапазон значений величин, для которых строятся графики). В поле **Min Value** следует оставить значение 0, так как запас деталей и потребность в них могут снижаться до нуля, но не могут быть ниже. Максимальное значение запаса составляет 250 (согласно постановке задачи), а максимальная потребность в деталях заранее точно не известна. Поэтому в поле **Max Value** можно указать, например, 300.

Edit Plot Window

New Expression

Label: Plot

Expression: Memorize

Window Contents

Title: Time Range:

Min Value: Max Value:

Label	Expression
Запас деталей	x\$zapas
Потребность в деталях	x\$SPROS

Remove

Memorized Expressions

Label	Expression
Запас деталей	x\$zapas
Потребность в деталях	x\$SPROS

Plot Delete

OK Cancel

Рис.8.2. Параметры графиков изменения запаса деталей и потребности в деталях для примера 8.3

Примечание. Значения величин, для которых строятся графики, заранее точно неизвестны, поэтому значение поля **Max Value** может оказаться неудачным. В этом случае требуется построить графики заново, изменив значение этого поля.

Окончательный вид окна **Edit Plot Window** показан на рис.8.2. После ввода всех параметров графиков следует нажать **OK**.

На экране появляется окно графиков (**PLOTS**). Вначале оно пустое, так как графики будут строиться в процессе моделирования. Это окно можно свернуть, но нельзя закрывать.

Так как требуется построить еще один график (отражающий изменение длины очереди), необходимо снова выбрать элемент меню **Window – Simulation Window – Plot Window**. В появившемся окне **Edit Plot Window** указать параметры графика. В поле **Expression** указать $Q_{\text{оч}}^{\text{оч}}$, в поле **Label** – произвольную подпись, например, “Количество приборов в очереди”. Нажать кнопку **Plot**, затем кнопку **Memorize**.

В поле **Title** указать заголовок графика, например, “Очередь на ремонт”. В поле **Time Range** указать 14 400. При заполнении полей **Min Value** и **Max Value** необходимо учесть, что количество приборов в очереди может достигать 20 (см. результаты моделирования в примере 5.1). Поэтому в поле **Max Value** можно ввести, например, 25. В поле **Min Value** оставить значение 0.

После ввода всех параметров нажать **OK**. На экране появляется еще одно пустое окно графиков.

Окна графиков требуется разместить на экране так, чтобы было удобно наблюдать построение графиков. После этого следует запустить модель на выполнение, выбрав элемент меню **Command – Start**.

Графики, построенные в ходе моделирования, приведены на рис.8.3 и 8.4.

Примечания:

1. Если к концу моделирования на экране отображается не весь график, следует воспользоваться полосами прокрутки, имеющимися у окна графика.

2. В некоторых случаях часть графика теряется (не выводится на экран). В этом случае следует увеличить количество точек графика, сохраняемых в памяти. Для этого требуется выбрать элемент меню **Edit – Settings**. В появившемся окне настройки (**Settings**) перейти на вкладку **Reports**. Увеличить количество сохраняемых точек графика, указанное в поле **Saved Plot Points** (значение по умолчанию – 10 000).

Из рис.8.3 видно, что к концу дня, как правило, потребность в деталях превышает их запас (соответствующие графики пересекаются). Имеются случаи, когда в течение некоторого времени запас деталей полностью отсутствует (равен нулю). Из-за этого к концу дня количество деталей, ожидающих ремонта, обычно увеличивается (см. рис.8.4).

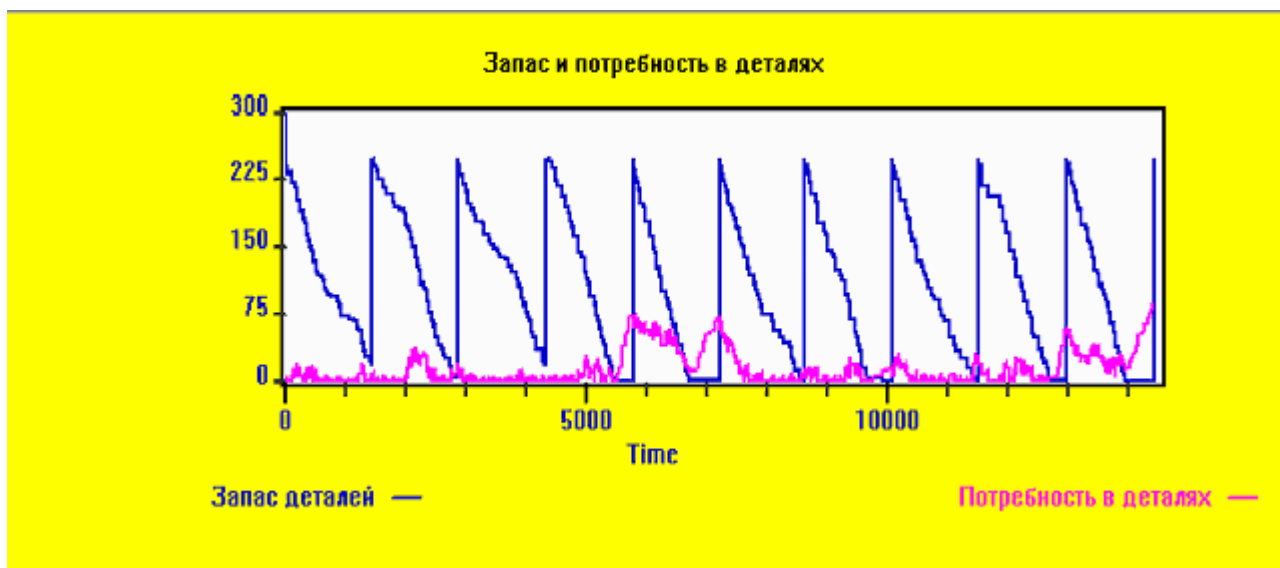


Рис.8.3. График изменения запаса деталей и потребности в деталях для примера 8.3

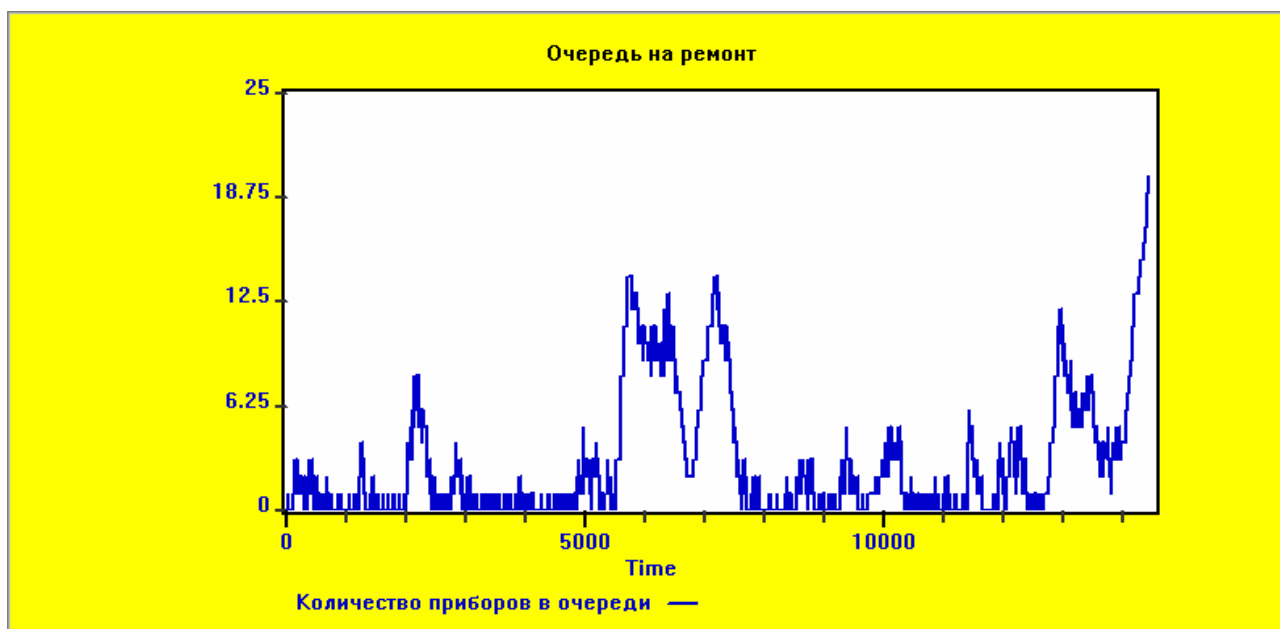


Рис.8.4. График изменения длины очереди для примера 8.3

Для устранения обнаруженного недостатка следует попытаться увеличить запас деталей. Пусть через каждые 24 часа запас деталей доводится до 300 штук. Предположим также, что начальный запас составляет 300 штук. Для имитации этих изменений в модели требуется заменить команду `INITIAL X$ZAPAS,250` на `INITIAL X$ZAPAS,300`, а команду `SAVEVALUE ZAPAS,250` на `SAVEVALUE ZAPAS,300`. После внесения изменений потребуется откомпилировать модель, а затем заново ввести параметры графиков, как показано для первоначального варианта модели.

Ниже приведены результаты моделирования и построенные графики.

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY		
OCH	8	3	512	226	0.696	19.578	35.049	0		
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
REMONTNIK	2	0	0	2	509	1	1.398	0.699	3	0

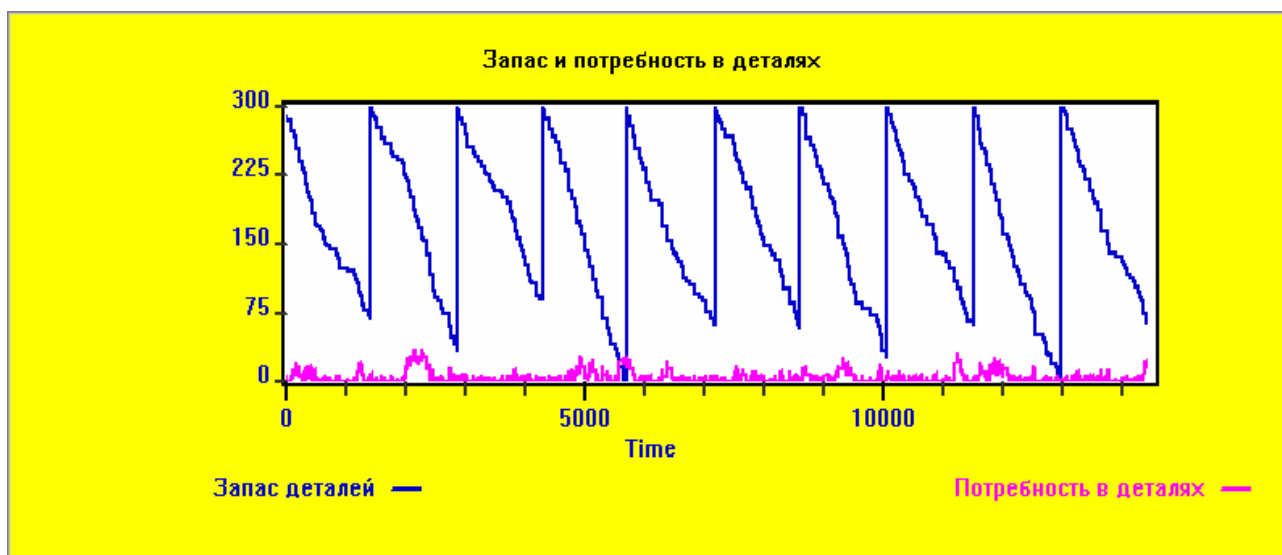


Рис.8.5. График изменения запаса деталей и потребности в деталях для примера 8.3 (модифицированный вариант)



Рис.8.6. График изменения длины очереди для примера 8.3 (модифицированный вариант)

Из результатов моделирования, а также из графиков (рис.8.5, 8.6) следует, что показатели работы ремонтной службы значительно улучшились. Среднее время ожидания ремонта снизилось с 78,635 мин до 19,578 мин, средняя длина очереди – с 2,79 до 0,696 прибора, максимальная длина очереди – с 20 до 8 приборов. Из графика на рис.8.5 видно, что нехватка деталей для ремонта возникла только один раз. Коэффициент загрузки ремонтной службы (0,699, или примерно 70%) также является приемлемым. Таким образом, предложенное изменение (пополнение запаса деталей до 300 штук через каждые 24 часа) следует признать рациональным.

