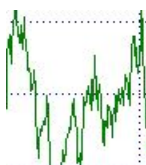


МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ИНСТИТУТ СОДЕРЖАНИЯ И МЕТОДОВ ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ
“КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

В.Н. ТОМАШЕВСКИЙ, Е.Г. ЖДАНОВА

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СРЕДСТВАМИ GPSS/PC



КИЕВ 1998

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ИНСТИТУТ СОДЕРЖАНИЯ И МЕТОДОВ ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ
“КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”**

В.Н. ТОМАШЕВСКИЙ, Е.Г. ЖДАНОВА

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СРЕДСТВАМИ GPSS/PC

Рекомендовано Министерством образования Украины
как учебное пособие для студентов специальности
“Информационные управляющие системы и технологии”

Киев 1998

УДК 681.32

Томашевський В.М., Жданова О.Г. Імітаційне моделювання засобами системи GPSS/PC: Навч. посібник. – К.: ІЗМН, НТТУ“КПІ”, 1998. - 123 с. – Рос. мовою.

У навчальному посібнику розглядаються основи імітаційного моделювання засобами мови GPSS/PC. Наведено багато прикладів і завдань, які дозволяють самостійно опанувати цю мову для моделювання економічних, технічних, технологічних і інших систем.

Призначений в першу чергу для студентів іноземців, що навчаються за фахом “Інформаційні управляючі системи і технології”, також буде корисним іншим студентам, магістрам і аспірантам.

Іл. 12. Табл. 39. Бібліогр.: 12 назв.

Рецензенти:

О.А. Волков, д-р техн. наук, проф.

В.П. Вінницький, д-р техн. наук, проф.

ISBN

© В.М. Томашевський, О.Г. Жданова, 1998

ВВЕДЕНИЕ

Имитационное моделирование применяется во всех сферах деятельности человека начиная от моделей технических, технологических и организационных

систем и заканчивая проблемами развития человечества и вселенной. в обзоре [1] показано, что имитационное моделирование является едва ли не самым распространенным средством, используемым на практике.

Основная ценность имитационного моделирования заключается в том, что в основе его лежит методология системного анализа. Оно позволяет осуществить исследования проектируемой или анализируемой системы по схеме операционного анализа, включающей взаимосвязанные этапы: содержательная постановка задачи, разработка концептуальной модели, разработка и программная реализация имитационной модели, проверка адекватности модели и оценка точности результатов моделирования, планирование и проведение экспериментов, принятие решений. Это позволяет использовать имитационное моделирование как универсальный метод для принятия решений в условиях неопределенности и для учета в моделях трудно формализуемых факторов, а также применять основные принципы системного подхода для решения практических задач.

Широкому использованию этого метода на практике препятствует необходимость создания программных реализаций имитационных моделей, которые воспроизводят в модельном времени динамику функционирования моделируемой системы. По сравнению с традиционными методами программирования разработка имитационной модели требует перестроения принципов мышления. Принципы, положенные в основу имитационного моделирования дали толчок к развитию объектного программирования. Поэтому усилия разработчиков программных средств имитации направлены на упрощение создания программных реализаций имитационных моделей. Для этого создаются специализированные языки и системы имитационного моделирования. Программные средства имитации в своем развитии прошли несколько поколений, начиная с языков моделирования и средств автоматизации конструирования моделей [2] до генераторов программ, интерактивных [3-7] и интеллектуальных систем [8-1.10]. Основное назначение всех этих средств – уменьшение трудоемкости создания программных реализаций имитационных моделей.

Одним из первых языков моделирования, облегчающих процесс написания имитационных программ, был язык GPSS, созданный фирмой IBM. Этот язык раньше входил в первую десятку лучших языков программирования, опережая транслятор с языка АЛГОЛ, и был реализован практически на всех типах ЭВМ. В настоящее время на персональных ЭВМ реализован транслятор для DOS – GPSS/PC и для OS2 - GPSS/H. Изучение и программирование на этом языке позволяет понять принципы разработки имитационных программ и научиться работать с имитационными моделями.

GPSS (*General Purpose Simulating System* - общецелевая система моделирования) является языком моделирования, используемым для построения дискретных моделей и проведения моделирования на ЭВМ.

Модели систем на GPSS могут быть записаны в виде блок-схем или представлены в виде последовательности строк программы, эквивалентных

блок-схеме. Блок-схема представляет собой набор фигур с характерным очертанием блоков языка GPSS, соединенных между собой линиями. Блоки представляют собой подпрограммы, реализованные средствами макроассемблера. Язык моделирования содержит 37 блоков для создания имитационных программ.

В GPSS входят специальные средства для описания динамического поведения систем путем смены состояний в дискретные моменты времени.

GPSS представляет собой язык и транслятор. Как каждый язык он содержит словарь и грамматику, с помощью которых могут быть разработаны модели систем определенного типа. Транслятор языка построен как компилятор-интерпретатор и работает в две фазы. На первой фазе компиляции проверяется синтаксис и семантика написания строк GPSS-программы, а на второй - интерпретации, осуществляется продвижение транзактов по модели от блока к блоку.

1. ОБЪЕКТЫ GPSS

Язык GPSS - это язык декларативного типа, построенный по принципам объектно-ориентированного языка. Основными элементами этого языка являются **транзакты** и **блоки**, которые отображают, соответственно, динамические и статические объекты моделируемой системы

Объекты моделируемой системы могут быть предназначены для различных целей. Выбор объектов в конкретной модели зависит от характеристик моделируемой системы. Каждый объект имеет некоторое число свойств, называемых в GPSS **стандартными числовыми атрибутами (СЧА)**. Часть СЧА доступна пользователю только для чтения, а на значения других он может воздействовать путем использования соответствующих блоков.

1.1. **Блоки и транзакты**

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

В GPSS концепция передачи управления от блока к блоку имеет специфические особенности. Последовательность блоков GPSS-модели отражает направления, по которым происходит движение перемещающихся элементов. Каждый такой элемент называется **транзактом**. Транзакты являются динамическими элементами GPSS-модели.

Блоки языка GPSS представляют собой подпрограммы, написанные на языке макроассемблера и содержат набор параметров (операндов) для обращения к ним. Как и все языки моделирования, GPSS имеет некоторый внутренний механизм передачи управления, который реализуется в модельном времени, что позволяет отображать динамические процессы, протекающие в реальных системах. Передача управления от блока к блоку в GPSS-программах осуществляется посредством движения транзактов в модельном времени, т. е.

обращение к подпрограммам блоков происходит через движение транзактов от блока к блоку программы GPSS.

Содержательное значение транзактов определяет разработчик модели. Именно он устанавливает аналогию между транзактами и реальными элементами моделируемой системы. Такая аналогия никогда не указывается интерпретатору GPSS, она остается в уме разработчика моделей. В таблице 1.1 представлены некоторые примеры возможных аналогий между транзактами и элементами реальных систем.

Таблица 1.1

<i>Система</i>	<i>Элементы систем, символизируемые транзактами</i>
Магазин	Покупатель
Автомобильное шоссе	Автомобиль
Склад	Заявка

В языке GPSS все транзакты нумеруются по мере их появления в модели. Параметры транзактов отображают свойства моделируемого динамического объекта. Например, если моделируется движение автомобилей на участке дороги, то параметрами транзакта (автомобиля) могут быть: скорость, тормозной путь, габариты и другие в зависимости от целей моделирования.

С программной точки зрения транзакт представляет некоторую структуру данных. Такая структура содержит поля:

имя или номер транзакта
текущее модельное время
номер блока, где находится транзакт
номер блока, куда он должен быть передан
момент времени передачи
приоритет транзакта
время возникновения транзакта
Параметры транзакта: P1,P2,...

Каждый транзакт занимает некоторое число ячеек памяти ЭВМ. После того, как он закончит свое движение по блокам модели, его необходимо уничтожить для освобождения памяти, чтобы избежать ее переполнения. Так как транслятору не известно, сколько транзактов одновременно будут находиться в модели, то память под транзакты должна выделяться динамически.

Таким образом, в самом начале моделирования в GPSS-модели нет ни одного транзакта. В процессе моделирования транзакты входят в определенный

момент времени в модель, в соответствии с теми логическими потребностями, которые возникают в моделируемой системе. Подобным же образом транзакты покидают модель в зависимости от специфики моделирования. В общем случае в модели существует некоторое количество транзактов, однако в каждый момент времени двигается только один транзакт.

Если транзакт начал свое движение, он перемещается от блока к блоку по пути, предписанному блок-схемой. В тот момент, когда транзакт входит в блок, на исполнение вызывается соответствующая подпрограмма и далее транзакт (в общем случае) пытается войти в следующий блок. Такое продвижение транзакта продолжается до тех пор, пока не произойдет одно из возможных событий:

- 1) транзакт входит в блок, функцией которого является задержка транзакта на некоторое определенное моделью время;
- 2) транзакт входит в блок, функцией которого является удаление транзакта;
- 3) транзакт пытается войти в следующий блок в соответствии с логикой модели, однако блок не принимает этот транзакт. В этом случае транзакт остается в том блоке, в котором он в настоящее время находится. Позднее он будет повторять свою попытку войти в следующий блок. Когда условия в модели изменятся, одна из таких попыток может оказаться успешной. После этого транзакт продолжает свое перемещение в модели.

Если возникло одно из указанных условий, транзакт остается на месте, и начинается перемещение в модели другого транзакта.

1.2. *Объекты типа "ресурсы"*

Аналогами обслуживающих устройств и реальных систем в GPSS являются объекты типа "ресурсы". К объектам этого типа относятся ***приборы, многоканальные устройства и логические ключи.***

Прибор (одноканальное устройство) представляет собой ресурс, который в любой момент может быть занят только одним транзактом. Интерпретатор автоматически вычисляет такие СЧА, как общее время занятости прибора и число транзактов, занимавших прибор, что позволяет ему определить коэффициент использования каждого прибора и среднее время занятости прибора одним транзактом.

Многоканальные устройства представляют собой объекты типа "ресурсы" для параллельной обработки, они могут быть использованы несколькими транзактами одновременно. Пользователь определяет емкость (количество каналов) каждого многоканального устройства, используемого в модели, а интерпретатор ведёт учет числа каналов, занятых в каждый момент времени. Интерпретатор автоматически подсчитывает число транзактов, входящих в многоканальное устройство, среднее число каналов, занятых одним транзактом и среднее время пребывания транзакта в устройстве.

Определённые события в системе могут заблокировать или изменить движение транзактов. (Например, кассир кинотеатра, уходя на обед, ставит

табличку "К другому окну", и все последующие клиенты в течение обеда обращаются в эту кассу). Для моделирования этих ситуаций введены логические ключи. Транзакт может устанавливать эти ключи в положение "включено" или "выключено". Впоследствии состояние ключа может быть проверено другими транзактами для определения пути следования или ожидания момента изменения состояния ключа. Состояние ключа может быть изменено любым другим транзактом.

1.3. *Переменные*

Арифметические переменные позволяют вычислять арифметические выражения, состоящие из операций над СЧА объектов и блоков. В выражениях переменных могут быть использованы пользовательские и библиотечные функции.

Булевы переменные позволяют пользователю проверять в одном блоке одновременно несколько условий, исходя из состояния или значения объектов или их СЧА.

1.4. *Функции*

Используя *функции*, пользователь может производить вычисления непрерывных или дискретных функциональных зависимостей между аргументом функции (независимая величина) и зависимым значением функции. Функции широко применяются, например, для задания случайного интервала времени между генерацией двух транзактов. Все функции в GPSS задаются табличным способом с помощью операторов описания функций.

1.5. *Ячейки и матрицы ячеек сохраняемых величин*

Ячейки и матрицы ячеек сохраняемых величин используются для сохранения некоторой пользовательской числовой информации. Любой транзакт может осуществить запись в эти объекты. Впоследствии, записанную информацию может считать любой транзакт. Таким образом, эти объекты являются глобальными и доступны из любой части модели.