Для нового варианта рассчитаем среднее время пребывания прибора в ремонтной службе, включающее время ожидания ремонта и время самого ремонта. Среднее время ожидания ремонта составляет 19,578 мин, или около 20 мин. Осмотр прибора занимает от 10 до 20 мин, т.е. в среднем 15 мин. В каждом

приборе требуется заменять от трех до семи деталей, т.е. в среднем пять деталей. Замена одной детали занимает в среднем пять минут; значит, замена деталей в приборе занимает в среднем $5 \cdot 5 = 25$ мин. Таким образом, среднее время с момента поступления прибора в ремонтную службу до окончания его ремонта составляет примерно $20 + 15 + 25 = 60$ мин, т.е. около одного часа. Для более точного определения среднего времени пребывания прибора в ремонтной службе следовало бы воспользоваться таблицей, как показано в подразд.8.1.

9. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD

В данном разделе рассматриваются некоторые возможности системы GPSS World, необходимые для решения сложных задач моделирования.

9.1. Дополнительные возможности управления перемещением транзактов

Для управления перемещением транзактов в модели часто применяется оператор TRANSFER. Выше были рассмотрены два режима использования этого оператора: режим безусловного перехода и статистический режим. Рассмотрим еще два режима работы этого оператора.

Режим BOTH (передача транзакта на один из двух операторов):

TRANSFER BOTH,A,B

где A,B – метки операторов.

Из оператора TRANSFER с операндом BOTH транзакт направляется на оператор с меткой A. Если переход на этот оператор оказывается невозможным (например, это оператор входа в устройство, которое оказалось занятым другим транзактом), то предпринимается попытка перехода на оператор с меткой B. Если и он оказывается занятым, то повторяется попытка перехода на оператор с меткой A. Попытки повторяются, пока транзакт не перейдет на один из операторов (с меткой A или B).

Если оператор записан в форме TRANSFER BOTH,,B, то сначала транзакт направляется на оператор, следующий за оператором TRANSFER. Если он оказывается занятым, то выполняется переход на оператор с меткой B; если и он занят – то снова на оператор, следующий за TRANSFER, и т.д.

Пример 9.1. Из узла связи UC1 на узел связи UC2 передаются сообщения. Поток сообщений можно считать пуассоновским; средний интервал времени между сообщениями - 5 мс. Узлы связаны двумя каналами связи (K1 и K2). Сначала предпринимается попытка передачи сообщения по каналу K1. Если он оказывается занятым, то делается попытка передачи по каналу K2. Если и он занят, то повторяется попытка передачи по каналу K1, и т.д. Передача сообщения по каналу K1 занимает от 5 до 10 мс, по каналу K2 – от 12 до 16 мс. Пока

сообщение не передано, оно находится в буфере (буфер – общий для обоих каналов).

Требуется разработать GPSS-модель для анализа передачи сообщений в течение одной минуты.

```

      GENERATE (EXPONENTIAL(1,0,5))
      QUEUE    BUFER
      TRANSFER BOTH,KAN1,KAN2
KAN1  SEIZE    1
      DEPART   BUFER
      ADVANCE  7.5,2.5
      RELEASE  1
      TERMINATE
KAN2  SEIZE    2
      DEPART   BUFER
      ADVANCE  14,2
      RELEASE  2
      TERMINATE
      GENERATE 60000
      TERMINATE 1
```

В данном примере можно было использовать оператор TRANSFER в следующей форме: TRANSFER BOTH,,KAN2. Метка KAN1 в этом случае не требуется.

Примечание. В данном примере для устройств, имитирующих каналы, использованы числовые имена 1 и 2.

По результатам моделирования коэффициенты загрузки обоих каналов составляют примерно 0,97. Среднее время ожидания сообщений в буфере составляет 68,2 мс, а среднее количество сообщений, ожидающих передачи – 13,6. Максимальное количество сообщений в буфере достигает 60. Таким образом, система передачи сообщений явно перегружена. Один из способов устранения этого недостатка рассматривается в примере 9.2.

Режим ALL (передача транзакта на один из нескольких операторов):

```
TRANSFER ALL,A,B,C
```

где A,B – метки операторов;

C – шаг.

Из оператора TRANSFER с операндом ALL транзакт направляется на оператор с меткой A. Если этот оператор занят, то предпринимается попытка перехода на оператор, расположенный на C строк ниже оператора с меткой A. Если и он занят, то делается попытка перехода еще на C строк ниже, и т.д., до оператора с меткой B. Если и он оказывается занятым, то повторяется попытка перехода на оператор с меткой A. Попытки повторяются, пока транзакт не перейдет на один из операторов.

Пример 9.2. В условиях примера 9.1 имеется возможность использовать еще один канал (K3). Попытка передачи по каналу K3 предпринимается, если

каналы K1 и K2 оказываются занятыми. Если канал K3 также занят, то попытки передачи повторяются, начиная с канала K1. Передача сообщения по каналу K3 занимает от 15 до 20 мс.

```

        GENERATE    (EXPONENTIAL(1,0,5))
        QUEUE       BUFER
        TRANSFER    ALL,KAN1,KAN3,5
KAN1    SEIZE       1
        DEPART      BUFER
        ADVANCE     7.5,2.5
        ASSIGN      1,1
        TRANSFER    ,VYHOD
        SEIZE       2
        DEPART      BUFER
        ADVANCE     14,2
        ASSIGN      1,2
        TRANSFER    ,VYHOD
KAN3    SEIZE       3
        DEPART      BUFER
        ADVANCE     17.5,2.5
        ASSIGN      1,3
VYHOD   RELEASE    P1
        TERMINATE
        GENERATE    60000
        TERMINATE   1

```

Здесь оператор `TRANSFER ALL,KAN1,KAN3,5` сначала передает транзакт на оператор с меткой `KAN1`, т.е. предпринимается попытка входа в устройство 1. Если оно занято, то происходит переход *на пять строк ниже*, т.е. на оператор `SEIZE 2`. Если устройство 2 также занято, то происходит переход еще на пять строк ниже, на оператор `KAN3 SEIZE 3`. Если устройство 3 также занято, то повторяется попытка перехода на оператор с меткой `KAN3`, и т.д.

В данном примере операторы `ASSIGN` использованы для записи в первый параметр транзакта номера (числового имени) устройства, в котором он обрабатывался. Это позволяет реализовать выход из всех трех устройств с помощью одного оператора (`RELEASE P1`). Конечно, можно было описывать выход из каждого устройства (канала) отдельно, но это удлинило бы текст модели.

По результатам моделирования коэффициенты загрузки первого и второго каналов составляют примерно 0,77, третьего – 0,71. Среднее время ожидания сообщений в буфере составляет 4,2 мс, а среднее количество сообщений, ожидающих передачи – 0,83. Количество сообщений в буфере никогда не превышает 11. Таким образом, добавление канала связи позволило значительно улучшить характеристики системы передачи сообщений.

9.2. Имитация недоступного состояния устройств

Недоступным состоянием устройства в GPSS-модели называется состояние, при котором вход транзактов в устройство невозможен, хотя устройство *не*

занято транзактом. Перевод устройства в состояние недоступности может использоваться, например, для имитации сбоев и отказов в работе производственного оборудования, переналадки станков, выключений оборудования для профилактического обслуживания и т.д.

Для перевода *одноканального* устройства в состояние недоступности используется следующий оператор:

FUNAVAIL A,B,C,D

где А - имя устройства, переводимого в состояние недоступности;

В - режим обработки транзакта, находившегося в устройстве в тот момент, когда оно стало недоступным. Этот операнд может принимать следующие значения:

значение по умолчанию (операнд В не указан): обработка транзакта в устройстве прерывается, а после восстановления доступности устройства - возобновляется и доводится до конца;

СО: обработка транзакта в устройстве продолжается и после того, как оно стало недоступным, и доводится до конца;

RE: обработка транзакта прерывается, и транзакт не возвращается в устройство после того, как оно снова станет доступным;

С - метка оператора, на который передается транзакт, обработка которого была прервана;

Д - номер параметра транзакта, обработка которого была прервана. В этот параметр записывается время, оставшееся до конца обработки транзакта в устройстве в тот момент, когда устройство стало недоступным.

Примечания:

1. Обычно операнды С и D используются, если в операнде В использовано значение RE. При других значениях операнда В указывать операнды С и D обычно не требуется.

2. У оператора FUNAVAIL имеется еще несколько операндов. Они не рассматриваются в данном пособии, так как применяются достаточно редко.

Для восстановления доступности *одноканального* устройства используется следующий оператор:

FAVAIL А

где А - имя устройства, переводимого в состояние доступности.

Состояние устройства (доступность или недоступность) отражается стандартным числовым атрибутом FV, принимающим значение 1, если устройство доступно, и 0 – если недоступно.

Для перевода *многоканального* устройства в состояние недоступности используется оператор SUNAVAIL А, а для восстановления его доступности – оператор SAVAIL А, где А – имя многоканального устройства. При использовании оператора SUNAVAIL недоступными становятся все каналы многоканального устройства. Если в устройстве в это время находится транзакт, то его об-

работка прерывается, а после восстановления доступности устройства (оператором SAVAIL) – возобновляется и доводится до конца.

Если требуется имитировать недоступность отдельных каналов многоканального устройства или управлять обработкой транзактов, оказавшихся в многоканальном устройстве в момент его перевода в состояние недоступности (например, направлять эти транзакты для продолжения обработки в другое устройство), то следует имитировать каждый канал как отдельное одноканальное устройство. Для имитации недоступного состояния каналов в этом случае следует использовать операторы FUNAVAIL и FAVAIL.

Пример 9.3. В цехе выполняется обработка деталей. Детали поступают на обработку ровно через каждые две минуты. Деталь, поступившая в цех, направляется на обработку на любой свободный станок; если в момент поступления детали оба станка оказываются занятыми, то деталь направляется на станок, освободившийся первым. Перед станками имеется общий накопитель для деталей, ожидающих обработки. Время обработки детали на станке – случайная величина, распределенная по гауссовскому закону со средним значением 3 мин и стандартным отклонением 30 с.

Время от времени в работе станков происходят сбои. Время работы станка без сбоев и время устранения сбоя – случайные величины, распределенные по экспоненциальному закону. Средняя длительность работы станка без сбоев – 2 часа. Устранение сбоя занимает в среднем 10 минут. Если сбой происходит во время обработки детали, то она бракуется.