1.6. *Очереди*

В любой системе движение потока транзактов может быть задержано из-за недоступности ресурсов. Например, требуемые приборы или многоканальные устройства уже заняты. В этом случае задержанные транзакты ставятся в *очередь* - еще один тип объектов GPSS. Учет этих очередей составляет одну из основных функций интерпретатора. Пользователь может специально определить точки в модели, в которых нужно собирать статистику об очередях, т.е. установить

регистратор очереди. Тогда интерпретатор будет автоматически собирать статистику об очередях (длину очереди, среднее время пребывания в очереди и т.д.). Вся эта информация является СЧА и доступна пользователю в процессе моделирования. Число задержанных транзактов и продолжительность этих задержек определяется только в этих заданных точках.

1.1. *Таблицы*

Объект **таблица** предназначен для сбора статистики о случайных величинах, заданных пользователем. Таблица состоит из частотных классов, в которые заносится число попаданий конкретной величины (обычно некоторого СЧА). Для каждой таблицы вычисляется математическое ожидание и среднеквадратичное отклонение.

2. **Таймер модельного времени**

Различные события реальных систем происходят в течение некоторого периода времени. Например, покупатели приходят в магазин, когда подходит их очередь, они попадают на обслуживание. Покупка совершена и покупатель покидает магазин. Если все эти события представить в модели, то их возникновение должно происходить на фоне модельного времени. Интерпретатор автоматически обслуживает таймер модельного времени.

Когда начинается моделирование, в интерпретаторе планируется приход первого транзакта. После этого таймер модельного времени устанавливается на значение времени, соответствующее моменту появления первого транзакта в модели. Этот транзакт, а также другие, если они приходят в этот же момент времени, входит в модель. Далее он продвигается через все возможные блоки модели, которые ему встречаются. События, возникающие как следствие продвижения транзакта через блоки, планируются на последующие моменты времени. Естественно, что в этот первый отмеченный момент времени ничего более в системе не происходит. Интерпретатор GPSS продвигает далее таймер до значения времени, когда происходит следующее запланированное им событие. Когда в этот второй, отмеченный таймером момент времени, не остается транзакта, который надо перемещать, таймер опять продвигается вперед и т.д. Именно таким образом и происходит изменение модельного времени, т. е. от события к событию.

Особенности таймера GPSS

1. Таймер GPSS регистрирует целые значения, т.е. события могут возникать **только в целые** моменты времени. Это сделано с целью ускорения процесса моделирования, так как целочисленная арифметика выполняется процессором ЭВМ быстрее и требует меньше памяти.

2. Единица времени, которая может быть отмечена таймером, определяется разработчиком. Однако единицу времени никогда не сообщают интерпретатору. Это значение выражают в неявном виде в форме временных данных, вводимых в модель. Если все такие данные выражены в минутах, то подразумеваемой единицей времени будет минута. Если все данные выражены в миллисекундах, то такой единицей времени будут миллисекунды. Разработчик может задавать такую единицу времени, которая ему удобна для того, чтобы правильно отразить события реальной системы в модели.

3. GPSS является интерпретатором "следующего события". Иначе говоря, после того, как модель полностью скорректирована в данный момент времени, таймер продвигается к следующему моменту времени, на который запланировано очередное событие. Интервал модельного времени пропускается, если на этом интервале нет событий.

3. Типы операторов GPSS

Операторы GPSS делятся на следующие три типа:

- блоки;
- операторы описания данных и контроля управления;
- команды GPSS.

Общие сведения о формате блоков GPSS

В GPSS для ссылок на числа, блоки и объекты применяются **имена**. Имя представляет собой алфавитно-цифровую последовательность, длиной до 20 символов, начинающуюся с буквы. Допустимо применение символов только латинского алфавита.

Формат GPSS-блоков таков:

[Номер строки] [<Метка>] <Операция> <Операнды> [<Комментарии>]

1. **Номер строки.** Необязательное поле. Начинается в первой позиции строки. Представляет собой десятичное число.

2. **Метка.** Содержимым поля является имя. В некоторых блоках поле обязательно.

3. **Операция.** Операциями блоков являются глаголы, описывающие основные функциональные назначения блоков. Каждый из блоков характеризуется своим собственным предписанным ему глаголом.

4. **Операнды.** Блоки могут содержать различные связанные с ними операнды. Операнды блоков задают информацию, специфичную для действия данного блока. Под значениями операндов могут подразумеваться значения аргументов, используемых при обращениях к подпрограмме.

Число операндов каждого блока зависит от типа блока. В блоках не может использоваться более 7 операндов. Операнды в общем случае обозначены символами: **A, B, C, D, E, F, G**.

Одни операнды некоторых блоков должны быть специфицированы всегда, а другие могут задаваться или не задаваться. Операнды следуют один за другим и отделяются запятыми, между ними не должно быть пробелов. Пробел в поле операндов означает, что операнды закончились и интерпретатор прекращает чтение строки. Пропуск в записи необязательных операторов отмечается символом ",".

5. **Комментарии.** Необязательное поле. В GPSS допускаются комментарии с использованием прописных и строчных букв латинского алфавита. Отделяются от поля операндов символом ";". Допускается запись комментария с начала строки. В этом случае в первой позиции строки ставится символ ";" или "*".

Строка описания блока может содержать до 79 символов. Скобки [] указывают на необязательные поля.

!

Именами и метками не могут быть начальные символы названий блоков, операторов, команд и СЧА.

4. Внесение транзактов в модель. Блок GENERATE (ГЕНЕРИРОВАТЬ)

Блок GENERATE - это блок, через который транзакты входят в модель. Не существует ограничений на число различных блоков GENERATE в одной модели.

Интервал времени между последовательными появлениями транзактов блока GENERATE называют **интервалом поступления**. Когда транзакт входит в модель через блок GENERATE, интерпретатор планирует время поступления следующего транзакта путем розыгрыша случайного числа в соответствии с распределением интервалов поступления и с последующим добавлением разыгранного значения к текущему значению таймера. При достижении этого значения времени следующий транзакт вводится в модель через блок GENERATE и т.д.

Формат блока :

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

<i>Операнд</i>	<i>Значение</i>	<i>Значение по умолчанию</i>
A	Средний интервал времени (число, СЧА)	0
B	Половина поля допуска равномерно распределенного интервала (число, СЧА)	0

<i>Операнд</i>	<i>Значение</i>	<i>Значение по умолчанию</i>
-----------------------	------------------------	-------------------------------------

C	Смещение интервалов	Смещение отсутствует
D	Ограничитель транзактов	∞
E	Уровень приоритета транзакта. Значения от 0 до 127	0

Разработчик должен задать функцию случайного распределения интервалов времени поступления транзактов в блоке GENERATE. Информация, необходимая для этого задается операндами **A** и **B**. Если операнд **B** не задается то интервал времени поступления будет детерминированный.

Все возможные виды случайных распределений интервалов поступления в GPSS делят на равномерное распределение и прочие виды распределений.

Таким образом, специально рассматривают простейшее из всех нетривиальных случайных распределений, а именно: равномерное распределение. Равномерное распределение задается значением операндов **A ± B**. Это связано с тем, что всегда легко определить некоторое значение и разброс возможных значений относительно него. Применение более сложных видов распределений требует использования в качестве операндов **A** и **B** ссылок на функции (см. разд. 13).

C - момент времени, в который должен появиться первый транзакт в блоке GENERATE. После этого первого прихода все остальные приходы транзактов возникают в соответствии с распределением, заданным операндами **A** и **B**.

D - задает граничное число общего числа транзактов, которое может войти в модель через данный блок GENERATE в течение времени моделирования. Если это число достигнуто, этот блок GENERATE перестает быть активным.

E - задает уровень или класс приоритета каждого из транзактов, вводимых в модель через данный блок GENERATE. Всего существует 128 различных уровней, которые задаются с помощью чисел от 0 до 127. Чем больше число, тем больше приоритет.

!

1. Транзакты не могут входить в блок GENERATE, так как он сам их генерирует. Если в модели встречается подряд два и более блоков GENERATE, то последний блок переопределяет операнды предыдущих блоков.
2. Когда операнды задаются в виде констант, они должны быть неотрицательными целыми числами.

Пример 4.1

1. Задание равномерного закона поступления:
GENERATE 6,4

Операнды: A=6, B=4. Интервал времени прибытия, таким образом, является случайным числом со средним значением 6 и полем допуска 8, т.е. он может принимать только одно из 9 различных значений: 2 3 4 5 6 7 8 9 10.

2. Задание детерминированного значения интервалов поступления:

GENERATE 10

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

3. Генерирование одиночного транзакта.

GENERATE „1

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

Блоки GENERATE являются основным средством для создания транзактов и ввода их в модель.

Помимо блока GENERATE, для создания транзактов используется также блок SPLIT, который создает заданное число копий вошедшего в блок транзакта. Эти копии принадлежат к тому же семейству, что и породивший их транзакт.

Для удаления транзактов копий, принадлежащих к одному семейству, может быть использован блок ASSEMBLE.

Блоки MATCH и GATHER предназначены для управления движением транзактов, принадлежащих к одному семейству. Для управления такими транзактами используются блоки GATE и GATE NM.

Блок имеет формат:

SPLIT A,[B],[C]

SPLIT выполняет функцию копирования входящего в него транзакта, который называется исходным или порождающим.

В поле A задается число создаваемых копий. Операнд A может быть именем, положительным целым, СЧА, СЧА*<параметр>. Если вычисленное значение аргумента поля A равно нулю, то блок SPLIT не выполняет никаких операций. После создания копий транзакт пытается перейти к следующему по номеру блока. Все копии формируются в момент входа порождающего транзакта в блок SPLIT.

Поле B задает номер следующего блока, к которому переходят копии исходного транзакта, причем значение вычисляется для каждой копии отдельно. Операнд B может быть именем, положительным целым, СЧА, СЧА*<параметр>.

В поле C может быть задан номер параметра, используемого для присвоения копиям последовательных номеров. Операнд C может быть именем, положительным целым, СЧА, СЧА*<параметр>.

Каждая новая копия становится членом семейства транзактов, порожденных одним исходным транзактом, который был создан блоком GENERATE. Транзакты, принадлежащие к одному семейству, интерпретатор объединяет в список.

По связям внутри семейства транзактов нельзя установить, какой из транзактов семейства является порождающим. Если копия транзакта входит в блок SPLIT, то вторичная копия становится членом того же семейства, что и первичная копия. Таким образом, каждый транзакт является членом одного и только одного семейства. Семейство может состоять из произвольного числа транзактов. Когда транзакт уничтожается, интерпретатор автоматически исключает его из членов соответствующего семейства. Следовательно, семейство существует до тех пор, пока из модели не удаляется последний из его членов. При каждом удалении транзакта связи между ними корректируются, чтобы транзакты данного семейства по-прежнему образовывали замкнутый список.

В модели одновременно может существовать произвольное число семейств, оно все время меняется, поскольку каждый генерируемый блоком GENERATE транзакт создает новое семейство.

1. Удаление транзактов из модели. Блок **TERMINATE** (ЗАВЕРШИТЬ)

Транзакты удаляются из модели, попадая в блок TERMINATE, т.е. освобождается память, выделенная для транзакта. Эти блоки всегда позволяют выйти всем транзактам, которые пытаются это сделать. В модели может быть любое число блоков TERMINATE.

Формат блока:

TERMINATE [A]

Операнд **A** является величиной уменьшения специального счетчика, называемого *счетчиком завершений*. Этот операнд задает величину, которая должна вычитаться из счетчика каждый раз, когда транзакт входит в блок TERMINATE. По умолчанию $A=0$. Вход транзакта в такой блок TERMINATE не вызывает уменьшения содержимого счетчика завершений.

Счетчик завершений - это ячейка памяти ЭВМ, которая хранит целое положительное число. Начальное значение этому счетчику присваивается в начале моделирования и равно значению операнда **A** команды **START** (НАЧАТЬ). В процессе моделирования транзакты попадают в блоки TERMINATE и, таким образом, происходит уменьшение содержимого счетчика. Моделирование завершается, когда счетчик сбрасывается в ноль.

!!

1. В модели может быть много блоков TERMINATE, но счетчик завершений - один.
2. Нельзя путать ограничитель транзактов в блоке GENERATE и счетчик завершений. Ограничитель задает число транзактов, вошедших в модель, а счетчик - вышедших из модели.