Требуется разработать GPSS-модель для анализа работы цеха в течение 100 часов. Предусмотреть подсчет количества выпущенных деталей и количества деталей, забракованных из-за сбоев, а также подсчет времени простоев станков, связанных со сбоями.

```

GENERATE      2
QUEUE         NAKOP
TRANSFER     BOTH, , STAN2
SEIZE         STANOK1
DEPART        NAKOP
ADVANCE       (NORMAL (1, 3, 0.5) )
RELEASE       STANOK1
SAVEVALUE     VIPUSK+, 1
TRANSFER     , KON
;
STAN2         SEIZE         STANOK2
DEPART        NAKOP
ADVANCE       (NORMAL (2, 3, 0.5) )
RELEASE       STANOK2
SAVEVALUE     VIPUSK+, 1
TRANSFER     , KON
;
BRAK1         RELEASE       STANOK1
SAVEVALUE     ZABRAKOVANO+, 1
TRANSFER     , KON

```

```

BRAK2   RELEASE   STANOK2
        SAVEVALUE ZABRAKOVANO+,1
KON     TERMINATE
;
        GENERATE   (EXPONENTIAL(3,0,120))
        FUNAVAIL   STANOK1,RE,BRAK1
        ADVANCE    (EXPONENTIAL(3,0,10))
        SAVEVALUE  SBOI+,1
        SAVEVALUE  VREM_SBOI+,M1
        FAVAIL     STANOK1
        TERMINATE
;
        GENERATE   (EXPONENTIAL(3,0,120))
        FUNAVAIL   STANOK2,RE,BRAK2
        ADVANCE    (EXPONENTIAL(3,0,10))
        SAVEVALUE  SBOI+,1
        SAVEVALUE  VREM_SBOI+,M1
        FAVAIL     STANOK2
        TERMINATE
;
        GENERATE   6000
        TERMINATE  1

```

В данном примере два станка фактически представляют собой двухканальную СМО. Тем не менее, необходимо моделировать их как два отдельных одноканальных устройства, так как требуется имитировать сбои каждого из станков: очевидно, что сбои в работе станков происходят независимо друг от друга, а не на обоих станках одновременно.

Операторы GENERATE 2...TRANSFER ,KON имитируют поступление деталей, их обработку на станках и подсчет выпущенных (годных) деталей.

Оператор BRAK1 RELEASE STANOK1 имитирует удаление из первого станка деталей, забракованных при сбоях этого станка. Передача транзактов, имитирующих забракованные детали, на этот оператор выполняется с помощью оператора FUNAVAIL, рассматриваемого ниже.

Аналогичным образом оператор BRAK2 RELEASE STANOK2 имитирует удаление из второго станка деталей, забракованных в результате сбоев.

Операторы SAVEVALUE ZABRAKOVANO+,1 выполняют подсчет забракованных деталей.

Группа операторов

```

GENERATE   (EXPONENTIAL(3,0,120))
FUNAVAIL   STANOK1,RE,BRAK1
ADVANCE    (EXPONENTIAL(3,0,10))
SAVEVALUE  SBOI+,1
SAVEVALUE  VREM_SBOI+,M1
FAVAIL     STANOK1
TERMINATE

```

имитирует сбои на первом станке. Оператор GENERATE (EXPONENTIAL(3,0,120)) создает транзакты, имитирующие сбои. Оператор FUNAVAIL

STANOK1,RE,BRAK1 непосредственно имитирует сбой. Устройство STANOK1 переводится в состояние недоступности. Если при этом в нем находился транзакт, то он удаляется из устройства и не возвращается в него (операнд RE), а передается на оператор с меткой BRAK1 (т.е. на оператор RELEASE STANOK1). Оператор ADVANCE (EXPONENTIAL (3,0,10)) имитирует время, необходимое для устранения сбоя. Оператор SAVEVALUE SBOI+,1 предназначен для подсчета количества сбоев, а SAVEVALUE VREM_SBOI+,M1 - для подсчета потерь времени, связанных со сбоями. Оператор FAVAIL STANOK1 имитирует возобновление работы станка после устранения сбоя.

Аналогично имитируются сбои на втором станке.

Примечание. Как видно из примера, при использовании оператора FUNAVAIL с операндом RE требуется передавать транзакт, обработка которого была прервана, на оператор RELEASE (т.е. имитировать выход транзакта из устройства, переводимого в состояние недоступности). Другими словами, если в операнде B оператора FUNAVAIL указано значение RE, то в операнде C должна указываться метка оператора RELEASE. Это верно для рассматриваемой версии системы имитационного моделирования (GPSS World Student Version 4.3.5). В некоторых других версиях языка GPSS и систем моделирования, построенных на его основе, применения оператора FUNAVAIL с операндом RE достаточно, чтобы удалить транзакт из устройства и освободить это устройство. В этих версиях метка, указанная в операнде C оператора FUNAVAIL, не должна представлять собой метку оператора RELEASE.

Приведем результаты моделирования:

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
STANOK1	1510	0.752	2.986	1	3095	0	0	1	0
STANOK2	1489	0.730	2.942	1	3094	0	0	1	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
NAKOP	10	1	3000	2260	0.469	0.938	3.801	0

SAVEVALUE	RETRY	VALUE
VIPUSK	0	2927.000
SBOI	0	93.000
VREM_SBOI	0	914.065
ZABRAKOVANO	0	70.000

Примечание. Транзакты, переводящие устройство в состояние недоступности, не считаются входящими в это устройство. Они не учитываются при подсчете количества транзактов, входивших в устройство, и при определении среднего времени пребывания транзакта в устройстве (величины ENTRIES и AVE.TIME в выходной статистике).

По результатам моделирования проанализируем использование рабочего времени. Общее анализируемое время составляет 100 часов (для двух станков – 200 часов, или 12 000 минут). Используя полученные коэффициенты загрузки, найдем время работы каждого из станков: $6000 \cdot 0,752 = 4512$ мин (первый станок), $6000 \cdot 0,73 = 4380$ мин (второй станок). Таким образом, время работы двух станков составило $4512 + 4380 = 8892$ мин, а средний коэффициент загрузки двух станков – $8892 / 12\,000 \approx 0,74$.

За 100 часов на устранение сбоев на двух станках было затрачено примерно 914 минут (сохраняемая величина VREM_SBOI). Это составляет $914/12\ 000 = 0,08$, или 8% от общего времени.

Простои станка из-за отсутствия деталей для обработки составили $12\ 000 - 8892 - 914 = 2194$ мин. Это составляет $2194/12\ 000 = 0,18$, или 18% от общего времени.

Таким образом, для обработки деталей используется 74% рабочего времени (потери составляют 8% из-за сбоев и 18% из-за отсутствия деталей). Доля брака, вызванного сбоями, составляет $70/(2927+70) \approx 0,02$, или около 2%. Такие показатели можно считать приемлемыми.

Пример 9.4. Для решения некоторых задач используются два компьютера: основной и резервный. Поток задач, поступающих на решение, можно считать пуассоновским; средний интервал между моментами поступления задач – 5 минут. Решение задачи на любом из компьютеров занимает от 2 до 5 минут.

Резервный компьютер используется для решения задач только при сбоях основного компьютера. Время работы основного компьютера без сбоев – экспоненциальная случайная величина; в среднем это время составляет 5 часов. Время, необходимое для устранения сбоя – также экспоненциальная случайная величина; в среднем устранение сбоя занимает 10 минут. Резервный компьютер работает без сбоев.

Если сбой происходит во время решения задачи, то она доводится до конца на резервном компьютере в первую очередь. Задача, направленная для решения на резервный компьютер, решается на нем до конца в любом случае, даже если за это время устраняется сбой основного компьютера.

Разработать GPSS-модель для анализа работы вычислительного комплекса в течение 100 часов. Предусмотреть подсчет количества задач, решение которых прерывалось из-за сбоев.

```

                GENERATE   (EXPONENTIAL(1,0,5))
                ASSIGN     1, (UNIFORM(2,2,5))
VOZVRAT  QUEUE         OCH
                TRANSFER   BOTH, , REZERV
                SEIZE       OSN_COMP
                DEPART      OCH
                ADVANCE     P1
                RELEASE     OSN_COMP
                TRANSFER    , KON
REZERV   SEIZE         REZ_COMP
                DEPART     OCH
                ADVANCE    P1
                RELEASE    REZ_COMP
KON      TERMINATE
;
SBOI     RELEASE      OSN_COMP
                PRIORITY 1
                SAVEVALUE PRERVANO+, 1
                TRANSFER , VOZVRAT

```

```

;
    GENERATE    , , , 1
    FUNAVAIL    REZ_COMP
    TERMINATE

;
    GENERATE    (EXPONENTIAL(3,0,300))
    FUNAVAIL    OSN_COMP,RE,SBOI,1
    FAVAIL      REZ_COMP
    ADVANCE     (EXPONENTIAL(4,0,10))
    FAVAIL      OSN_COMP
    FUNAVAIL    REZ_COMP,CO
    TERMINATE

;
    GENERATE    6000
    TERMINATE   1

```

Оператор GENERATE (EXPONENTIAL(1,0,5)) имитирует поступление задач на решение. В операторе ASSIGN 1,(UNIFORM(2,2,5)) в первый параметр транзакта записывается время, необходимое для решения имитируемой задачи. Операторы QUEUE OCH...TRANSFER ,KON имитируют решение задач на основном или резервном компьютере.

Группа операторов

```

SBOI      RELEASE    OSN_COMP
          PRIORITY    1
          SAVEVALUE   POVT+,1
          TRANSFER     ,VOZVRAT

```

имитирует обработку задач, решение которых было прервано из-за сбоев. Транзакт, имитирующий прерванную задачу, выходит из устройства OSN_COMP; для этого используется оператор RELEASE OSN_COMP. В операторе PRIORITY 1 транзакту присваивается *приоритет*, равный единице, т.е. более высокий приоритет, чем у других транзактов (по умолчанию все транзакты имеют приоритет 0). Это требуется, чтобы имитировать решение прерванной задачи *в первую очередь*. Оператор SAVEVALUE PRERVANO+,1 имитирует подсчет задач, решение которых было прервано. Оператор TRANSFER ,VOZVRAT имитирует возврат задачи в очередь для продолжения решения.

Примечание. Оператор PRIORITY будет подробнее рассмотрен в подразд.9.3.

Группа операторов

```

GENERATE    , , , 1
FUNAVAIL    REZ_COMP
TERMINATE

```

переводит устройство REZ_COMP (имитирующее резервный компьютер) в состояние недоступности *в начале моделирования*. Это необходимо предусмотреть, так как резервный компьютер должен быть доступен только во время сбоев основного компьютера.

Группа операторов

```
GENERATE (EXPONENTIAL(3,0,300))
FUNAVAIL OSN_COMP,RE,SBOI,1
FAVAIL REZ_COMP
ADVANCE (EXPONENTIAL(4,0,10))
FAVAIL OSN_COMP
FUNAVAIL REZ_COMP,CO
TERMINATE
```

имитирует сбой в работе основного компьютера. Оператор FUNAVAIL OSN_COMP,RE,SBOI,1 переводит устройство OSN_COMP в состояние недоступности. Если в этот момент в устройстве находился транзакт, то он удаляется из устройства (для этого использован операнд RE) и направляется на метку SBOI. При этом время, оставшееся до конца обработки транзакта в устройстве OSN_COMP, записывается в его первый параметр. Оператор FAVAIL REZ_COMP переводит устройство REZ_COMP в состояние доступности. Оператор ADVANCE (EXPONENTIAL(4,0,10)) имитирует время, необходимое для устранения сбоя. Затем имитируется возобновление работы основного компьютера (FAVAIL OSN_COMP) и прекращение работы резервного компьютера (FUNAVAIL REZ_COMP,CO). При этом, если в устройстве REZ_COMP находится транзакт, то его обработка доводится до конца (для этого указан операнд CO). Таким образом имитируется доведение до конца задачи, решавшейся на резервном компьютере.

9.3. Имитация обслуживания заявок с приоритетами

Приоритеты заявок определяют порядок их обслуживания. Если в очереди находятся заявки с разными приоритетами, то первыми на обслуживание поступают заявки с более высоким приоритетом.

Приоритеты заявок могут быть относительными или абсолютными.

При обслуживании *с относительными приоритетами* заявка, поступившая на обслуживание, всегда обслуживается до конца, даже если в это время поступает заявка с более высоким приоритетом.

При обслуживании *с абсолютными приоритетами* обслуживание заявки прерывается, если поступает заявка с более высоким приоритетом. Заявка, обслуживание которой было прервано, возвращается в очередь и поступает на дообслуживание только тогда, когда в очереди не останется ни одной заявки с более высоким приоритетом.

В GPSS-моделях приоритеты транзактов, имитирующих заявки, задаются при их создании (с помощью операнда E в операторе GENERATE) или в ходе моделирования (с помощью оператора PRIORITY). По умолчанию приоритет транзактов равен нулю (самый низкий приоритет). Приоритет представляет собой один из СЧА транзакта, обозначаемый как PR. На него можно ссылаться в других операторах и командах GPSS (например, сравнивать с другими величинами).

нами в операторе TEST, использовать в качестве аргумента в команде FUNCTION и т.д.).

Если требуется имитировать обслуживание **с относительными приоритетами**, то достаточно *задать* приоритеты транзактов. Выбор транзактов из очереди с учетом их приоритетов производится автоматически.

Пример 9.5. Пусть в условиях примера 2.4 требуется в первую очередь обрабатывать детали типа 1. Для этого достаточно внести в текст модели следующие изменения:

```
STANOK STORAGE      2
      GENERATE      7.5,2.5,,,2
      ASSIGN        1,1
      TRANSFER      ,OBR
      GENERATE      10,2,,,1
      ASSIGN        1,2
```

...см. пример 2.4...

Этот же фрагмент модели можно составить по-другому:

```
STANOK STORAGE      2
      GENERATE      7.5,2.5
      ASSIGN        1,1
      PRIORITY      2
      TRANSFER      ,OBR
      GENERATE      10,2
      ASSIGN        1,2
      PRIORITY      1
```

...см. пример 2.4...

Для определения типа детали (и соответствующего времени обработки) вместо оператора TEST E P1,1,TIP2, использованного в примере 2.4, можно применить следующий оператор: TEST E PR,2,TIP2. Здесь приоритет транзакта (величина PR) сравнивается с числом 2. Если PR=2, значит, поступил транзакт, имитирующий деталь типа 1, иначе – деталь типа 2.

Если требуется имитировать обслуживание **с абсолютными приоритетами**, то необходимо применять специальные операторы.

Для имитации входа в устройство заявки, имеющей абсолютный приоритет (т.е. имеющей возможность прерывать обслуживание других заявок), используется следующий оператор:

```
PREEMPT      A,B,C,D,E
```

где А - имя устройства, в которое входит прерывающий транзакт;

В - операнд, устанавливающий, требуется ли учитывать приоритеты транзактов. Этот операнд может принимать следующие значения:

значение по умолчанию (операнд В не указан): приоритеты транзактов не учитываются. Транзакт, входящий в устройство через оператор PREEMPT, прерывает обслуживание транзакта, находящегося в этот

момент в устройстве, если этот транзакт вошел в устройство с помощью оператора SEIZE. Если в устройстве уже находится транзакт, вошедший в него с помощью оператора PREEMPT, то обслуживание такого транзакта не может быть прервано;

PR: транзакт, проходящий через оператор PREEMPT, прерывает обслуживание транзакта, находящегося в этот момент в устройстве, если приоритет транзакта, входящего через оператор PREEMPT, выше, чем приоритет транзакта, находящегося в устройстве;

С - метка оператора, на который передается транзакт, обработка которого была прервана;

D - номер параметра транзакта, обработка которого была прервана. В этот параметр записывается время, остававшееся до конца обработки транзакта в устройстве в момент прерывания его обслуживания;

Е - режим обработки транзакта, находившегося в устройстве в тот момент, когда в него вошел прерывающий транзакт. Этот операнд может принимать следующие значения:

значение по умолчанию (операнд Е не указан): обработка транзакта в устройстве прерывается, а после выхода прерывающего транзакта из устройства - возобновляется и доводится до конца;

RE: обработка транзакта прерывается, и транзакт не возвращается в устройство после того, как прерывающий транзакт выйдет из него.

Для имитации выхода прерывающего транзакта из устройства используется следующий оператор:

RETURN A

где А - имя устройства, из которого выходит транзакт.

Транзакты, входящие в устройство через оператор PREEMPT, учитываются в результатах моделирования точно так же, как и транзакты, входящие в устройство через оператор SEIZE.

Примечания:

1. Действие операндов С,D,E оператора PREEMPT аналогично действию операндов С,D,B (соответственно) оператора FUNAVAIL. Различие состоит в следующем: если в операнде Е оператора PREEMPT указано значение RE, то этого достаточно, чтобы удалить транзакт из устройства и освободить это устройство. Метка, указанная в операнде С оператора PREEMPT, *не должна* представлять собой метку оператора RELEASE или RETURN.

2. Операторы PREEMPT и RETURN действуют только на одноканальные устройства. Если требуется имитировать обслуживание с абсолютными приоритетами в системе с многоканальными устройствами, то необходимо моделировать каждый канал как отдельное одноканальное устройство.

Пример 9.6. В автоматизированной системе управления технологическим процессом (АСУТП) обрабатываются сигналы трех типов (сигналы А,В,С), поступающие от производственного оборудования. В среднем поступает 10 сигналов в секунду. Известно, что из всех сигналов примерно 15% составляют

сигналы типа А, 20% - сигналы типа В, 65% - сигналы типа С. Обработка одного сигнала типа А занимает в среднем 20 мс, сигнала типа В – 50 мс, сигнала типа С – 100 мс. Интервалы времени между сигналами и время обработки сигналов можно считать случайными величинами, распределенными по экспоненциальному закону.

Предлагаются три варианта дисциплины обслуживания сигналов: а) в порядке поступления (дисциплина FIFO); б) с относительными приоритетами; в) с абсолютными приоритетами. При обслуживании с приоритетами более высокий приоритет должны иметь сигналы, требующие меньшего времени обработки (поэтому высший приоритет будут иметь сигналы А, менее высокий – сигналы В, самый низкий – сигналы С).

Требуется выбрать дисциплину обслуживания, обеспечивающую минимальное среднее время реакции на все сигналы. Под временем реакции здесь понимается время от *поступления* сигнала до *окончания* его обработки, т.е. время ожидания обработки и самой обработки.

Для решения задачи воспользуемся GPSS-моделью для имитации работы АСУТП в течение десяти минут. Все интервалы времени в модели указаны в миллисекундах.

а) Дисциплина обслуживания FIFO

```

TIP      FUNCTION    RN1,D3
0.15,1/0.35,2/1,3
VREM_OBR FUNCTION    P1,D3
1,20/2,50/3,100
VREM_A   TABLE     M1,100,100,10
VREM_B   TABLE     M1,100,100,10
VREM_C   TABLE     M1,100,100,10
VREM_VSE TABLE     M1,100,100,10
          GENERATE    (EXPONENTIAL(1,0,100))
          ASSIGN      1,FN$TIP
          ASSIGN      2,FN$VREM_OBR
          QUEUE        OCH
          SEIZE        COMP
          DEPART       OCH
          ADVANCE      (EXPONENTIAL(2,0,P2))
          RELEASE      COMP
          TEST E       P1,1,TIP_B
          TABULATE     VREM_A
          TRANSFER     ,KON
TIP_B    TEST E       P1,2,TIP_C
          TABULATE     VREM_B
          TRANSFER     ,KON
TIP_C    TABULATE     VREM_C
KON      TABULATE     VREM_VSE
          TERMINATE
          GENERATE     600000
          TERMINATE    1

```

Все возможности языка GPSS, использованные в этой модели, рассмотрены ранее (например, в подразд.3.2). Таблицы VREM_A, VREM_B и VREM_C введены для получения информации о времени реакции АСУТП на сигналы А, В и С соответственно, а таблица VREM_VSE – для получения информации о времени реакции на все сигналы. Под временем реакции здесь понимается время от поступления сигнала до окончания его обработки, т.е. время ожидания обработки и самой обработки.

Значения среднего времени реакции на сигналы, полученные в результате моделирования, приведены в табл.9.1.

б) Обслуживание с относительными приоритетами

```
TIP          FUNCTION    RN1,D3
0.15,1/0.35,2/1,3
VREM_OBR     FUNCTION    P1,D3
1,20/2,50/3,100
VAZNOST      FUNCTION    P1,D3
1,3/2,2/3,1
VREM_A       TABLE      M1,100,100,10
VREM_B       TABLE      M1,100,100,10
VREM_C       TABLE      M1,100,100,10
VREM_VSE     TABLE      M1,100,100,10
              GENERATE    (EXPONENTIAL(1,0,100))
              ASSIGN      1,FN$TIP
              ASSIGN      2,FN$VREM_OBR
              PRIORITY     FN$VAZNOST
              QUEUE        OCH
```

... см. пример 9.6а)...

Здесь приоритет транзакта задается в операторе PRIORITY FN\$VAZNOST. Для определения приоритета в зависимости от значения первого параметра транзакта (т.е. от типа сигнала, имитируемого транзактом) введена дискретная функция VAZNOST. Максимальный приоритет (равный 3) получают транзакты, у которых первый параметр равен единице, т.е. имитирующие сигналы типа А.

Результаты моделирования приведены в табл.9.1. Как и следовало ожидать, по сравнению с дисциплиной обслуживания FIFO среднее время реакции АСУТП на сигналы с высоким приоритетом (типа А и В) уменьшилось, а на сигналы с низким приоритетом (С) – увеличилось. Среднее время реакции на сигналы уменьшилось.

в) Обслуживание с абсолютными приоритетами

```
TIP          FUNCTION    RN1,D3
0.15,1/0.35,2/1,3
VREM_OBR     FUNCTION    P1,D3
1,20/2,50/3,100
VAZNOST      FUNCTION    P1,D3
1,3/2,2/3,1
VREM_A       TABLE      M1,100,100,10
VREM_B       TABLE      M1,100,100,10
```



```

VREM_C      TABLE      M1,100,100,10
VREM_VSE    TABLE      M1,100,100,10
              GENERATE    (EXPONENTIAL(1,0,100))
              ASSIGN      1,FN$TIP
              ASSIGN      2,FN$VREM_OBR
              PRIORITY     FN$VAZNOST
              QUEUE        OCH
              PREEMPT      COMP,PR
              DEPART OCH
              ADVANCE      (EXPONENTIAL(2,0,P2))
              RETURN       COMP
              TEST E       P1,1,TIP_B

```

... см. пример 9.6a)...

Здесь для имитации поступления сигнала на обработку в компьютер (устройство COMP) использован оператор PREEMPT. Если в устройстве COMP уже имеется транзакт, но его приоритет ниже, чем у поступившего транзакта, то его обработка прерывается. Для этого в операторе PREEMPT использован операнд PR. Прерванная обработка транзакта возобновляется по окончании обработки транзакта с более высоким приоритетом.

Результаты моделирования приведены в табл.9.1.

Таблица 9.1

Дисциплина обслуживания	FIFO				С относительными приоритетами				С абсолютными приоритетами			
	А	В	С	Все	А	В	С	Все	А	В	С	Все
Время реакции АСУТП, мс	331,3	387,5	423,9	402,9	95,5	146,8	480,6	357,2	24,3	75,1	487,7	337,0

Таким образом, для того, чтобы среднее время реакции АСУТП на сигналы было минимальным, следует использовать дисциплину обслуживания с абсолютными приоритетами.