Интерпретатор начинает моделирование, когда встречается команда START. Её формат:

START A

Управление продолжительностью процесса моделирования

В GPSS можно управлять продолжительностью процесса моделирования двумя способами:

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

Первый способ:

1. В команде START операнду A присваивается значение заданного числа транзактов.
2. Во всех блоках TERMINATE, через которые покидают модель транзакты заданного типа, операнду A присваивается значение 1 или другое, отличное от нуля (соответственно содержательному смыслу транзакта).
3. Во всех прочих блоках TERMINATE в модели подразумевается использование операнда A (по умолчанию A=0), т.е. значение счетчика завершений не будет зависеть от этих блоков.

Первый способ позволяет закончить моделирование, когда через модель пройдет заданное число транзактов, например 1000.

GENERATE 40,5

...

TERMINATE 1

START 1000

Второй способ:

Пусть разработчик выбрал в качестве единицы времени 1 мин. и хочет промоделировать поведение системы в течение 8 час, после чего моделирование должно быть закончено. Это можно сделать так:

1. Включить в модель сегмент таймера из двух блоков.

GENERATE 480

TERMINATE 1

2. Во всех прочих блоках TERMINATE в модели подразумевается использование операнда A (по умолчанию A=0). Это означает, что прекращение моделирования, определяемое счетчиком завершений, не должно зависеть от других блоков TERMINATE.

3. В команде START в качестве операнда A должна быть использована 1.

Таким образом, в процессе моделирования завершения движения транзактов, которые происходят время от времени в других блоках TERMINATE, не оказывают воздействия на счетчик завершений. В момент времени 480 транзакт выйдет из блока GENERATE и сразу же перейдет к блоку TERMINATE. Счетчик завершений уменьшится на 1 и интерпретатор завершит моделирование.

2. Элементы, отображающие одноканальные обслуживающие устройства

Рассмотрим теперь, какие элементы можно использовать для представления обслуживания. Аналогами обслуживающих устройств могут быть люди, механизмы, линии связи и другие объекты реальных систем. В GPSS такие объекты моделируются с помощью ресурсов. К объектам этого типа относятся **приборы, многоканальные устройства и логические ключи**. Рассмотрим первые два типа.

Приборы характеризуются двумя основными свойствами:

1. Каждый прибор в любой момент времени может обслуживать только одно требование. Если в процессе обслуживания появляется новое требование, то оно должно:
 - либо подождать своей очереди;
 - либо направиться куда-нибудь в другое место;
 - либо, если вновь пришедшее требование является более важным, прибор прерывает текущее обслуживание и начинает обслуживать новое требование.
2. Когда в прибор поступает требование, оно должно пробыть там время, необходимое для обслуживания.

Так как в системе может быть несколько приборов, то для их различия им должны быть даны имена. Имена могут быть либо числовыми, либо символическими. Если приборам присваивают числовые имена, числа должны быть целыми положительными.

Для того, чтобы использовать одноканальное обслуживающее устройство, необходимо выполнить такие шаги:

1. Ожидать своей очереди, если это необходимо. Ожидание требует некоторого интервала времени.
2. Когда подходит очередь - занять прибор. Событие "занятие прибора" происходит в некоторой точке модельного времени.
3. Прибор остается в состоянии занятости, пока не закончится обслуживание. Для обслуживания требуется некоторый интервал времени.
4. Когда обслуживание заканчивается - освободить прибор. Событие "освобождение прибора" происходит в некоторой точке времени.

Эта последовательность шагов выполняется GPSS при моделировании использования прибора. Шаги 2 и 4 реализуются блоками **SEIZE** (ЗАНЯТЬ) и **RELEASE** (ОСВОБОДИТЬ).

Формат блока :

SEIZE A

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
----------------	-----------------	-------------------------------

A	Имя (символическое или числовое) занимаемого прибора	Ошибка
----------	--	--------

Этот блок обладает такими свойствами:

1. Если в текущий момент прибор используется, транзакт не может войти в этот блок, т.е. ему не разрешается занять прибор в данный момент времени, то он должен ожидать своей очереди. Такой запрет на вход в блок приводит к тому, что транзакт на время исключается из рассмотрения.

2. Если прибор свободен, транзакт может войти в блок. Вход транзакта в блок вызывает выполнение подпрограммы обработки этого блока. Результатом выполнения подпрограммы является изменение статуса прибора из "незанят" в "занят".

Не требуется предварительного описания (объявления) прибора в модели, тот факт, что блок SEIZE используется, вызывает установление факта существования данного прибора.

Назначением блока RELEASE является изменение состояния ранее занятого прибора из "занят" в "незанят". Блок RELEASE никогда не запрещает вход транзакта.

Формат блока:

RELEASE A

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Имя (символическое или числовое) освобождаемого прибора	Ошибка

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

При моделировании работы прибора автоматически собирается статистическая информация о его работе:

- 1) какую часть времени прибор был занят;
- 2) сколько раз прибор был занят;
- 3) каков средний интервал времени занятости прибора.

Если в модели используются объекты типа "прибор", то в файле стандартной статистики будет представлена информация об использованных приборах:

FACILITY <i>Номер или имя прибора</i>	ENTRIES <i>Количество входов</i>	UTILIZATION <i>Коэффициент использования</i>	AVERAGE TIME <i>Среднее время пребывания транзактов в приборе</i>	AVAILABLE <i>Состояние готовности</i>
1	50	0.07	74.06	
2	51	0.10	100.29	

3	51	0.12	119.76	
OWNER <i>Номер последнего транзакта</i>	PEND <i>Количество прерванных транзактов в приборе</i>	INTER <i>Количество прерывающих прибор транзактов</i>	RETRY <i>Количество транзактов, ожидających специальных условий</i>	DELAY <i>Количество транзактов, ожидających занятие прибора</i>

Статистику о работе устройства можно наблюдать в окне устройств при работе с программой, перейдя в это окно с помощью нажатия клавиш [ALT + F].

!

1. После блока SEIZE может следовать сразу же другой блок SEIZE, если транзакт должен одновременно занять два или более приборов, например: рабочего и инструмент.
2. Транзакт не может освободить прибор, который он не занимал.

3. **Задержки во времени. Блок ADVANCE (ЗАДЕРЖАТЬ)**

Блок ADVANCE осуществляет задержку продвижения транзакта в течение некоторого интервала времени. Обычно этот интервал задается случайной переменной.

Возможные варианты распределения времени обслуживания делятся на:

- равномерное распределение интервала обслуживания,
- прочие распределения интервалов обслуживания.

Как и при использовании блока GENERATE особо рассматривается равномерное распределение случайных величин. Применение более сложных видов распределений требует использования функций (см. раздел 13).

Формат блока:

ADVANCE A,[B]

<i>Операнд</i>	<i>Значение</i>	<i>Значение по умолчанию</i>
A	Среднее время задержки на обслуживание (число, СЧА)	0
B	Половина поля допуска равномерно распределенного времени задержки (число, СЧА)	0

Блок никогда не препятствует входу транзакта. Любое число транзактов может находиться в этом блока одновременно. Когда транзакт попадает в такой

блок, выполняется соответствующая подпрограмма и вычисляется время пребывания транзакта в нем. Вновь прибывший транзакт никак не влияет на уже находящийся в блоке транзакт.

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

!

1. Дробные значения времени задержки не допускаются. Отрицательное значение задержки всегда вызывает ошибку.

Примеры использования блока ADVANCE

ADVANCE 30,5

Время задержки транзакта - случайная величина, равномерно распределенная на интервале [25,35], и принимающая одно из 11 целых значений.

Классический случай использования последовательности SEIZE - ADVANCE - RELEASE:

SEIZE PRIB
ADVANCE 16,4
RELEASE PRIB

Транзакт, двигаясь по цепочке блоков, займет устройство с символическим именем PRIB, задержится там на 16 ± 4 единицы времени и затем покинет его. После того, как транзакт войдет в блок RELEASE и соответствующая этому блоку подпрограмма закончится, интерпретатор попытается продвинуть транзакт в следующий блок модели и очередной транзакт может уже использовать устройство PRIB.

!

Блоки ADVANCE можно располагать в любых местах программы, а не только после SEIZE или перед RELEASE.

8. Сбор статистики при ожидании. Блоки QUEUE (СТАТЬ В ОЧЕРЕДЬ), DEPART (ПОКИНУТЬ ОЧЕРЕДЬ)

Эти блоки обеспечивают в GPSS возможность автоматического сбора статистических данных, описывающих вынужденное ожидание, которое может происходить время от времени в различных точках модели.

GPSS обеспечивает возможность сбора статистики с помощью средства, называемого *регистратором очереди*. При использовании регистратора очереди

в тех точках модели, где ресурсы ограничены, интерпретатор автоматически собирает информацию об ожидании:

- 1) сколько раз требования приходили в очередь;
- 2) сколько пришедших требований фактически присоединилось к очереди и сколько сразу заняли прибор;
- 3) каково было максимальное значение длины очереди;
- 4) каково было среднее число ожидающих требований;
- 5) каково было среднее время ожидания тех требований, которым пришлось ждать.

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

Формат этих блоков:

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

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Имя очереди, к которой необходимо присоединить транзакт или которую надо покинуть (числовое или символическое имя, СЧА)	Ошибка
B	Число единиц, на которое увеличивается (уменьшается) длина очереди (число, СЧА)	1

При входе транзакта в блок QUEUE выполняются четыре действия:

1. **"Счетчик входов"** для данной очереди увеличивается на **B**.
2. **"Счетчик текущего содержимого"** для данной очереди также увеличивается на **B**.
3. транзакт "привязывается" к очереди с запоминанием ее имени и значения текущего модельного времени.
4. Транзакт перестает быть элементом очереди только после того, как он переходит в блок DEPART соответствующей очереди. Когда это происходит, интерпретатор выполняет следующее:
 1. "счетчик текущего содержимого" соответствующей очереди уменьшается на **B**.
 2. используя привязку к значению времени, интерпретатор определяет, является ли время, проведенное транзактом в очереди, нулевым. Если да, то такой транзакт по определению является транзактом с "нулевым пребыванием" в очереди. Соответственно изменяется **"счетчик нулевых вхождений"**.
 3. ликвидируется привязка транзакта к очереди.

Если в модели используются объекты типа "очередь", то в файле стандартной статистики будет представлена информация об этих объектах. В конце моделирования интерпретатор автоматически выдает статистические данные, такие, как "счетчик входов", "максимальное содержимое", "среднее значение содержимого" очереди, "счетчик текущего содержимого", среднее время нахождения в очереди. Статистическая информация об ожидании выдается в таком виде:

QUEUE <i>Номер или имя очереди</i>	MAX. <i>Макси-мал ьная длина очереди</i>	CONTENT <i>Текущая длина очереди</i>	ENTRIES <i>Об щее количес-во входов</i>	ENTRIES (0) <i>Количество “нулевых” входов</i>
1	4	0.04	894	792

AVE.CON <i>Среднее содержи-мое очереди</i>	AVERAGE TIME <i>Среднее время пребывания транзактов в очереди</i>	AVE.(-0) <i>Среднее время пребывания в очереди, без учета “нулевых” входов</i>
88.59	1.37	11.99

Пример 8.1

Пусть нам нужно собирать статистику об ожидании в очереди на обслуживание прибором PRIB, тогда в сегмент модели будут введены блоки QUEUE и DEPART.

```

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

```

В этом примере все транзакты, попадающие на прибор, должны пройти через пару QUEUE - DEPART, даже если прибор свободен и его можно сразу же занять.

Пример 8.2

```
QUEUE QPR1
```

Увеличение на 1 единицу длины Q\$QPR1 очереди QPR1.

```
QUEUE QPR2,2
```

Увеличение на 2 единицы длины Q\$QPR2 очереди QPR2.

```
DEPARTQWORKER
```

Уменьшение на 1 единицу длины Q\$QWORKER очереди QWORKER.

DEPART 1,Q1

Обнуление длины очереди 1.

!

1. Когда транзакт входит в блок QUEUE, то ищется очередь с именем, определённым операндом A. Если необходимо, очередь создаётся. Блок QUEUE не поддерживает список членов очереди, он только добавляет единицы к длине очереди. Использование регистратора очереди необязательно. С его помощью интерпретатор собирает лишь статистику об ожидании. Если же регистратор не используется, то статистика не собирается, но всюду, где должна возникать очередь, она возникает. Ожидание является следствием состояния прибора, а не следствием использования регистратора. Если в планы не входит обработка статистических данных об очередях, то лучше не собирать статистику, это сэкономит время, расходуемое на моделирование.
2. Один и тот же транзакт может одновременно увеличить длины нескольких очередей.

9. Переход транзакта в блок, отличный от последующего. Блок TRANSFER (ПЕРЕДАТЬ)

В GPSS этот блок может быть использован в девяти разных режимах. Рассмотрим три основных из них.

9.1. Блок TRANSFER в режиме безусловной передачи

Формат блока:

TRANSFER ,B

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Не используется	
B	Позиция блока, в которую должен перейти транзакт	Ошибка

Позиция блока -это номер либо метка блока. Так как **A** не используется, то перед операндом **B** должна стоять запятая. В режиме безусловной передачи блок

TRANSFER не может отказывать транзакту во входе. Если транзакт входит в блок, он сразу же пытается войти в блок **В**.

!

Транслятор GPSS не улавливает пропущенную запятую вместо операнда **А**.

9.2. Статистический режим

В этом режиме осуществляется передача транзакта в один из двух блоков случайным образом.

Формат блока:

TRANSFER A,[B],C

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
А	Частота передачи транзакта в блок С, задаваемая в долях тысячи	Ошибка
В	Позиция блока, в которую должен перейти транзакт (с частотой 1- А)	Следующий по порядку блок
С	Позиция блока, в которую должен перейти транзакт (с частотой А)	Ошибка

!

Частота - не более трех цифр, первый символ записи частоты - "." (десятичная точка).

Пример 9.1

```

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

```

С частотой 0.667 переходим на блок с меткой LPRIB1 и с частотой 0.333 - на блок с меткой LPRIB2.

Пример 9.2

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

С частотой 0.6 переходим на блок SEIZE PR1 и с частотой 0.4 - на блок с меткой LPRIB2.

9.3. Режим BOTH

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

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

Пример 9.3

```
TRANSFER    BOTH, L1, L2
...
L1           SEIZE    PRI1
...
L2           SEIZE    PRI2
```

Транзакт сначала пытается перейти к блоку, помеченному L1. Если прибор PRI1 занят, транзакт пытается войти в блок с меткой L2. Если транзакт не может войти и в этот блок (прибор PRI2 тоже занят), он остается в списке текущих событий и повторяет эти попытки при каждом просмотре списка до тех пор, пока не выйдет из блока **TRANSFER**.

!

Не путать метку блока **SEIZE** и имя соответствующего этому блоку прибора.

10. Моделирование многоканальных устройств

Два или более приборов часто работают параллельно, выполняя аналогичное обслуживание (парикмахеры, контролеры, подъемные краны, турникеты и т.п.).

Прибор в GPSS используют для моделирования единичного устройства обслуживания. Два или более находящихся рядом обслуживающих устройства могут быть промоделированы в GPSS двумя или более одноканальными приборами, располагаемыми рядом, т.е. работающими параллельно. Обычно к этому необходимо прибегать, когда отдельные приборы являются разнородными, например, имеют различную интенсивность обслуживания.

Очень часто, однако, различные параллельно работающие приборы являются однородными. GPSS предоставляет для моделирования однородных параллельных приборов объект, называемый **многоканальным устройством (МКУ)**.

Количество приборов, которое моделируется каждым из МКУ, определяется пользователем. В этом смысле употребляют термин "**емкость МКУ**".

1.1. **Блоки ENTER (ВОЙТИ) и LEAVE (ВЫЙТИ)**

Использование МКУ аналогично использованию одиночного прибора. Элементом, который занимает и использует МКУ, является транзакт. При этом события происходят в хронологическом порядке:

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

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

Формат блоков:

ENTER A,[B]
LEAVE A,[B]

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

Когда транзакт входит в блок ENTER, интерпретатор выполняет такие действия:

- 1) "**счетчик входов**" МКУ увеличивается на значение операнда **B**;

2) "*текущее содержимое*" МКУ также увеличивается на значение операнда **В**;

3) "*доступная емкость*" МКУ уменьшается на значение операнда **В**.

Когда транзакт входит в блок LEAVE, выполняются обратные действия:

1) "*текущее содержимое*" МКУ уменьшается на значение операнда **В**;

2) "*доступная емкость*" МКУ увеличивается на значение операнда **В**.

Операнду **В** можно присвоить значение, не равное 1. Например, пусть транзакт моделирует корабль, а МКУ - причалы в порту. В зависимости от размера корабль может потребовать использования нескольких причалов, т.е. **В** > 1.

Если в модели используются объекты типа "многоканальное устройство", то в файле стандартной статистики будет представлена информация об этих объектах:

STORAGE	CAP.	REMAIN	MIN	MAX	ENTRIES
RPOOL	3	3	0	1	50

AVL.	AVE.C.	UTIL.	RETRY	DELAY
1	0.99	0.331	0	0

Поле STORAGE определяет имя или номер объекта типа "многоканальное устройство".

Поле CAP. определяет емкость многоканального устройства, заданную оператором STORAGE.

Поле REMAIN определяет количество единиц свободной емкости многоканального устройства в конце периода моделирования.

Поле MIN определяет минимальное количество используемой емкости многоканального устройства за период моделирования.

Поле MAX определяет максимальное количество используемой емкости многоканального устройства за период моделирования.

Поле ENTRIES определяет количество входов в многоканальное устройство за период моделирования.

Поле AVL. определяет состояние готовности многоканального устройства в конце периода моделирования. 1 - означает, что многоканальное устройство готово, 0 - не готово.

Поле AVE.C определяет среднее значение занятой емкости за период моделирования.

Поле UTIL. определяет часть периода моделирования, в течение которого многоканальное устройство использовалось.

Поле RETRY определяет количество транзактов, ожидающих специальных условий, зависящих от состояния многоканального устройства.

Поле DELAY определяет количество транзактов, ожидающих возможности входа в блок ENTER.

Статистику о работе МКУ можно наблюдать в окне МКУ при работе с программой, перейдя в это окно с помощью клавиш [ALT+S].

10.2. Определение емкости МКУ

Все используемые в модели МКУ должны быть предварительно описаны, т.е. должно быть определено количество однотипных приборов, входящих в МКУ. Для этого используется оператор STORAGE.

Формат оператора задания емкости МКУ :

<i>Поле</i>	<i>Информация в поле</i>
Метка	Символическое имя МКУ
Операция	STORAGE
Операнд А	Емкость МКУ

Пример 10.1

Пусть моделируемая система включает в себя 8 механиков и 10 подъемных кранов, тогда в GPSS-модель могут быть введены следующие МКУ:

MECHANICS	STORAGE 8
CRANES	STORAGE 10

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

Пример [12] 10.2

Если нам нужно сравнить несколько комбинаций значений пары "количество механиков - количество кранов" за один этап моделирования, то для этого нужно использовать следующую последовательность команд START и операторов STORAGE:

MECHANICS	STORAGE <значение 1.1>
CRANES	STORAGE <значение 1.2>
<текст GPSS - программы>	
START	1
MECHANICS	STORAGE <значение 2.1>
CRANES	STORAGE <значение 2.2>
START	1
...	
MECHANICS	STORAGE <значение n.1>
CRANES	STORAGE <значение n.2>
START	1

11. Примеры построения GPSS-моделей

Пример [12] 11.1

Интервалы прихода клиентов в парикмахерскую с одним креслом распределены равномерно в интервале 18 ± 6 мин. Время стрижки также распределено равномерно, 16 ± 4 мин. Клиенты приходят в парикмахерскую, стригутся в порядке "первым пришел - первым обслужен" и затем уходят. Модель парикмахерской на GPSS должна обеспечить сбор статистических данных об очереди. Необходимо промоделировать работу парикмахерской в течение 8 ч модельного времени.

I. Построение модели

Порядок блоков в модели соответствует порядку фаз, в которых клиент оказывается при движении в реальной системе:

- клиент приходит;
- если необходимо, ждёт своей очереди;
- садится в кресло к парикмахеру;
- парикмахер стрижет клиента;
- клиент уходит из парикмахерской.

II. Таблица определений

Элементы GPSS	Интерпретация
Транзакты:	
1-й сегмент модели	Клиенты
2-й сегмент модели	Таймер
Приборы:	

MST	Парикмахер
Очереди: MSTQ	Очередь, используемая для сбора статистики об ожидании клиентов

Единица модельного времени - 1 мин.

III. Программа

*	<i>MODEL SEGMENT 1</i>		
	GENERATE	18,6	приход клиентов
	QUEUE	MSTQ	присоединение к очереди
	SEIZE	MST	переход в кресло парикмахера
	DEPART	MSTQ	уход из очереди
	ADVANCE	16,4	обслуживание у парикмахера
	RELEASE	MST	освобождение парикмахера
	TERMINATE	0	уход из парикмахерской
*	<i>MODEL SEGMENT 2</i>		
	GENERATE	480	транзакт-таймер приходит в момент 480
	TERMINATE	1	завершение прогона
	START	1	

Пример [12] 11.2

В парикмахерскую с одним креслом могут приходить клиенты двух типов. Клиенты первого типа желают только стричься. Распределение интервалов их прихода 35 ± 10 мин. Клиенты второго типа желают постричься и побриться. Распределение интервалов их прихода 60 ± 20 мин. Парикмахер обслуживает клиентов в порядке "первым пришел - первым обслужен". На стрижку уходит 18 ± 6 мин, а на бритье 10 ± 2 мин. Необходимо написать модель парикмахерской на GPSS, обеспечив сбор данных об очереди, образуемой ожидающими клиентами.

I. Построение модели

В отличие от модели примера 11.1 необходимо реализовать различные виды обслуживания: только стригущиеся и стригущиеся - бреющиеся клиенты.

Такую систему можно промоделировать с помощью двух сегментов. Один из них моделирует обслуживание только стригущихся клиентов, а второй - стригущихся и бреющихся. В каждом из сегментов пара QUEUE-DEPART должна описывать одну и ту же очередь. Таким же образом, пара блоков SEIZE-RELEASE

должна описывать в каждом из двух сегментов один и тот же прибор и моделировать работу парикмахера.

II. Таблица определений

Элементы GPSS	Интерпретация
Транзакты:	
1-й сегмент модели	Только стригущиеся клиенты
2-й сегмент модели	Стригущиеся и бреющиеся клиенты
3-й сегмент модели	Таймер
Приборы:	
MST	Парикмахер
Очереди:	
MSTQ	Очередь, используемая для сбора статистики об ожидании клиентов обоих типов

Единица модельного времени - 1 мин.

III. Программа

*	<i>CUTTING</i>			
	GENERATE	35,10		приход только стригущихся клиентов
	QUEUE	MSTQ		присоединение к очереди
	SEIZE	MST		переход в кресло парикмахера
	DEPART	MSTQ		уход из очереди
	ADVANCE	18,6		стрижка у парикмахера
	RELEASE	MSTQ		освобождение парикмахера
	TERMINATE	0		уход из парикмахерской
*	<i>CUTTING AND SHAVING</i>			
	GENERATE	60,20		приход стригущихся и бреющихся
	QUEUE	MSTQ		присоединение к очереди
	SEIZE	MST		переход в кресло парикмахера
	DEPART	MSTQ		уход из очереди
	ADVANCE	10,2		бритье у парикмахера
	ADVANCE	18,6		стрижка у парикмахера
	RELEASE	MST		освобождение парикмахера
	TERMINATE	0		уход из парикмахерской
*	<i>TIMER</i>			(сегмент таймера)
	GENERATE	480		транзакт - таймер приходит в момент
480				
	TERMINATE	1		завершение прогона
	START	1		

Пример 11.3

На некоторой фабрике в кладовой работает один кладовщик. Он выдает запасные части механикам, обслуживающим станки и устанавливающим эти части на испорченных станках. Запасные части довольно дороги и, кроме того, их ассортимент слишком велик для того, чтобы каждый механик мог иметь запасную часть в своем ящике. Время, необходимое для удовлетворения запроса, зависит от типа запасной части. Запросы бывают 2-х категорий. Соответствующие данные приведены в таблице 11.1.

Таблица 11.1

<i>Категория запроса</i>	<i>Интервалы времени прихода механиков, с</i>	<i>Время обслуживания, с</i>
1	420±360	300±90
2	360±240	100±30

Структурная схема модели изображена на рис. 11.1.