9.4. Матрицы

Матрицы в GPSS-модели представляют собой прямоугольные таблицы чисел. Они аналогичны двумерным массивам в обычных языках программирования.

Каждая матрица должна быть объявлена в разделе команд GPSS-модели следующей командой:

```
имя_матрицы  MATRIX      ,В,С
```

где имя_матрицы – произвольное имя (символьное или числовое);

В - количество строк матрицы;

С - количество столбцов матрицы.

Примечание. Запятая перед операндом В обязательна.

Для изменения значения элементов матрицы используется следующий оператор:

MSAVEVALUE A,B,C,D

где A - имя матрицы и (необязательно) знак + или -;

B,C - номер строки и столбца элемента матрицы, значение которого требуется изменить;

D - величина, на которую изменяется значение элемента матрицы.

Пример 9.7. В первом из приведенных ниже операторов MSAVEVALUE элементу матрицы ABC, расположенному в четвертой строке и втором столбце (будем обозначать такой элемент как ABC (4,2)), присваивается значение 10. Во втором из этих операторов элемент матрицы ABC (2,3) увеличивается на 2. В третьем операторе элемент матрицы ABC (2,5) уменьшается на 3.

```
MSAVEVALUE      ABC, 4, 2, 10
MSAVEVALUE      ABC+, 2, 3, 2
MSAVEVALUE      ABC-, 2, 5, 3
```

На значения элементов матрицы можно ссылаться в операторах GPSS-модели. Ссылка на элемент матрицы указывается следующим образом: MX\$имя_матрицы(номер_строки,номер_столбца).

Для элементов матриц могут задаваться начальные значения с помощью оператора INITIAL.

В стандартной настройке системы моделирования GPSS World вывод матриц в отчет *не предусмотрен*. Поэтому, прежде чем выполнять компиляцию модели, в которой используются матрицы, необходимо выбрать команду меню **Edit – Settings**, перейти на вкладку **Reports**, установить флажок **Matrices** и нажать кнопку **OK**.

Пример 9.8. Система передачи данных (СПД) передает сообщения от двух пунктов-источников (А и В) к трем пунктам-приемникам (С,D,E). Поток сообщений от обоих пунктов-источников - пуассоновские; средний интервал между сообщениями, поступающими из пункта А, составляет 10 мс, из пункта В - 25 мс. Известно, что из пункта А примерно 20% сообщений передается в пункт С, 60% - в пункт D, 20% - в пункт Е. Из пункта В примерно 60% сообщений передается в пункт С, 10% - в пункт D, 30% - в пункт Е. Передача одного сообщения занимает от 4 до 8 мс.

Требуется разработать GPSS-модель для анализа работы СПД в течение одной секунды. Предусмотреть подсчет сообщений, переданных из каждого пункта-источника в каждый из пунктов-приемников.

```
KOL      MATRIX      , 2, 3
ADR_A    FUNCTION    RN1, D3
0.2, 1/0.8, 2/1, 3
ADR_B    FUNCTION    RN1, D3
0.6, 1/0.7, 2/1, 3
          GENERATE    (EXPONENTIAL(1, 0, 10))
          ASSIGN      1, 1
          ASSIGN      2, FN$ADR_A
          TRANSFER     , OBR
```

```

OBR      GENERATE      (EXPONENTIAL(1,0,25))
          ASSIGN       1,2
          ASSIGN       2, FN$ADR_B
          QUEUE        OCH
          SEIZE        SPD
          DEPART       OCH
          ADVANCE      6,2
          RELEASE      SPD
          MSAVEVALUE   KOL+,P1,P2,1
          TERMINATE
          GENERATE     1000
          TERMINATE    1

```

Здесь в команде MATRIX объявлена матрица с именем KOL, состоящая из двух строк и трех столбцов. Элементы матрицы будут использоваться для подсчета переданных сообщений. В первой строке будут подсчитываться сообщения из пункта А, во второй – из пункта В. Например, элемент KOL(1,3) будет представлять собой количество сообщений, переданных из пункта А в пункт Е.

Операторы GENERATE (EXPONENTIAL(1,0,10))...ASSIGN 2, FN\$ADR_B имитируют поступление сообщений из пунктов-источников. При этом в первый параметр каждого транзакта записывается номер пункта-источника (1 - пункт А, 2 – пункт В). Во второй параметр записывается номер пункта-приемника (1 - пункт С, 2 - D, 3 - Е). Для имитации потоков сообщений от пункта А к пунктам-приемникам используется дискретная функция ADR_A, для имитации потоков сообщений от пункта В - функция ADR_B. Операторы QUEUE OCH...RELEASE SPD имитируют работу СПД. Оператор MSAVEVALUE KOL+,P1,P2,1 реализует подсчет переданных сообщений. При поступлении транзакта элемент матрицы KOL, задаваемый параметрами этого транзакта, увеличивается на единицу. Например, при поступлении транзакта, первый параметр которого равен 1, а второй – 3, увеличивается на единицу элемент матрицы KOL(1,3), т.е. элемент, расположенный в первой строке и третьем столбце. Таким образом регистрируется сообщение, переданное из пункта А в пункт Е.

Приведем матрицу KOL, полученную по результатам моделирования:

MATRIX	RETRY	INDICES	VALUE
KOL	0		
		1 1	24
		1 2	57
		1 3	14
		2 1	20
		2 2	1
		2 3	15

Под заголовком INDICES указываются номера строк и столбцов (первая колонка – номера строк, вторая – номера столбцов). Под заголовком VALUE указаны значения соответствующих элементов матрицы.

По результатам моделирования, за одну минуту из пункта А было передано 24 сообщения в пункт С, 57 – в D, 14 – в Е. Из пункта В было передано 20 сообщений в пункт С, одно – в пункт D, 15 – в пункт Е.

Пример 9.9. Цех выполняет сборку насосов. Заказы на сборку насосов могут быть внутренними (от других подразделений предприятия, в состав которого входит цех) и внешними (от других предприятий). В среднем в течение восьмичасового рабочего дня поступает 12 заказов на сборку насосов; поток заказов можно считать пуассоновским. Одновременно в цехе может выполняться работа по сборке восьми насосов. Характеристики заказов, поступающих в цех, приведены в табл.9.2.

Таблица 9.2

Тип насоса	Для водопроводной воды	Для промышленных стоков	Для нефтепродуктов
Доля в общем потоке заказов, %	30	45	25
Время сборки, часы	от 2 до 4	от 2 до 5	от 3 до 7
Прибыль цеха (для внутренних заказов), ден.ед.	100	130	200

Внешние заказы составляют 40% от всех заказов. Прибыль от выполнения внешнего заказа на 20% выше, чем прибыль от выполнения внутреннего заказа на выпуск насоса того же типа.

Требуется разработать GPSS-модель для анализа работы цеха в течение 100 рабочих дней. Предусмотреть подсчет прибыли цеха, а также количества выпущенных насосов каждого типа. При этом насосы, выпущенные по внутренним и по внешним заказам, должны подсчитываться отдельно.

Предварительно рассчитаем некоторые величины, которые потребуется указать в модели. Найдем средний интервал между моментами поступления заказов. Так как за восьмичасовой рабочий день поступает в среднем 12 заказов, значит, средний интервал между заказами составляет 0,75 часа. Найдем также значения прибыли предприятия от выпуска насосов по внешним заказам: при выпуске насоса для водопроводной воды эта прибыль составит 120 ден.ед., для промышленных стоков – 156 ден.ед., для нефтепродуктов – 240 ден.ед.

```

PRIBYL      MATRIX      , 3, 2
              INITIAL    MX$PRIBYL(1,1), 100
              INITIAL    MX$PRIBYL(1,2), 120
              INITIAL    MX$PRIBYL(2,1), 130
              INITIAL    MX$PRIBYL(2,2), 156
              INITIAL    MX$PRIBYL(3,1), 200
              INITIAL    MX$PRIBYL(3,2), 240
VIPUSK      MATRIX      , 3, 2
TIP_NASOSA  FUNCTION    RN1, D3
0.3,1/0.75,2/1,3
VID_ZAKAZA  FUNCTION    RN2, D2
0.6,1/1,2
CEH         STORAGE     8
              GENERATE   (EXPONENTIAL(1,0,0.75))
              ASSIGN     1, FN$TIP_NASOSA
              ASSIGN     2, FN$VID_ZAKAZA
              QUEUE      OCH

```

	ENTER	CEH
	DEPART	OCH
	TEST E	P1, 1, STOK
	ADVANCE	3, 1
	TRANSFER	, KON
STOK	TEST E	P1, 2, NEFT
	ADVANCE	3.5, 1.5
	TRANSFER	, KON
NEFT	ADVANCE	5, 2
KON	LEAVE	CEH
	SAVEVALUE	SUM_PRIB+, MX\$PRIBYL (P1, P2)
	MSAVEVALUE	VIPUSK+, P1, P2, 1
	TERMINATE	
	GENERATE	800
	TERMINATE	1

Здесь матрица `PRIBYL` содержит значения прибыли от выпуска насосов. Строки матрицы соответствуют типам насосов; в первом столбце матрицы указываются значения прибыли от выполнения внутренних заказов, во втором – от внешних заказов. Матрица `VIPUSK` предназначена для подсчета выпущенных насосов. В ходе моделирования в первом столбце матрицы будут подсчитываться насосы, выпущенные по внутренним заказам, во втором – по внешним.

Оператор `GENERATE (EXPONENTIAL(1,0,0.75))` имитирует поступление заказов. В операторе `ASSIGN 1, FN$TIP_NASOSA` в первый параметр транзакта записывается обозначение типа заказанного насоса (1 – для водопроводной воды, 2 – для промышленных стоков, 3 – для нефтепродуктов). В операторе `ASSIGN 2, FN$VID_ZAKAZA` во второй параметр транзакта записывается вид заказа (1 - внутренний заказ, 2 - внешний). Операторы `QUEUE OCH...LEAVE CEH` имитируют решение задач. В операторе `SAVEVALUE SUM_PRIB+, MX$PRIBYL (P1, P2)` выполняется подсчет выручки. При этом величина платы за решенную задачу определяется по матрице `PRIBYL` в зависимости от значений первого и второго параметров поступившего транзакта. Например, при поступлении транзакта, в первом параметре которого указано значение 1, а во втором - 2 (насос для промышленных стоков, внешний заказ), сохраняемая величина `SUM_PRIB` увеличивается на 120, т.е. на величину `PRIBYL(1,2)`. В операторе `MSAVEVALUE VIPUSK+, P1, P2, 1` выполняется подсчет выпущенных насосов (аналогично предыдущему примеру).

9.5. Цепи пользователя

Цепь (список) пользователя представляет собой набор транзактов, аналогичный очереди. Основное отличие цепи пользователя от очереди состоит в том, что транзакты могут входить в цепь пользователя и выходить из нее не только в порядке поступления, но и в любом другом порядке, заданном в GPSS-модели.

С каждой цепью пользователя связан *индикатор*, который может находиться в одном из состояний: “установлен” или “сброшен”. В начале имитации все индикаторы цепей пользователя сброшены.

Важное (но не единственное) применение цепей пользователя - имитация сложных дисциплин обслуживания в СМО: “последним пришел - первым обслужен” (LIFO), ограничение на время пребывания в очереди и т.д.

Для имитации входа транзакта в цепь пользователя применяется следующий оператор:

LINK A,B,C

где A - имя цепи пользователя;

B - порядок включения транзактов в список пользователя. Операнд B может принимать следующие значения:

FIFO: транзакт помещается в конец цепи пользователя;

LIFO: транзакт помещается в начало цепи пользователя;

PR: транзакты размещаются в цепи пользователя в порядке приоритетов;

Rномер_параметра: транзакты размещаются в цепи пользователя в порядке возрастания указанного параметра (чем меньше значение параметра, тем ближе располагается транзакт к началу цепи пользователя);

C - метка оператора, на который передается транзакт, если в момент его поступления в оператор LINK индикатор цепи пользователя оказывается сброшенным. В этом случае транзакт не включается в цепь пользователя. Индикатор цепи пользователя переводится в состояние “установлен”. Если операнд C не указан, то транзакт включается в цепь пользователя в любом случае.

Транзакт, вошедший в цепь пользователя, *не может* продвигаться дальше (например, в следующий оператор), пока он не будет выведен из цепи пользователя *другим* транзактом. Для вывода транзактов из цепи пользователя применяется следующий оператор:

UNLINK операция A,B,C,D,E,F

где операция - обозначение операции сравнения: L - меньше, LE - меньше или равно, E - равно, NE - не равно, G - больше, GE - больше или равно;

A - имя цепи пользователя, из которой выводятся транзакты;

B - метка оператора, на который передаются транзакты, выводимые из цепи пользователя;

C - количество транзактов, выводимых из цепи пользователя. Если этот операнд не указан или указано слово ALL, то из цепи пользователя выводятся все транзакты;

D – может принимать одно из следующих значений:

BACK: транзакты выводятся из цепи пользователя, начиная с ее конца (количество выводимых транзактов определяется операндом C);

номер параметра: для всех транзактов, *находящихся в цепи пользователя*, выполняется операция сравнения указанного параметра с величиной, указанной в операнде Е. Из цепи пользователя выводятся все транзакты, для которых выполняется условие, заданное операцией сравнения;

имя логической переменной: из цепи пользователя выводятся все транзакты, для которых выполняется условие, заданное логической переменной;

Е – величина, с которой сравнивается указанный в операнде D параметр транзактов, находящихся в цепи пользователя;

Е – метка оператора, на который передается транзакт, вошедший в оператор UNLINK, если не удастся вывести из цепи пользователя столько транзактов, сколько указано в операнде С (например, из-за того, что количество транзактов в цепи пользователя меньше указанного в операнде С), или не удастся вывести ни одного транзакта (например, из-за того, что условие, заданное операндами D и Е, не выполняется ни для одного транзакта).

Если после прохождения транзакта через оператор UNLINK оказывается, что в цепи пользователя нет транзактов, то индикатор цепи пользователя сбрасывается.

Примечание. Здесь рассмотрены не все возможности операторов LINK и UNLINK. Кроме того, приведенное описание работы операторов является упрощенным. Достаточно полное описание операторов LINK и UNLINK имеется в [1].

Пример 9.10. Рассмотрим следующий пример оператора UNLINK:

```
UNLINK LE   NAKOP,OBR,3,2,10,KON
```

При поступлении транзакта в этот оператор выполняются следующие действия. Из цепи пользователя NAKOP, начиная с ее начала, выводятся транзакты, у которых значение второго параметра не превышает 10. Количество транзактов, выводимых из цепи пользователя, не превышает трех. Если заданное условие ($P2 \leq 10$) выполняется более чем для трех транзактов, находящихся в цепи пользователя, то выводятся только три из них – те, что находятся ближе к началу цепи пользователя. Транзакты, *выведенные из цепи пользователя*, направляются на оператор с меткой OBR. Если из цепи 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.

Пример 9.11. В отделе информационного обеспечения некоторого предприятия 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); б) дисциплина LIFO (первым выполняется заказ, поступивший последним); в) обслуживание по минимальному времени (в первую очередь выполняются заказы, для выполнения которых требуется меньшее время).

Для выбора лучшей дисциплины обслуживания проведем 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))
	ASSIGN	1, (1.5#DUNIFORM(2, 1, 25))
	QUEUE	OCH
	SEIZE	XEROX
	DEPART	OCH
	ADVANCE	P1
	RELEASE	XEROX
	TABULATE	VREM
	TERMINATE	1

Здесь в операторе ASSIGN 1, (1.5#DUNIFORM(2, 1, 25)) вычисляется время, необходимое для выполнения заказа; эта величина присваивается первому параметру транзакта, имитирующего заказ. 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).

б) Дисциплина обслуживания LIFO

VREM	TABLE	M1, 20, 20, 12
	GENERATE	(EXPONENTIAL(1, 0, 30))
	ASSIGN	1, (1.5#DUNIFORM(2, 1, 25))
	QUEUE	OCH
	LINK	OZHIDANIE, LIFO, COPIR
COPIR	SEIZE	XEROX
	DEPART	OCH
	ADVANCE	P1
	RELEASE	XEROX
	UNLINK	OZHIDANIE, COPIR, 1
	TABULATE	VREM
	TERMINATE	1

Рассмотрим применение цепи пользователя в данной модели.

Первый транзакт, поступивший на оператор LINK, не входит в цепь пользователя (так как ее индикатор сначала сброшен), а направляется на метку COPIR. При этом индикатор цепи пользователя переходит в состояние “установлен”, поэтому очередной транзакт (если он поступит на оператор LINK до окончания обработки первого транзакта) уже войдет в цепь пользователя. Поступающие транзакты будут добавляться *в начало* цепи пользователя, так как в операторе LINK указан операнд LIFO. По окончании имитации копирования (операторы SEIZE XEROX...RELEASE XEROX) транзакт попадает на оператор UNLINK и выводит один транзакт из цепи пользователя, если он там есть. Транзакт, выведенный из цепи пользователя, направляется на метку COPIR (как указано в операторе UNLINK).

Если при прохождении транзакта через оператор UNLINK в цепи пользователя нет транзактов, то индикатор цепи пользователя сбрасывается. В этом случае следующий транзакт, поступивший на оператор LINK, будет сразу направлен на метку COPIR, не входя в цепь пользователя.

Таким образом, индикатор цепи пользователя *установлен*, когда *устройство*, доступ к которому имитируется с помощью цепи пользователя (в данном примере – устройство XEROX), *занято*, и *сброшен*, когда это *устройство* *свободно*.

Приведем фрагмент файла-отчета: данные об очереди (QUEUE), о цепи пользователя (USER CHAIN), а также часть таблицы VREM, введенной в модель для определения среднего времени выполнения заказов.

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
OCH	8	1	1001	381	0.824	25.367	40.955	0

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
VREM	44.495	64.994		0		

...таблица VREM...

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
OZHIDANIE	0	0	0.824	620	8	40.955

Раздел отчета USER CHAIN содержит следующую **информацию о цепи пользователя**:

- USER CHAIN - имя цепи пользователя;
- SIZE - длина цепи пользователя (количество транзактов в ней) в момент окончания моделирования;
- RETRY - внутренние данные о работе системы моделирования GPSS World (не рассматриваются в данном пособии);
- AVE.CONT.- средняя длина цепи пользователя;
- ENTRIES - количество транзактов, входивших в цепь пользователя;
- MAX - максимальная длина цепи пользователя за время моделирования;
- AVE.TIME - среднее время пребывания транзактов в цепи пользователя.

В данном случае среднее количество заказов, ожидавших выполнения, составило 0,824. Максимальное количество заказов, ожидавших выполнения – восемь. Среднее время ожидания выполнения заказа составило 40,955 мин; однако необходимо учитывать, что эта величина – среднее время ожидания *только* для тех заказов, которым потребовалось *ожидать* выполнения хотя бы некоторое время (так как транзакты, которым *не требовалось ждать*, не входили в цепь пользователя). Среднее время ожидания для *всех* заказов имеется в разделе данных об очередях (AVE.TIME) и составляет 25,367 мин.

Среднее время выполнения заказа составило 44,495 мин (эта величина указана в колонке MEAN таблицы VREM).

в) Обслуживание по минимальному времени

Для имитации такой дисциплины обслуживания требуется в модели, приведенной в примере б), изменить оператор LINK следующим образом: LINK OZHIDANIE,P1,COPR. В этом случае транзакт, поступивший в оператор LINK, включается в цепь пользователя в соответствии со значением своего первого параметра. Чем меньше эта величина, тем ближе оказывается транзакт к началу цепи пользователя.

Среднее время выполнения заказа в данном случае составило 37,391 мин (величина MEAN в таблице VREM). Это меньше, чем для дисциплин обслуживания FIFO и LIFO (см. выше).

Таким образом, минимальное среднее время выполнения всех заказов достигается, если выполнять в первую очередь те заказы, для которых требуется меньше времени.

Пример 9.12. 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 час \pm 10 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.

UPAK	STORAGE	2
	GENERATE	(EXPONENTIAL(1,0,5))
	LINK	PLS,FIFO
VYVOZ	ASSIGN	1,X\$VREMPER
	ADVANCE	P1
	SAVEVALUE	PEREVEZENO+,1

	QUEUE	OCH_UPAK
	ENTER	UPAK
	DEPART	OCH_UPAK
	ADVANCE	10,5
	LEAVE	UPAK
	SAVEVALUE	GOTOV+,1
	TERMINATE	
;		
	GENERATE	,,,1
PRIEZD	ADVANCE	60,10
	SAVEVALUE	VREMPER, (UNIFORM(2,10,15))
	SAVEVALUE	KOL_PEREVOZOK+,1
	TEST G	CH\$PLS,20,VSE
	UNLINK	PLS,VYVOZ,20
	TRANSFER	,PEREVOZKA
VSE	UNLINK	PLS,VYVOZ,ALL
PEREVOZKA	ADVANCE	X\$VREMPER
	TRANSFER	,PRIEZD
	GENERATE	1440
	TERMINATE	1

В операторе GENERATE (EXPONENTIAL(1,0,5)) имитируется поступление изделий на площадку. В операторе LINK PLS,FIFO транзакты, имитирующие изделия, входят в цепь пользователя PLS; она применяется для имитации пребывания деталей на площадке. После того, как транзакты *выходят* из цепи пользователя (см. ниже), в операторе ASSIGN 1,X\$VREMPER в первый параметр каждого транзакта записывается время перевозки изделий на тележке (определение этого времени также рассматривается ниже). В операторе ADVANCE P1 имитируется перевозка изделий на участок упаковки, выполняемая тележкой. Затем выполняется подсчет количества перевезенных изделий, имитация упаковки и подсчет количества упакованных изделий.

В операторе GENERATE,,,1 создается один транзакт, имитирующий транспортную тележку. Оператор ADVANCE 60,10 имитирует интервалы между приездами тележки в цех. В операторе SAVEVALUE VREMPER, (UNIFORM(2,10,15)) разыгрывается время, которое будет затрачено тележкой на перевозку изделий; это время записывается в сохраняемую величину VREMPER и затем используется для имитации перевозки. В операторе TEST G CH\$PLS,20,VSE проверяется количество изделий, ожидающих вывоза с площадки (количество транзактов в цепи пользователя PLS). Здесь CH – стандартный числовой атрибут цепи пользователя, указывающий текущее количество транзактов в ней. Если количество изделий на площадке превышает 20, то имитируется вывоз 20 изделий: 20 транзактов выводятся из цепи пользователя PLS (оператор UNLINK PLS,VYVOZ,20). Если количество изделий, ожидающих вывоза, меньше (или равно) 20, то имитируется вывоз всех изделий (оператор UNLINK PLS,VYVOZ,ALL).

Примечание. Используемый в данном примере способ имитации времени перевозки (его запись в сохраняемую величину, а затем – чтение из этой величины) требуется для того, чтобы время перевозки всех изделий, доставляемых тележкой на участок упаковки за одну перевозку, было одинаковым.