Порядок обслуживания механиков кладовщиком таков: запросы категории 1 обслуживаются только в том случае, когда в очереди нет ни одного запроса категории 2. Внутри одной категории дисциплина обслуживания - "первый пришел - первый обслужен". Необходимо создать модель работы кладовой, моделирование выполнять в течение 8-часового рабочего дня.

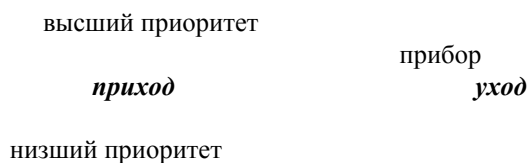


Рис 11.1

Дисциплина обслуживания "первый пришел - первым обслужен" при наличии двух категорий запросов с приоритетом.

I. Построение модели

Эта система очень похожа на систему из примера 11.2. Есть два различных типа заявок, поступающих на обслуживание к единственному прибору. Различаются распределения интервалов приходов и времени обслуживания для этих типов заявок. Отличие заключается в том, что один из типов имеет

преимущество в обслуживании. Построить модель можно, используя тот же подход, что и в примере 11.2. Но при этом надо использовать средство задания различных приоритетов заявкам.

Итак, запросы категории 1 будем моделировать одним сегментом, а запросы категории 2 - другим. Разные относительные приоритеты задаются путем использования в качестве операнда E блока GENERATE для запросов категории 2 большего значения, чем для запросов категории 1.

II. *Таблица определений*

<i>Элементы GPSS</i>	<i>Интерпретация</i>
Транзакты: 1-й сегмент модели 2-й сегмент модели 3-й сегмент модели	Механики, делающие запрос категории 1 Механики, делающие запрос категории 2 Таймер
Приборы: STOCKMAN	Кладовщик
Очереди: QSTCKM1 QSTCKM2	Очереди, используемые для сбора статистики о механиках, делающих запросы 1-й и 2-й категорий

Единица модельного времени - 1 с.

III. *Программа*

* *MECHANICS OF TYPE 1*

GENERATE	420,360,,,1	приход механиков первой категории
QUEUE	QSTCKM1	присоединение к очереди 1-й кат.
SEIZE	STOCKMAN	занятие кладовщика
DEPART	QSTCKM1	уход из очереди
ADVANCE	300,90	обслуживание кладовщиком
RELEASE	STOCKMAN	освобождение кладовщика
TERMINATE	0	уход из кладовой

* *MECHANICS OF TYPE 2*

GENERATE	360,240,,,2	приход механиков второй категории
QUEUE	QSTCKM2	присоединение к очереди 2-й кат.
SEIZE	STOCKMAN	занятие кладовщика

DEPART	QSTCKM2	уход из очереди
ADVANCE	100,30	обслуживание кладовщиком
RELEASE	STOCKMAN	освобождение кладовщика
TERMINATE	0	уход из кладовой
*	<i>TIMER</i>	(сегмент таймера)
GENERATE	28800	транзакт-таймер приходит в момент
TERMINATE	1	завершение прогона
START	1	

Пример [12] 11.4

Производство деталей определенного вида включает длительный процесс сборки, заканчивающийся коротким периодом обжига в печи. Поскольку содержание печи обходится очень дорого, несколько сборщиков используют одну печь, в которой одновременно можно обжигать только одну деталь. Сборщик не может начать новую сборку, пока не вытащит из печи предыдущую деталь.

Таким образом, сборщик работает в таком режиме:

1. Собирает следующую деталь.
2. Ожидает возможности использования печи по принципу "первым пришел - первым обслужен".
3. Использует печь.
4. Возвращается к п. 1.

Время, необходимое на различные операции, стоимость операций и изделий представлены в таблицах 11.2 и 11.3.

Таблица 11.2

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

Таблица 11.3

Стоимостные характеристики	
Зарплата сборщика	3.75 ед. стоимости в час
Стоимость печи	80 ед. ст. за восьмичасовой рабочий день (независимо от степени использования)
Цена материала	2 ед. ст. на одну деталь

Стоимость готового изделия	7 ед. ст. за деталь
----------------------------	---------------------

Необходимо построить на GPSS модель описанного процесса. Имея эту модель, следует определить оптимальное количество сборщиков, использующих одну печь. Под оптимальным понимают такое количество, которое дает оптимальную прибыль при моделировании в течение 40 ч модельного времени. Предполагается, что в течение рабочего дня нет перерывов, а рабочие дни идут подряд без выходных дней.

I. *Построение модели*

В данном случае есть два вида условий: имеется лишь одна печь; существует некоторое фиксированное количество сборщиков, работающих в системе. Естественно для моделирования печи использовать понятие "прибор", также естественно как отождествлять сборщиков с транзактами. Тогда можно считать, что сборщики "циркулируют" в системе, периодически осуществляя сборку и обжиг, аналогично транзакты должны циркулировать в GPSS - модели.

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

Для вычисления прибыли, при заданном числе сборщиков, необходимо знать, сколько готовых деталей они сделают в течение моделируемого периода. Количество выходов из печи в точности совпадает с этим значением, т.е. в процессе моделирования нас интересует только эта величина.

II. *Таблица определений*

<i>Элементы GPSS</i>	<i>Интерпретация</i>
Транзакты: 1-й сегмент модели 2-й сегмент модели	Сборщики Таймер
Приборы: PECH	Печь

Единица модельного времени - 1 мин.

III. *Программа*

*

BACK1	GENERATE	,,,4	Определение количества сборщиков
	ADVANCE	30,5	Сборка следующей детали
	SAIZE	1	Занятие печи
	ADVANCE	8,2	Использование печи
	RELEASE	1	Освобождение печи
	TRANSFER	,BACK1	Переход к сборке следующей детали

*

GENERATE	2400	Моделирование 5 дней работы
TERMINATE	1	Завершение работы

*

START	1	Начать прогон модели
--------------	----------	----------------------

III. *Использование модели*

IV.1. Поведение модели исследовали при трех значениях **количества сборщиков**. При каждом новом значении менялся только операнд D блока GENERATE 1-го сегмента модели.

IV.2. Задавшись некоторым значением доверительной вероятности, было определено количество прогонов N, которое нужно выполнить для каждого значения количества сборщиков.

V. *Таблица результатов моделирования*

<i>Количество сборщиков</i>	D=4	D=5	D=6
<i>Среднее (по количеству прогонов N) число выпущенных деталей</i>	236	279	294

Стоимость использования печи в течение 40-часового периода моделирования равна $80 * 5 = 400$ (единиц стоимости). Каждый сборщик зарабатывает за одну неделю $3,75 * 40 = 150$ (единиц стоимости).

Значения прибыли для каждого из трех случаев приведены в таблице 11.4.

Таблица 11.4

<i>Количество сборщиков</i>	D=4	D=5	D=6
<i>Среднее значение прибыли</i>	175	245	170

Таким образом, в данной системе оптимальное количество сборщиков составляет 5 человек.

Пример [12] 11.5

Корабли двух типов прибывают в порт, где происходит их разгрузка. В порту имеются 2 буксира, обеспечивающих ввод и вывод кораблей из порта. Первый тип кораблей включает корабли малой тоннажности и требует при входе и выходе использования одного буксира. Второй тип кораблей имеет большой размер, и для их ввода и вывода из порта требуется два буксира. Из-за различия размеров двух типов кораблей необходимы причалы различного размера, кроме того, корабли имеют разное время погрузки-разгрузки. Исходные данные приведены в таблице 11.5.

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

Таблица 11.5

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

GPSS - программа

PRCH1 **STORAGE 6** 6 причалов для кораблей 1-го типа
PRCH2 **STORAGE 3** 3 причала для кораблей 2-го типа
BUKS **STORAGE 2** 2 буксира

*

SHIPS OF TYPE 1

*

GENERATE **130,30**
QUEUE **TYPE1** подход к порту
ENTER **PRCH1** получить причал
ENTER **BUKS** получить буксир
DEPART **TYPE1** выход из порта
ADVANCE **30,7** буксировка (причаливание)
LEAVE **BUKS** освобождение буксира

ADVANCE	720,120	погрузка-разгрузка
ENTER	BUKS	получить буксир
LEAVE	PRCH1	
ADVANCE	20,5	буксировка (отчаливание)
LEAVE	BUKS	освобождение буксира
TERMINATE		

*

*

*

SHIPS OF TYPE 2

GENERATE	390,60
QUEUE	TYPE2
ENTER	PRCH2
ENTER	BUKS,2
DEPART	TYPE2
ADVANCE	45,12
LEAVE	BUKS,2
ADVANCE	1080,240
ENTER	BUKS,2
LEAVE	PRCH2
ADVANCE	35,10
LEAVE	BUKS,2
TERMINATE	0

*

GENERATE	48000
TERMINATE	1
START	1

Среднее время ожидания кораблями каждого типа входа в порт получаем в конце моделирования из стандартной статистики по очередям: оно равно показателю AVERAGE TIME соответствующей очереди. Это же значение дают такие стандартные числовые атрибуты, как QT\$TYPE1 и QT\$TYPE2, подробнее об этом см. разд. 14.

12. Переменные

1.1. *Общая характеристика переменных*

При построении модели системы, пользователю возможно понадобится задать сложные логические или математические соотношения между атрибутами системы. Для этой цели в программе используются *переменные*.

В GPSS/PC имеется три типа переменных:

- арифметические переменные;

- арифметические переменные с "плавающей точкой";
- булевы переменные.

Значение арифметических переменных может использоваться таким образом:

1. Как операнд блока. В этом случае значение арифметической переменной может представлять собой:
 - номер объекта (прибора, МКУ, очереди и т.п.);
 - номер параметра транзакта;
 - значение стандартного числового атрибута;
2. Как операнд А функции.
3. Как операнд А таблицы (см. разд. 19).
4. Как операнд другой арифметической переменной или булевой переменной.

В выражениях арифметических переменных используются арифметические операции:

- + алгебраическое сложение;
- алгебраическое вычитание;
- # алгебраическое умножение;
- / алгебраическое деление (результатом операции является целая часть частного);
- @ - деление по модулю;
- ^ возведение в степень;
- \ деление нацело (перед делением у обоих операндов отбрасываются дробные части, результатом операции является целая часть частного).

В выражении переменной может быть задано произвольное число перечисленных выше операций в различных сочетаниях. Знак результата вычислений определяется по обычным алгебраическим правилам. Допустимы отрицательные значения переменных. Выражения переменных анализируются слева направо. Возведение в степень, умножение, деление и деление по модулю выполняются раньше, чем сложение и вычитание.

Значение переменной является её стандартным числовым атрибутом.

1.1 . *Арифметические переменные*

Арифметические переменные представляют собой арифметические сочетания значений различных СЧА, в том числе и других арифметических переменных аналогично арифметическим выражениям в алгоритмических языках. Переменная задается одним оператором, называемым оператором описания переменной (оператором VARIABLE), который содержит арифметическое выражение.

Значение переменной V<номер переменной> (V\$<имя переменной>) является её **стандартным числовым атрибутом**.

Формат оператора описания переменной:

<i>Поле</i>	<i>Информация, задаваемая в поле</i>
<i>Метка</i>	<i>Имя (числовое или символическое) переменной</i>
<i>Операция</i>	VARIABLE
Операнды: A	выражение, которое используется для вычисления значения переменной

Пример 12.1

Оператор описания VARIABLE определяет арифметическую переменную RSLTV:

RSLTV VARIABLE Q\$WAITL+3-P7#FN\$DSTRB

При любом обращении к переменной RSLTV (употребляется обозначение V\$RSLTV) её значение вычисляется как текущая длина очереди WAITL плюс константа 3, минус произведение значения параметра 7, обрабатываемого в данный момент времени транзакта, на значение функции DSTRB. СЧА описаны в разделе 14.

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

Любые стандартные числовые атрибуты, функции и другие арифметические переменные могут быть использованы в выражении арифметической переменной. Запрещается только использование самой вычисляемой переменной. Нельзя использовать непосредственно переменные со знаком, так как знаки в этом случае рассматриваются как знаки арифметических операций.

GPSS/PC допускает использование скобок в выражениях арифметических переменных. Скобки могут быть использованы для группировки членов или для обозначения операции умножения. Выражение переменной может содержать не более пяти пар скобок (не считая скобок, используемых при описании матриц ячеек). В выражении, содержащем скобки, прежде всего вычисляется значение группы элементов, ограниченных самой правой из левых скобок.