Приведем фрагмент результатов моделирования:

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY		
OCH_UPAK	22	10	267	13	7.860	42.394	44.563	0		
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
UPAK	2	0	0	2	257	1	1.738	0.869	0	10
USER CHAIN	SIZE	RETRY	AVE.CONT		ENTRIES	MAX	AVE.TIME			
PLS	9	0	7.244		276	24	37.795			
SAVEVALUE	RETRY		VALUE							
VREMPER	0		10.089							
KOL_PEREVOZOK	0		19.000							
PEREVEZENO	0		267.000							
GOTOV	0		255.000							

Из полученных результатов видно, что на площадке для временного хранения находилось в среднем 7,22 изделия. Максимальное количество изделий, скапливавшихся на площадке – 24. Среднее время пребывания изделия в ожидании вывоза составило примерно 37,8 мин. Всего было выполнено 19 перевозок и перевезено 267 изделий; таким образом, за одну перевозку тележка перевозила в среднем $267/19 \approx 14$ изделий (при вместимости 20 изделий). Возможно, следует увеличить интервалы между приездами тележки в цех; это позволит больше использовать ее на других работах. Другой вариант – использование для перевозки изделий тележки меньшей вместимости.

Пример 9.13. На формовочную машину, выпускающую пластмассовые изделия, поступают формы с расплавленной пластмассой. Обработка пластмассы, находящейся в форме, занимает от 2 до 5 мин. Поток форм с пластмассой можно считать пуассоновским; в среднем интервал между моментами поступления форм составляет 3 мин. Если форма с пластмассой ожидает обработки на формовочной машине более шести минут, то пластмасса, находящаяся в форме, затвердевает, и ее обработка становится невозможной (пластмасса бракуется).

Требуется разработать модель для анализа работы формовочной машины в течение восьми часов. Подсчитать количество выпущенных изделий и случаев брака.

	GENERATE	(EXPONENTIAL (1, 0, 3))
	QUEUE	OCH_FORM
	ASSIGN	1, (C1+6)
	LINK	OZHIDANIE, FIFO, OBR
OBR	SEIZE	FORM_MASH
	DEPART	OCH_FORM
	ADVANCE	3.5, 1.5
	SAVEVALUE	IZDELIA+, 1
	RELEASE	FORM_MASH

```

        UNLINK L   OZHIDANIE,BRAK,ALL,1,C1
        UNLINK     OZHIDANIE,OBR,1
        TERMINATE

;
BRAK      SAVEVALUE ZABRAKOVANO+,1
        DEPART     OCH_FORM
        TERMINATE

;
        GENERATE   480
        TERMINATE  1

```

Здесь в операторе ASSIGN 1, (C1+6) в первый параметр транзакта записывается *предельно допустимое* время выхода из очереди, определяемое как время поступления в очередь (C1) плюс шесть минут. С помощью оператора LINK OZHIDANIE,FIFO,OBR транзакт включается в цепь пользователя OZHIDANIE (или сразу направляется на метку OBR, если индикатор цепи пользователя сброшен, т.е. устройство FORM_MASH свободно). С помощью операторов SEIZE...RELEASE имитируется обработка пластмассы на формовочной машине.

Оператор UNLINK L OZHIDANIE,BRAK,ALL,1,C1 обеспечивает удаление из цепи пользователя всех транзактов, первый параметр которых меньше значения текущего времени (C1). Эти транзакты направляются на оператор с меткой BRAK. Таким образом имитируется удаление всех форм, у которых время ожидания превысило допустимую величину. Оператор UNLINK OZHIDANIE,OBR,1 выводит из цепи пользователя один транзакт и направляет его на метку OBR, т.е. на вход в устройство FORM_MASH.

9.6. Косвенная адресация

Под косвенной адресацией будем понимать ссылку на СЧА некоторого объекта модели, номер которого записан в одном из параметров транзакта. При использовании косвенной адресации имя объекта модели (очереди, устройства и т.д.) должно быть *числовым*. Ссылка на СЧА имеет следующий вид: имя_СЧА*номер_параметра.

Пример 9.14. На узел связи поступают сообщения для передачи. Поток сообщений можно считать пуассоновским; средний интервал времени между сообщениями - 15 мс.

Узел связи оснащен двумя комплектами аппаратуры передачи данных (АПД). Сообщение, поступившее на узел связи, направляется на свободный комплект АПД. Если оба комплекта заняты передачей сообщений, то поступившее сообщение направляется на тот комплект, где меньше сообщений, ожидающих передачи. Каждый комплект АПД оснащен буфером для сообщений, ожидающих передачи; емкость буфера - пять сообщений. Если в момент поступления сообщения оба буфера АПД заполнены, то сообщение не передается (получает отказ). Длительность передачи сообщения - случайная величина, рас-

пределенная по экспоненциальному закону. В среднем передача одного сообщения занимает 25 мс.

Требуется разработать GPSS-модель для анализа работы узла связи в течение одной минуты.

```

                GENERATE      (EXPONENTIAL (1,0,15))
                SELECT NU     1,1,2,,,SELQ
                TRANSFER      ,PEREDACHA
SELQ            SELECT MIN   1,1,2,,Q
                TEST LE      Q*1,4,ZAPOLN
PEREDACHA       QUEUE       P1
                SEIZE        P1
                DEPART       P1
                ADVANCE      (EXPONENTIAL (2,0,25))
                RELEASE      P1
                TRANSFER      ,KON
ZAPOLN          SAVEVALUE   OTKAZ+,1
KON             TERMINATE
                GENERATE     60000
                TERMINATE    1

```

Здесь оператор `GENERATE (EXPONENTIAL (1,0,15))` имитирует поступление сообщений. В операторе `SELECT NU 1,1,2,,,SELQ` из устройств с номерами 1 и 2 выбирается свободное устройство (АПД); номер свободного устройства записывается в первый параметр транзакта, имитирующего сообщение. Если свободного устройства нет (оба комплекта АПД заняты передачей сообщений), то транзакт направляется на оператор с меткой `SELQ`. Если свободное устройство найдено, то транзакт передается на следующий оператор (`TRANSFER ,PEREDACHA`), а из него - на оператор `QUEUE P1` для имитации передачи. В операторе `SELECT MIN 1,1,2,,Q` (с меткой `SELQ`) из очередей с номерами 1 и 2 выбирается очередь минимальной длины, т.е. буфер, содержащий минимальное количество сообщений. Номер этой очереди записывается в первый параметр транзакта.

В операторе `TEST LE Q*1,4,ZAPOLN` выполняется имитация отказов при заполнении обоих буферов. Здесь `Q` - имя СЧА (длина очереди), `1` - номер *параметра*, в котором содержится номер очереди, длину которой требуется проверить. Этот оператор означает следующее: если длина очереди, *номер которой записан в первом параметре транзакта*, меньше или равна четырем, то транзакт пропускается дальше (т.е. имитируется поступление сообщения в буфер); в противном случае он направляется на метку `ZAPOLN` (имитируется отказ).

Операторы `QUEUE P1...RELEASE P1` имитируют передачу сообщений. Здесь имитируется работа *двух* комплектов АПД (с номерами 1 и 2), а также *двух* очередей к ним. Номер очереди (буфера) и устройства (АПД), используемых транзактом, находится в первом параметре (`P1`) этого транзакта.

Примечание. Использование оператора `SELECT` рассмотрено в подразд.4.3.

9.7. Арифметические переменные

В языке GPSS имеется возможность описывать математические вычисления, выполняемые в модели, в виде арифметических переменных. Арифметическая переменная задается следующей командой:

имя_переменной VARIABLE выражение

где выражение – математическое выражение для вычисления переменной.

Примечание. Вместо команды VARIABLE может использоваться FVARIABLE. В системе моделирования GPSS World эти команды действуют одинаково. В других системах моделирования, также основанных на языке GPSS, действие этих команд может различаться.

Ссылка на значение переменной в GPSS-модели указывается в следующей форме: V\$имя_переменной, если имя переменной – символьное, и Vномер, если имя переменной – числовое.

Пример 9.15. В условиях примера 9.11 воспользуемся переменными для описания интервалов времени между заказами и затрат времени на выполнение копировальных работ.

```
VREM                    TABLE        M1,20,20,12
INTERVAL_ZAKAZOV      VARIABLE      EXPONENTIAL(1,0,30)
VREM_COPIR            VARIABLE      1.5#DUNIFORM(2,1,25)
;
                      GENERATE      V$INTERVAL_ZAKAZOV
                      ASSIGN        1,V$VREM_COPIR
                      QUEUE         OCH
```

...см. пример 9.11...

Использование арифметических переменных делает текст модели более удобным для понимания, отладки и внесения изменений.

ЛИТЕРАТУРА

Основная

1. Томашевский В.Н., Жданова Е.Г. Имитационное моделирование в среде GPSS. М.: Бестселлер, 2003. – 416 с.
2. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004. – 320 с.
3. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World. СПб.: БХВ-Петербург, 2004. – 360 с.
4. Рыжиков Ю.И. Имитационное моделирование. Теории и технологии. СПб.: Корона-принт, 2004. – 384 с.
5. Лоу А.М., Кельтон В.Д. Имитационное моделирование. Классика Computer science. СПб.: Изд. дом “Питер”, 2004. – 848 с.

Дополнительная

6. Таха Х. Введение в исследование операций. М.: Изд. дом “Вильямс”, 2001. – 912 с.
7. Харин Ю.С., Малюгин В.И., Кирлица В.П. и др. Основы имитационного и статистического моделирования. Мн.: Дизайн ПРО, 1997. – 288 с.
8. Кнут Д.Э. Искусство программирования. В 3 т. Т. 2. Получисленные алгоритмы. М.: Изд. дом “Вильямс”, 2003. – 832 с.
9. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Высш. шк., 2001. – 342 с.
10. Советов Б.Я., Яковлев С.А. Моделирование систем. Практикум. М.: Высш. шк., 1999. – 224 с.
11. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II. М.: Мир, 1987. – 646 с.
12. Шрайбер Т. Моделирование на GPSS. М.: Машиностроение, 1980. – 592 с.