Пробелы между символами в выражении не допустимы. Левый пробел записи считается концом выражения. Для записи выражения, превышающего длину строки, можно ввести второй оператор VARIABLE с именем, отличным от имени первой переменной, и включить значение новой переменной в качестве одного из операндов в выражении первой арифметической переменной.

Пример 12.2

VADD VARIABLE P10+25

При обращении к арифметической переменной VADD её значение вычисляется как сумма значений параметра 10 сообщения, обрабатываемого в данный момент, и константы 25.

VFIX1	VARIABLE	Q9+3#V\$VFIX2-V\$VFIX3^FNSIO
VFIX2	VARIABLE	9+R13-FN19#Q10
VFIX3	VARIABLE	

FNSTYPE1+S\$POOL#R\$RESCS-Q\$ENTRY

VFIX1 содержит в качестве элементов переменные VFIX2 и VFIX3.

WER1	VARIABLE	V*P*FN*P2
-------------	-----------------	------------------

Переменная WER1 будет вычислена следующим образом: вначале находится значение функции с номером, который равен содержимому параметра P2; затем определяется значение параметра с номером, равным значению этой функции (пусть содержимым этого параметра будет число *n*); переменной WER1 присваивается значение переменной *n*.

COMP	VARIABLE	X*P4/100
-------------	-----------------	-----------------

При обращении к арифметической переменной COMP значение этой переменной вычисляется как частное от деления значения, записанного в сохраняемой величине, номер которой определяется параметром 4, обрабатываемого в данный момент транзакта, на константу 100. Такой прием используется для выделения определенных старших разрядов числа, находящегося в сохраняемой величине.

Например, пусть в параметре 4 обрабатываемого в данный момент транзакта, записано число 125. Предположим, что в ячейке 125 хранится некоторое значение 12345. При обращении к переменной COMP происходит следующее:

- определяется значение параметра 4, получается число 125;
- определяется значение 12345, хранимое в ячейке X*P4, т.е. в ячейке 125;
- значение 12345 делится на 100 (с отбрасыванием остатка), в результате получается 123.

1.1. *Арифметические переменные с плавающей точкой*

Арифметические переменные с плавающей точкой аналогичны арифметическим переменным, за исключением того, что все операции над операндами выражений переменных с плавающей точкой выполняются без преобразования операндов и промежуточных результатов к целому виду. Лишь окончательный результат вычисления преобразуется к целому числу.

Формат операторов описания арифметических переменных с плавающей точкой идентичен рассмотренному выше формату операторов описания арифметических переменных, за исключением того, что в поле операции

записывается слово **FVARIABLE**. Правила написания операторов те же, что и для арифметических переменных. Арифметическая переменная и переменная с плавающей точкой не могут иметь одинаковые номера. Если они имеют одинаковые номера, то при вычислении используется более позднее из двух описаний.

Различие результатов, полученных при вычислении переменных с плавающей и фиксированной точками, можно увидеть из приведенного ниже примера 12.3. Значение переменной **FLOAT** будет равно 36, поскольку константа 10 будет умножена на 3.67, и от результата 36.7 будет взята целая часть. Переменная **FIXED** равна 30, поскольку результат промежуточной операции деления будет округлен до 3.

Пример 12.3

FLOAT	FVARIABLE	10#(11/3)
FIXED	VARIABLE	10#(11/3)

!

1. Для переменных с плавающей точкой недопустима операция деления по модулю. Только при описании переменных с плавающей точкой допускается применение дробных констант.
2. Стандартный числовой атрибут **V\$<имя переменной>** используется для обращения к значениям как арифметических переменных, так и к значениям переменных с плавающей точкой. Способ вычисления переменной **V** определяется оператором описания этой переменной.

13. Определение функции в GPSS

В GPSS рассматриваются пять типов функций:

- дискретная числовая (**D**),
- непрерывная числовая (**C**),
- табличная числовая (**L**),
- дискретная атрибутивная (**E**),
- табличная атрибутивная (**M**).

Рассматриваются два первых типа.

Дискретная функция представляет собой кусочно-постоянную функцию, а **непрерывная** - кусочно-непрерывную, то есть дискретная функция состоит из горизонтальных ступеней (рис. 13.1). Непрерывная функция в GPSS состоит из прямых отрезков, представляя собой ломаную линию (рис. 13.2). Чтобы задать дискретную функцию,

необходимо задать координаты правых точек горизонтальных полочек, а для непрерывной функции - координаты всех точек, которые являются концами отрезков.

Y

Y

X
Рис 13.1

X
Рис.13.2

Для определения дискретной и непрерывной GPSS - функции необходимо:

1. Присвоить функции имя. Имя может быть числовым и символическим.
2. Задать аргумент функции. Аргументом является источник случайных чисел, используемый для розыгрыша в соответствии с распределением, заданным функцией, стандартный числовой атрибут (см. разд.13, 15-17) или любая другая функция. В первом случае аргумент задается в виде RN_j , $j=1,...,7$. В GPSS возможно обращение к 7 идентичным генераторам случайных чисел. При работе с функциями генераторы выдают случайные числа в диапазоне 0 – .999999.

1. Задать тип функции и число различных значений, которые может принимать случайная переменная.

1. Задать значения переменной (аргумента) и соответствующие значения функции.

Три первые элемента информации указываются в *операторе определения функции*.

Его формат:

Поле	Информация, задаваемая в поле
Метка	Имя (числовое или символическое) функции
Операция	FUNCTION
Операнды:	
А	RN_j ($j=1,...,7$), СЧА.
В	Dn либо Cn ,

	где D определяет дискретную функцию, C определяет непрерывную функцию; n для дискретной функции - это число различных значений, получаемых функцией (количество горизонтальных полочек), для непрерывной функции - это число на единицу больше числа отрезков, составляющих функцию (количество “крайних” точек)
--	---

За каждым оператором описания FUNCTION следуют операторы для задания координат точек функции (значений аргументов X_i и соответствующих им значений функции Y_i). Это **операторы описания координат функции**. Вид задания этих операторов зависит количество точек координат функции:

1. Координаты всех точек помещаются в одной строке:

$$X_1, Y_1 / X_2, Y_2 / \dots / X_n, Y_n$$

2. Координаты всех точек расположены в нескольких строках:

$$X_1, Y_1 / X_2, Y_2 / \dots / X_i, Y_i$$

$$X_{i+1}, Y_{i+1} / X_{i+2}, Y_{i+2} / \dots / X_k, Y_k$$

$$\dots \quad \dots$$

$$X_{k+1}, Y_{k+1} / X_{k+2}, Y_{k+2} / \dots / X_n, Y_n$$

X_i и Y_i являются координатами i -й точки функции. В случае моделирования случайной величины значение X_i являются i -й суммарной (кумулятивной) частотой, а Y_i -соответствующим значением моделируемой случайной величины.

Особенности оператора описания координат функции:

- * Основной единицей информации этого оператора является пара значений X_i, Y_i координат точки i функции.
- * значения координат X_i и Y_i одной точки функции разделяются запятой.
- * Последовательные наборы координат разделяются знаком "/".
- * Координаты X_i и Y_i , относящиеся к одной точке, должны задаваться одним оператором, т.е. пара координат, начинающаяся в некоторой строке должна быть на ней же и завершена.
- * Все строки описания координат функции должны начинаться с 1-й позиции.
- * Между оператором определения функции и операторами описания координат функции комментарии не допустимы.
- Во всех случаях значения аргумента должны удовлетворять следующим неравенствам:

$$X_1 < X_2 < \dots < X_i < \dots < X_n.$$

Значение функции является ее стандартным числовым атрибутом.
 Способ ссылки на это атрибут зависит от того, как задано имя функции: в символьном или числовом виде. Если имя числовое, то к значению функции обращаемся через FN j, где j - номер функции. Если имя символическое, то к значению функции обращаемся через FN\$<имя функции>.

!

1. В качестве аргумента функции может быть использовано и значение любой другой функции. Каждая функция должна иметь, по крайней мере, две описанные точки X_i, Y_i .

Пример 13.1

Пусть необходимо разыгрывать следующую дискретную случайную переменную, заданную в таблице 13.1.

Таблица 13.1

<i>Значение случайной переменной</i>	<i>Относительная частота</i>	<i>Суммарная частота</i>	<i>Диапазон</i>	<i>Интервал</i>
2	0,15	0,15	[0.0 - 0.15]	1
5	0,20	0,35]0.15 - 0.35]	2
8	0,25	0,60]0.35 - 0.60]	3
9	0,22	0,82]0.60 - 0.82]	4
12	0,18	1,00]0.82 - 1.0]	5

Графическая интерпретация функции представлена на рис. 13.3.

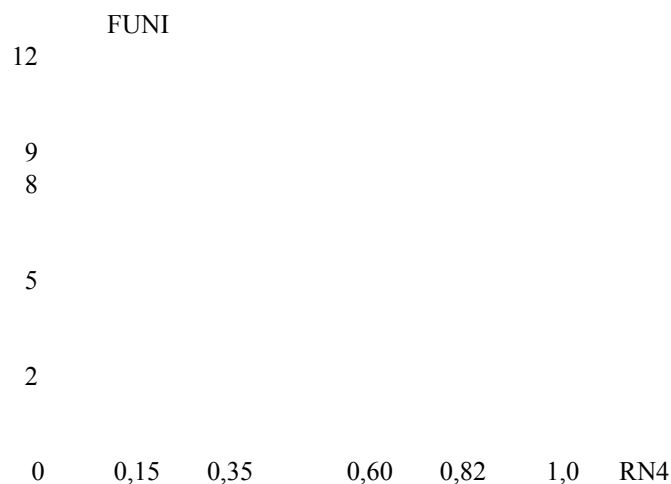


Рис. 13.3

Тогда можно определить GPSS-функцию таким образом:

FUNIFUNCTION RN4,D5
.15,2/.35,5/.6,8/.82,9/1,12

13.1. Особенности вычисления дискретных и непрерывных GPSS-функций

В начальной фазе вычисления дискретной и непрерывной функций одинаковы. При обращении к функции определяется значение аргумента функции. Затем просматривается упорядоченный ряд значений $X_1 < X_2 < \dots < X_i < \dots < X_n$ для определения интервала, в который попало значение аргумента (пусть это будет интервал между точками $i-1$ и i).

Если функция дискретная: второй элемент соответствующей пары X_i, Y_i является значением функции.

Если функция непрерывная: выполняется линейная интерполяция для пары точек $i-1$ и i , находящихся по краям интервала значений функции, на который указало значение аргумента. **Целая часть** результата интерполяции и является значением функции.

Если значение аргумента функции больше значения координаты X_n последней точки, то в обоих случаях в качестве значения функции берется значение Y_n .

Пример 13.2

Моделирование случайной переменной, равномерно распределенной на интервале $[2,6]$.

INN FUNCTION RN2,C2
0,2/1,6

INN

6

5

4

3

2

1

0

1,0 RN2

Рис. 13.4

Фактические значения интервалов времени распределены равномерно и получают одно из значений 2,3,4,5. Так как генератор случайных чисел выдает

максимальное значение 0.9999, функция INN примет значение 5.9999, т.е. целая часть будет равна 5. В случае, если генератор случайных чисел RN2 выдаст значение, представленное на рис.13.4, функция INN примет значение 3.

Точные граничные значения RN2, соответствующие возможным значениям функции представлены в таблице 13.2.

Таблица 13.2

Целая часть значения функции	Диапазон значений RN2
2	[0,0 - 0,25]
3]0,25 - 0,50]
4]0,50 - 0,75]
5]0,75 - 0,999999]

Отметим, что равномерное распределение [2,3,4,5] не может быть задано непосредственно с помощью операндов A и B блока GENERATE. Здесь имеем 4 возможных значения, тогда как интервал $A \leq B$ (A и B целые) всегда имеет нечетное число элементов.

13.2. Моделирование неравномерных случайных величин. Использование функций в блоках GENERATE и ADVANCE

Пусть распределение интервалов поступления через определенный блок GENERATE или время задержки в некотором блоке ADVANCE **не является равномерным**, либо является равномерным с "плавающими во времени", т.е. не фиксированными, значениями среднего и половины поля допуска. Для входов транзактов в модель через такой блок GENERATE или задания закона для времени задержки в соответствующем блоке ADVANCE необходимо использовать функции и/или СЧА.

Использование функций, заданных в операндах блоков, зависит от контекста. От значения функции берется целая часть, за исключением тех случаев, когда это значение используется в качестве операнда B блоков GENERATE, ADVANCE или операнда C блока ASSIGN. В таблице 13.3 показаны различные варианты использования функций и СЧА в качестве операндов A и B

блоков GENERATE и ADVANCE. Под результатом понимается время интервала поступления или задержки транзактов.

Таблица 13.3

	<i>Операнд A</i>	<i>Операнд B</i>	<i>Результат</i>
1	α (число или СЧА)	β (число или СЧА)	Генерируется случайное число, равномерно распределенное на интервале α ± β . Результат равен полученному числу.
2	FN\$A	отсутствует	Результат равен значению функции A
3	Отсутствует	FN\$B	Данное сочетание недопустимо
4	FN\$A	β (число или СЧА)	Вначале вычисляется значение функции A, берется целая часть этого значения (пусть это будет число α), после чего генерируется случайное число, равномерно распределенное на интервале α ± β . Результат равен полученному числу.
5	α (число или СЧА)	FN\$B	Вначале вычисляется значение функции B (пусть это будет число β), после чего находится произведение α × β . Результат равен целой части этого произведения.
6	FN\$A	FN\$B	Вычисляются значения функций A и B (пусть это будет числа α и β), после чего находится произведение α × β . Результат равен целой части этого произведения.

Пример 13.3

FF FUNCTION RN3,D5
.05,5/.15,6/.75,8/.93,9/1,10

К этой функции можно обратиться так:

GENERATE FN\$FF
...

ADVANCE FNSFF

Пример 13.4

Пусть в моделируемой системе время обслуживания некоторым прибором распределено равномерно на интервале $A \pm 2$, где среднее время обслуживания A с вероятностью 0.4 принимает значение 5, а с вероятностью 0.6 - значение 7. Эту ситуацию можно промоделировать, определив функцию AverageT так:

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

Используем ее в блоке ADVANCE:

ADVANCE FNSAverageT,2

Выполнение подпрограммы блока ADVANCE включает расчет функции AverageT. Это требует обращения к генератору случайных чисел. RN1. Пусть генератор выдал значение меньшее, чем 0.4. Тогда соответствующее значение функции AverageT равно 5. Таким образом, время задержки в приборе текущего транзакта равномерно распределено на интервале 5 ± 2 .

1.1 . *Непрерывные случайные переменные, рассматриваемые как дискретные*

Как известно, дискретные случайные переменные могут иметь только фиксированное число значений. В противоположность этому, непрерывные (в классическом смысле этого термина) случайные переменные могут иметь неограниченное число различных значений.

В действительности, обычно достаточно, чтобы все случайные переменные имели конечное число конкретных значений. Нет необходимости в тщательном определении значений этих случайных переменных за исключением случаев, когда необходимо сделать расчет с высокой степенью точности. Таким образом, вполне возможна дискретизация непрерывных распределений. После дискретизации они могут быть определены в GPSS с помощью дискретных и непрерывных GPSS-функций (непрерывные GPSS-функции по существу также являются дискретными, так как множество значений их дискретно и конечно).

13.4. Функции распределения случайных величин

В языке GPSS возможность задания функций распределения случайных величин ограничена заданием их в табличном виде путем аппроксимации непрерывными функциями. Поэтому можно задать только те функции, которые легко можно преобразовать для новых значений параметров. К таким функциям относятся функции экспоненциального распределения с параметром $\lambda = 1$, а также стандартного нормального распределения с математическим ожиданием $m = 0$ и стандартным отклонением $\sigma = 1$. Эти функции соответственно задаются 24 и 25 точками.

13.4.1. Моделирование пуассоновского потока

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

где λ - интенсивность потока.

Интервалы времени между соседними требованиями пуассоновского потока распределены по экспоненциальному закону. Согласно *методу обратной функции*, можно получить ряд чисел, имеющих экспоненциальное распределение, если ряд случайных чисел, равномерно распределенных на интервале $[0 \div 1]$, подвергнуть преобразованию, обратному экспоненциальной функции распределения:

где $-j$ - j -й разыгранный интервал времени поступления, \bar{r} - средний интервал времени поступления (равен $1 / \lambda$), r_j - j -е число последовательности случайных чисел, равномерно распределенных в интервале $[0 \div 1]$.

Разработчиками GPSS была осуществлена аппроксимация функции, обратной экспоненциальной функции распределения. В результате функция заменена 23 отрезками, используемыми для преобразования значений RN_j в значение $-\ln(1 - RN_j)$.

Приведенная ниже функция XPDIS определяет пуассоновский поток с интенсивностью равной 1.

```

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

```

Пуассоновский входящий поток с интенсивностью λ , отличной от 1, моделируется с помощью блока GENERATE:

!

1. В качестве операнда A используют среднее значение интервалов времени ($T=1/\lambda$, λ - интенсивность пуассоновского потока). В качестве операнда B используют СЧА - значение функции FN\$XPDIS (строки определения и значения которой представлены выше).

Пример 13.5

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

GENERATE 120, FN\$XPDIS.

Свойство ординарности пуассоновского потока гласит, что вероятность равна нулю поступления двух или более требований в течение малого временного интервала. Пусть пуассоновский поток моделируется блоком

GENERATE 5, FN\$XPDIS

Если в результате обращения к функции XPDIS получили значение меньшее, чем $1/5$, то целая часть произведения равна 5 и XPDIS равна 0. Отсюда следует нарушение указанного свойства. Во избежание этого рекомендуется, чтобы операнд A в блоке GENERATE был **больше 50**. Это легко достигается путем варьирования единицей модельного времени.

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

ADVANCE 345, FN\$XPDIS

13.4.2. Моделирование эрланговского потока

Экспоненциальная функция распределения может также использоваться для моделирования распределения Эрланга. Например поток Эрланга третьего порядка со средним значением времени поступления 120 можно задать таким образом:

GENERATE 40, FN\$XPDIS
ADVANCE 40, FN\$XPDIS
ADVANCE 40, FN\$XPDIS

13.4.3 Моделирование гипер – и гипозэкспоненциальных распределений

Экспоненциальная функция распределения может быть использована также для моделирования гипер – и гипозэкспоненциальных распределений.

Не экспоненциальное распределение с коэффициентом вариации $C > 1$ можно аппроксимировать с помощью взвешенной суммы экспонент (гиперэкспоненциальное распределение):

$C=1$, если $\mu_i := \mu$ для всех i , т.е. распределение экспоненциальное.

Гиперэкспоненциальное распределение можно получить при параллельном соединении k экспоненциальных обслуживающих приборов с интенсивностью обслуживания μ_i и вероятностью быть использованным для обслуживания,

равной ω_i (рис.13.5). Причем в произвольный момент времени может быть занято не более одного прибора из k .

Рис. 13.5

Гипоэкспоненциальное распределение с $C < 1$ можно получить, например, из выражения:

Это можно получить последовательным соединением обслуживающих экспоненциальных приборов, причем в любой момент времени должно быть занято не более одного прибора (рис. 13.6).

Рис. 13.6

При равенстве всех коэффициентов μ , распределение обслуживающего центра будет k-распределение Эрланга

Коэффициент вариации C – это отношение стандартного отклонения к среднему значению.

Например, для моделирования гиперэкспоненциального распределения со средним значением 6,28 и стандартным отклонением 8,4 необходимо выполнить следующее:

HYPER FVARIABLE (410 + ((RN2 'L' 234) # (1334 – 410))) # FN\$XPDIS

Тогда эту переменную можно, например, использовать в блоке задержки:

ADVANCE V\$HYPER

13.4.4. Моделирование нормального закона распределения

Функция стандартного нормального закона распределения ($m = 0$, $\sigma = 1$) задается в GPSS/PC 24 отрезками следующим образом:

NOR FUNCTION RN1,C25

0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2
 .06681,-1.5/.11507,-1.2/.15866,-1/.21186,-.8/.27425,-.6
 .34458,-.4/.42074,-.2/.5,0/.57926,.2/.65542,.4
 .72575,.6/.78814,.8/.84134,1/.88493,1.2/.93319,1.5
 .97725,2/.99379,2.5/.99865,3/.99997,4/1,5

Для того, чтобы получить функцию нормального распределения с математическим ожиданием m_x и среднеквадратичным отклонением σ_x необходимо произвести вычисления по формуле:

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

где Z - функция стандартного нормального закона распределения.

В GPSS такой пересчет, например, для функции с $m_x = 60$ и $\sigma_x = 10$, осуществляется следующим образом:

NOR1 FVARIABLE 60+10#FN\$NOR

Если необходимо осуществить задержку для этого закона распределения, то использует блок:

ADVANCE V\$NOR1

При использовании функции нормального распределения для блоков GENERATE и ADVANCE необходимо обеспечить положительные значения разыгрываемых времен задержки. Это можно сделать, если $m_x \geq 5 \sigma_x$.

13.4.5. Моделирование других законов распределений

Все другие виды функций распределений случайных величин в GPSS необходимо задавать табличным способом для конкретных значений параметров этих функций. Для этого можно использовать специальные программы, позволяющие численным образом вычислять необходимое значение числа отрезков аппроксимации этих функций. Пример меню такой программы представлен на рис. 13.7.

Функция гамма – распределения для параметров (рис. 13.7) будет задана так:

GAMMA FUNCTION RN1,C26
 0,0/0.24217,0.29471/0.33719,0.89451/0.40669,0.88412
 0.46254,1.17882/0.50946,1.47353/0.5499,1.76824/0.58536,2.06294
 0.64503,2.65235/0.69351,3.24176/0.73373,3.83118/0.76758,4.42059
 0.79637,5.01/0.83207,5.89412/0.86078,6.77824/0.88409,7.66235
 0.90874,8.84118/0.9278,10.02/0.94583,11.49353/0.96139,13.26176
 0.97381,15.32471/0.98307,17.68235/0.99062,20.92412/0.99575,25.34471
 0.99878,32.41765/0.99992,48.33176/1,50.1

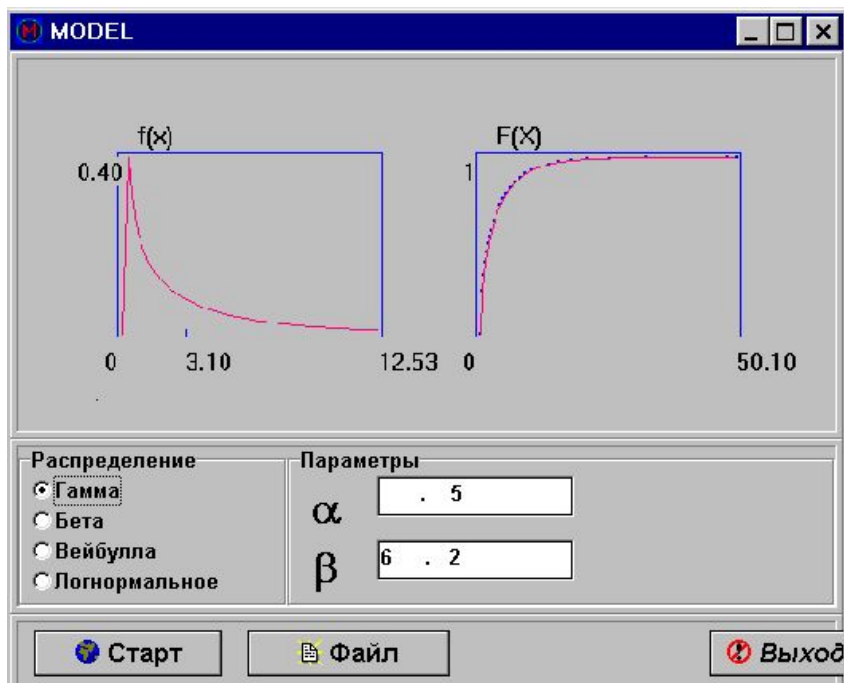


Рис. 13.7

14. Стандартные числовые атрибуты и параметры транзактов

В процессе моделирования интерпретатор автоматически регистрирует и корректирует информацию, касающуюся различных элементов, используемых в модели. Большая часть информации доступна только интерпретатору. Однако к некоторым атрибутам (свойствам) объектов может обращаться и программист, управляя процессом моделирования в зависимости от их значений.