СТРУКТУРА СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD



**КОМАНДЫ ЯЗЫКА МОДЕЛИРОВАНИЯ GPSS
ДЛЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ GPSS WORLD**

BVARIABLE	Описание логической переменной
CLEAR	Сброс статистики и удаление транзактов из модели
CONDUCT	Проведение эксперимента с использованием языка PLUS
CONTINUE	Возобновление моделирования после прерывания
EQU	Установление эквивалентных имен (символьного и числового)
EXIT	Завершение сеанса работы GPSS World
FUNCTION	Описание функции
FVARIABLE	Описание арифметической переменной
HALT	Приостановка процесса моделирования и удаление всех команд, ожидающих выполнения
INCLUDE	Загрузка и трансляция дополнительного файла модели
INITIAL	Установка начального значения сохраняемой величины, матрицы или логического переключателя
INTEGRATE	Интегрирование переменной по времени
MATRIX	Описание матрицы
QTABLE	Описание таблицы для очереди
REPORT	Указание имени файла отчета или немедленная выдача отчета
RESET	Сброс статистики
RMULT	Установка параметров генераторов случайных чисел
SHOW	Вычисление выражения и вывод полученного результата
START	Установка счетчика завершений и начало моделирования
STEP	Установление ограничения на количество попыток входа транзакта в оператор
STOP	Указание условия остановки в зависимости от количества попыток входа в оператор
STORAGE	Описание многоканального устройства
TABLE	Описание таблицы
VARIABLE	Описание числовой переменной

**ОПЕРАТОРЫ ЯЗЫКА МОДЕЛИРОВАНИЯ GPSS
ДЛЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ GPSS WORLD**

ADOPT	Изменение ансамбля транзакта
ADVANCE	Задержка транзакта на указанное время
ALTER	Проверка и изменение транзакта в группе
ASSEMBLE	Объединение заданного количества транзактов в один
ASSIGN	Изменение параметра транзакта
BUFFER	Помещение транзакта в конец цепи текущих событий
CLOSE	Заккрытие потока данных
COUNT	Подсчет количества объектов, соответствующих определенным условиям, и его запись в параметр транзакта
DEPART	Уменьшение длины очереди
DISPLACE	Изменение последующего оператора транзакта
ENTER	Вход в многоканальное устройство
EXAMINE	Проверка принадлежности группе
EXECUTE	Выполнение другого оператора
FAVAIL	Перевод одноканального устройства в состояние “доступно”
FUNAVAIL	Перевод одноканального устройства в состояние “недоступно”
GATE	Проверка состояния объекта модели и выбор направления движения транзакта
GATHER	Ожидание заданного количества транзактов
GENERATE	Создание транзакта
INDEX	Изменение параметра транзакта
INTEGRATION	Включение / отмена интегрирования переменных
JOIN	Ввод элемента в числовую группу или в группу транзактов
LEAVE	Выход из многоканального устройства
LINK	Вход в цепь пользователя
LOGIC	Изменение состояния логического переключателя
LOOP	Цикл
MARK	Начало отсчета времени пребывания транзакта в модели с нуля
MATCH	Ожидание входа транзакта в сопряженный оператор MATCH
MSAVEVALUE	Присвоение значения элементу матрицы
OPEN	Открытие потока данных
PLUS	Вычисление выражения на языке PLUS и его присваивание параметру
PREEMPT	Вход транзакта в устройство с возможностью прерывания обработки другого транзакта
PRIORITY	Изменение приоритета транзакта
QUEUE	Увеличение очереди
READ	Ввод следующей строки из потока данных
RELEASE	Выход из одноканального устройства

REMOVE	Удаление элемента из числовой группы или из группы транзактов
RETURN	Освобождение устройства, занятого оператором PREEMPT
SAVAIL	Перевод многоканального устройства в состояние “доступно”
SAVEVALUE	Присваивание значения сохраняемой величине
SCAN	Проверка группы транзактов
SEEK	Изменение указателя в потоке данных
SEIZE	Вход в одноканальное устройство
SELECT	Запись номера выбранного объекта в параметр транзакта
SPLIT	Создание ансамбля транзактов
SUNAVAIL	Перевод многоканального устройства в состояние “недоступно”
TABULATE	Внесение изменений в таблицу
TERMINATE	Удаление транзакта
TEST	Проверка условия и выбор направления движения транзакта
TRACE	Трассировка активного транзакта
TRANSFER	Переход к указанному оператору
UNLINK	Удаление транзакта из цепи пользователя
UNTRACE	Отмена трассировки транзакта
WRITE	Передача величины в поток данных

ФУНКЦИИ ДЛЯ ИМИТАЦИИ СЛУЧАЙНЫХ ВЕЛИЧИН В GPSS WORLD

Распределение	Функция	Аргументы
Биномиальное	BINOMIAL (A,B,C)	А - номер генератора СРРЧ В - количество испытаний С – вероятность
Гамма	GAMMA (A,B,C,D)	А - номер генератора СРРЧ В - смещение С – интенсивность D – форма
	<u>Примечание.</u> Одним из частных случаев гамма-распределения является распределение Эрланга. Поэтому функция GAMMA применяется для имитации случайных величин, распределенных по закону Эрланга k -го порядка с математическим ожиданием M . При этом аргументы функции GAMMA(A,B,C,D) должны быть следующими: $B=0$, $C=M/k$, $D=k$.	
Гауссовское (нормальное)	NORMAL (A,B,C)	А - номер генератора СРРЧ В - математическое ожидание (среднее значение) С - стандартное (среднее квадратическое) отклонение
Дискретное равномерное	DUNIFORM (A, B, C)	А - номер генератора СРРЧ В - минимальное значение С - максимальное значение
Пуассоновское	POISSON (A, B)	А - номер генератора СРРЧ В - математическое ожидание (среднее значение)
Равномерное	UNIFORM (A, B, C)	А - номер генератора СРРЧ В - минимальное значение С - максимальное значение
Треугольное	TRIANGULAR (A, B, C, D)	А - номер генератора СРРЧ В - минимальное значение С - максимальное значение D - мода
Экспоненциальное	EXPONENTIAL (A, B, C)	А - номер генератора СРРЧ В - смещение С - математическое ожидание (среднее значение)

СТАНДАРТНЫЕ ЧИСЛОВЫЕ АТТРИБУТЫ

Системные СЧА

RNj	Значение j-го генератора случайных чисел
C1	Текущее значение относительного системного времени
AC1	Текущее значение абсолютного системного времени
TG1	Текущее значение счетчика завершений
XN1	Номер активного транзакта
Z1	Размер свободной оперативной памяти в байтах

СЧА транзактов

M1	Время пребывания транзакта в модели
PR	Приоритет транзакта
Pj	Значение j-го параметра
MPj	Разность модельного времени и значения j-го параметра
MBj	Признак синхронизации: 1, если транзакт в операторе j принадлежит тому же ансамблю, что и текущий транзакт, 0 – в противном случае

СЧА операторов

Nj	Общее количество транзактов, вошедших в оператор j
Wj	Текущее количество транзактов, находящихся в операторе j

СЧА одноканальных устройств

F	Текущее состояние устройства: 0 – устройство свободно, 1 – занято (операторами PREEMPT или SEIZE)
FI	Признак состояния прерывания: 1 – устройство в состоянии прерывания (оператором PREEMPT), 0 – нет состояния прерывания
FR	Коэффициент загрузки
FV	Признак доступности устройства: 1 – доступно, 0 – недоступно (переведено в состояние недоступности оператором FUNAVAIL)
FC	Общее количество входов в устройство
FT	Среднее время пребывания транзакта в устройстве

СЧА многоканальных устройств

S	Текущее количество занятых каналов
R	Текущее количество свободных каналов
SR	Коэффициент загрузки
SA	Среднее количество занятых каналов
SM	Максимальное количество занятых каналов
SC	Общее количество входов в устройство
ST	Среднее время пребывания транзакта в устройстве

SE	Признак свободного состояния устройства: 1 – свободно, 0 – хотя бы один канал занят
SF	Признак заполненности устройства: 1 – заполнено, 0 – хотя бы один канал свободен
SV	Признак доступности устройства: 1 – доступно, 0 – недоступно (переведено в состояние недоступности оператором SUNAVAIL)

СЧА очередей

Q	Длина очереди
QA	Средняя длина очереди
QM	Максимальная длина очереди
QC	Общее количество входов в очередь
QZ	Количество нулевых входов
QT	Среднее время пребывания транзакта в очереди (включая нулевые входы)
QX	Среднее время пребывания транзакта в очереди (без учета нулевых входов)

СЧА таблиц

ТВ	Среднее значение табулируемой величины
ТС	Общее количество входов в таблицу
TD	Стандартное отклонение табулируемой величины

СЧА цепей пользователя

C	Текущее количество транзактов в цепи пользователя
CA	Среднее количество транзактов в цепи пользователя
CM	Максимальное количество транзактов в цепи пользователя
CC	Общее количество входов в цепь пользователя
CT	Среднее время пребывания транзакта в цепи пользователя

СОДЕРЖАНИЕ

Введение.....	3
1. Система имитационного моделирования GPSS World:	
концептуальные основы и принципы работы	5
1.1. Назначение и принципы работы системы моделирования GPSS World.....	5
1.2. Работа с системой моделирования GPSS World. Пример GPSS-модели	6
1.3. Интерпретация результатов моделирования	8
2. Система имитационного моделирования GPSS World:	
основные возможности	11
2.1. Моделирование потоков событий. Транзакты.....	11
2.2. Структура GPSS-модели.....	12
2.3. Моделирование многоканальных СМО	13
2.4. Моделирование работы объекта за заданное время.....	14
2.5. Моделирование систем с заявками разных типов. Управление движением транзактов.	
Параметры транзактов	15
2.6. Моделирование систем с заданным соотношением заявок двух типов.....	18
2.7. Простейшие вычисления в GPSS-модели. Сохраняемые величины	20
2.8. Циклы	22
3. Имитация случайных событий и величин в GPSS-моделях.....	24
3.1. Имитация случайных величин в GPSS World.....	24
3.2. Имитация случайных событий в GPSS World. Функции.	
Моделирование систем с заявками нескольких типов.....	27
4. Стандартные числовые атрибуты	30
4.1. Понятие стандартных числовых атрибутов	30
4.2. Примеры применения СЧА: моделирование СМО с ограничениями на очередь.....	30
4.3. Примеры применения СЧА: выбор устройства для обслуживания заявки	32
5. Моделирование взаимосвязанных процессов.....	37
5.1. Проверка нескольких условий. Логические переменные	37
5.2. Создание ансамблей транзактов	40
5.3. Объединение нескольких транзактов в один	41
5.4. Логические переключатели	42
5.5. Управление передвижением транзактов в зависимости от состояния элементов модели	42
5.6. Синхронизация движения транзактов в модели	46
6. Средства анализа времени в GPSS-моделях	48
6.1. Анализ времени пребывания транзакта в модели	48
6.2. Время моделирования	50
7. Моделирование замкнутых систем массового обслуживания.....	53
8. Отображение информации в GPSS-моделях в табличной и графической форме	57
8.1. Сбор данных о распределении значений характеристик модели. Таблицы	57
8.2. Графическое отображение моделируемых величин во времени	61
9. Дополнительные возможности системы имитационного моделирования GPSS World.....	67
9.1. Дополнительные возможности управления перемещением транзактов.....	67
9.2. Имитация недоступного состояния устройств	69
9.3. Имитация обслуживания заявок с приоритетами.....	76
9.4. Матрицы.....	81
9.5. Цепи пользователя	85
9.6. Косвенная адресация.....	93
9.7. Арифметические переменные	95
Литература	96
Приложение 1. Структура системы имитационного моделирования GPSS World.....	97
Приложение 2. Команды языка моделирования GPSS для системы моделирования GPSS World.....	98
Приложение 3. Операторы языка моделирования GPSS для системы моделирования GPSS World..	99
Приложение 4. Функции для имитации случайных величин в GPSS World	101
Приложение 5. Стандартные числовые атрибуты.....	102

Учебное издание

Сморodinский Сергей Степанович,
Батин Николай Владимирович

ОПТИМИЗАЦИЯ РЕШЕНИЙ НА ОСНОВЕ КОМПЬЮТЕРНЫХ ИМИТАЦИОННЫХ МЕТОДОВ И МОДЕЛЕЙ

УЧЕБНОЕ ПОСОБИЕ

по курсу “Моделирование систем”
для студентов специальности
"Автоматизированные системы обработки информации"
дневной и дистанционной форм обучения

В 2-х частях

Часть 2

Редактор Т.Н.Крюкова
Корректор Е.Н.Батурчик

Подписано в печать	Формат 60x84 1/16.	Бумага офсетная
Гарнитура “Таймс”.	Печать ризографическая.	Усл. печ. л.
Уч.-изд. л. 5,5	Тираж 100 экз.	Заказ

Издатель и полиграфическое исполнение: Учреждение образования
“Белорусский государственный университет информатики и радиоэлектроники”
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0131518 от 30.04.2004.
220013, Минск, П. Бровки, 6