Рассмотрим несколько примеров зависимости функционирования элементов модели от системных атрибутов:

1. Интенсивность работы некоторого прибора, зависит от длины очереди. Для определения времени обслуживания при каждом поступлении транзакта на обслуживание необходимо знать значение системного атрибута, как длина очереди.
2. Интенсивность обслуживания некоторого прибора зависит от общей продолжительности его функционирования (проявление усталости - интенсивность со временем падает, разогрев прибора - интенсивность со временем увеличивается). Время обслуживания - функция, зависящая от интервала времени, прошедшего от начала работы.
3. Имеется два прибора и диспетчер, распределяющий работы между ними таким образом, чтобы нагрузка на приборы была равномерной. Для этого в точке диспетчеризации необходимо иметь информацию о коэффициентах

нагрузки приборов и выбор пути продвижения транзакта должен зависеть от этих двух величин.

Условно все атрибуты можно разделить на 2 категории:

- атрибуты системы;
- атрибуты транзактов.

Атрибуты системы - это параметры, которые описывают состояние модели. Такие количественные показатели, как "текущее содержимое очереди" или "коэффициент нагрузки прибора", являются типичными системными атрибутами. Стандартный набор атрибутов, подобных указанному, автоматически поддерживается интерпретатором GPSS. Транзакты также могут иметь некоторые числовые характеристики. Одной из них является уровень приоритета. Кроме того, транзакт снабжается некоторым числом параметров. В GPSS все подобные атрибуты называются **стандартными числовыми атрибутами (СЧА)**.

Каждый объект GPSS имеет свой набор СЧА (свойства объекта). Доступ к СЧА осуществляется при использовании специальных наименований этих атрибутов. Имя СЧА состоит из двух частей:

- **групповое имя** (состоит из одной-двух букв, идентифицирует тип объекта и тип информации о нем);

- **имя конкретного члена группы** (т.е. имя прибора, многоканального устройства, очереди и т.д.).

Объекты могут идентифицироваться с помощью числовых и символьных имен. Если объект идентифицируется с помощью номера, то ссылка на его стандартный числовой атрибут записывается как СЧА_j, где j - номер объекта (целое число). При символической идентификации объекта ссылка на его стандартный числовой атрибут записывается как СЧА\$<имя объекта> (в данной записи под "СЧА" понимается групповое имя).

1.1. **Стандартные числовые атрибуты**

СЧА представляют свойства объектов. В таблицах 14.1, 14.2, 14.3 приведены различные СЧА для таких объектов GPSS, как приборы, многоканальные устройства и очереди.

Таблица 14.1 СЧА приборов

Наименование	Значение
Fj или F\$имя	Показатель занятости прибора (0, если не занят, 1, если занят).
FCj или FC\$имя	Число занятий прибора.

FRj или FR\$имя	Нагрузка прибора, выраженная в долях тысячи.
FTj или FT\$имя	Целая часть значения среднего времени задержки транзакта на приборе.

Таблица 14.2 СЧА многоканальных устройств

Наименование	Значение
Rj или R\$имя	Емкость незаполненной части устройства.
Sj или S\$имя	Емкость заполненной части устройства.
SAj или SA\$имя	Целая часть среднего заполнения устройства.
SCj или SC\$имя	Счетчик числа входов в МКУ. (При каждом выполнении блока ENTER для устройства, значение счетчика увеличивается на величину операнда В этого блока).
SMj или SM\$имя	Максимально занятая емкость устройства. Запоминает максимальное значение Sj (S\$имя).
SRj или SR\$имя	Нагрузка МКУ, выраженная в долях тысячи.
STj или ST\$имя	Целая часть среднего времени пребывания транзакта в устройстве.

Таблица 14.3 СЧА очередей

Наименование	Значение
Qj или Q\$имя	Текущее значение длины очереди (текущее содержимое).
QAj или QA\$имя	Целая часть среднего значения длины очереди.
QCj или QC\$имя	Число входов в очередь. При каждом выполнении в блоке QUEUE для конкретной очереди значение QCj (QC\$имя) увеличивается на значение операнда В
QMj или QM\$имя	Максимальное значение длины очереди (максимальное значение Qj (Q\$имя)).
QTj или QT\$имя	Целая часть среднего времени пребывания в очереди для всех входов (включая и нулевые).
QXj или QX\$имя	Целая часть среднего времени пребывания в очереди для ненулевых входов).
SZj или SZ\$имя	Число нулевых входов в очередь.

!

Все СЧА - целочисленные.

1.2. СЧА блоков и системные числовые атрибуты

Блоки имеют два стандартных числовых атрибута (их подсчет ведется автоматически):

* **W_j** (W\$<метка блока>) - *счетчик текущего содержимого* блока с номером j (с именем имя блока);

* **N_j** (N\$<метка блока>) - *счетчиков входов*, т.е. общее число транзактов, вошедших в блок j от начала счета.

Так W\$BL1 является числом транзактов, находящихся в блоке с меткой BL1; N\$QP - число транзактов, вошедших в блок с меткой QP; W210 - значение счетчика содержимого блока, занимающего 210-ю позицию в модели.

Системные числовые атрибуты

C1 - текущее значение *относительного* модельного *времени*.

Автоматически изменяется интерпретатором и устанавливается в 0 командой CLEAR и RESET. (Команды описаны в разделе 22).

AC1 - текущее значение *абсолютного* модельного времени. Автоматически изменяется интерпретатором и устанавливается в 0 командой CLEAR.

TG1 - текущее значение *счетчика завершений*.

PR - приоритет обрабатываемого в данный момент транзакта.

M1 - время пребывания в модели транзакта, обрабатываемого интерпретатором в данный момент.

MP - время с момента входа транзакта в блок MARK.

Пример 14.1

ENTER MK3,R3

При входе транзакта в блок ENTER, он занимает R3 каналов устройства с именем MK3. Так как R3 - число доступных каналов МКУ 3, то транзакт занимает все оставшиеся до его входа свободные каналы.

ENTER SH,RSSH

При входе транзакта в блок ENTER, он занимает RSSH каналов устройства с именем SH.

ADVANCE FC\$PRI

Задержка в этом блоке равна числу занятий прибора PRI.

1.3. Параметры транзактов

Параметры транзакта - это свойства транзакта, определяемые пользователем. Они являются локальными переменными, принадлежащие только данному конкретному транзакту.

Свойства транзактов, как объектов, определяются множеством его параметров, к которым можно обратиться через СЧА. В процессе перемещения

транзакта по модели, его параметры могут задаваться и модифицироваться в соответствии с логикой работы модели.

Особенности параметров транзактов:

1. Доступ к параметрам транзакта осуществляется таким образом:
P<номер> или **P\$<имя>**,
где **P** - СЧА транзакта, определяющий его групповое имя, т.е. имя всех параметров транзактов.
Номера конкретных членов группы задаются с помощью целых чисел 1,2,... или имен. Например, P22 – это 22 –й параметр транзакта или P\$COLOR – параметр транзакта с именем COLOR.
1. При входе транзакта в модель начальные значения его параметров не определены.
2. Значения параметров и их изменения задает пользователь.
3. Транзакт может обращаться только к своим параметрам. Если необходимо получить доступ к параметрам других транзактов, то это можно сделать через ячейки сохранения или используя группы транзактов.
4. Параметры можно использовать в качестве операндов блоков и в качестве аргументов функций.
5. Параметры позволяют организовать косвенную адресацию блоков. Это дает возможность агрегированного представления объектов моделирования в программе.

!

Значениями параметров могут быть только целые числа со знаком.

Пример 14.2

AAA FUNCTION P3,D3
-3,4/3,7/10,8

...
ADVANCE FN\$AAA,3

Выполнение подпрограммы блока ADVANCE включает расчет функции AAA. Это в свою очередь требует определения значения аргумента функции, т.е. P3. Пусть значение третьего параметра P3 транзакта, вошедшего в блок ADVANCE, равно 4. Соответствующее значение функции AAA равно 8. Время задержки, таким образом, равномерно распределено на интервале 8 \div 3.

Пример 14.3

SAIZE P\$PRIB
ADVANCE P\$T4
RELISE P\$PRIB

Транзактом занимается прибор, номер которого задан параметром транзакта PRIB. Время обслуживания задается в параметре T4.

1.4. *Изменение значений параметров. Блок ASSIGN (НАЗНАЧИТЬ)*

При входе в этот блок значения параметров могут задаваться или изменяться.

Формат блока:

ASSIGN A[+,-],B[,C]

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Номер или имя модифицируемого параметра	Ошибка
B	Величина, используемая для модификации (число или СЧА)	Ошибка
C	Имя функции	

Блок ASSIGN может быть использован как в режиме замещения величины, так и в режиме накопления и уменьшения. В режиме накопления предыдущее значение параметра увеличивается на значение, стоящее в операнде **B**. В режиме уменьшения оно уменьшается на величину, стоящую в операнде **B**. Режимы накопления и уменьшения определяются введением соответственно знака плюс и минус перед запятой, разделяющей операнды **A** и **B**.

При использовании операнда **C** значение операнда **B** умножается на значение функции, указанной в операнде **C**. Параметр, заданный в операнде **A**, изменяется на целую часть полученного произведения.

Пример 14.4

Блок ASSIGN в *режиме присваивания*:

ASSIGN MEST,36

Параметру транзакта с именем MEST присваивается значение 36.

ASSIGN 3,25

Параметру P3 присваивается значение 25.

ASSIGN P3,FR\$BB

Параметру с номером, записанном в параметре P3, присваивается значение величины нагрузки прибора BB (оба операнда заданы косвенным образом).

Блок ASSIGN в *режимах приращения и вычитания*:

ASSIGN 4+,Q5

Параметр 4 увеличивается на значение, равное текущему содержимому очереди 5.

ASSIGN P2-,7

Из значения параметра, номер которого задан параметром P2, вычитается 7.

1.5. *Отметка времени*

При каждом входе транзакта в модель интерпретатор фиксирует для него текущее значение времени. Это значение времени называется *отметкой времени*. Она может быть интерпретирована как время "рождения" транзакта или время входа транзакта в модель. В явном виде отметка времени недоступна. Однако существует СЧА, который тесно связан со значением времени входа транзакта в модель. Его имя **M1**, а значение определяется таким образом:

$$M1 = \left\{ \begin{array}{l} \text{текущее значение} \\ \text{таймера абсолютного} \\ \text{времени} \end{array} \right\} - \left\{ \begin{array}{l} \text{значение времени} \\ \text{входа транзакта} \\ \text{модель} \end{array} \right\}$$

Значение M1 для каждого транзакта изменяется в процессе моделирования. Сразу после входа транзакта в модель M1=0, через 10 единиц - M1=10 и т.д.

14.6. *Транзитное время. Блок MARK (ОТМЕТИТЬ)*

При входе транзакта в блок MARK значение таймера абсолютного времени записывается в качестве одного из его параметров. Такую запись называют *отметкой транзакта*.

Формат блока MARK:

MARK A

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Номер параметра, в который записывается значение таймера абсолютного времени (целое число, СЧА)	При отсутствии операнда A, производится замена отметки времени на текущее значение абсолютного времени.

СЧА **M1** измеряет время прошедшее от момента входа транзакта в модель. Очень часто требуется знать сколько времени проходит при продвижении транзакта между двумя точками модели. Пусть нам требуется определить интервал времени, в течение которого транзакт проходит от точки A до точки B. Для этого следует выполнить два действия:

- 1) в точку А поместить блок MARK j, где j - номер параметра, в котором записывается значение таймера абсолютного времени в текущий момент времени;
- 2) в точке В обратиться к СЧА с именем MPj, где j - номер параметра, в котором транзакт отмечен. При этом значение MPj будет равно следующему:

$$MPj = \left\{ \begin{array}{l} \text{[текущее значение} \\ \text{таймера абсолютного} \\ \text{времени} \end{array} \right\} - \left\{ \begin{array}{l} \text{[значение]} \\ \text{[j - го} \\ \text{параметра]} \end{array} \right\}$$

14.7. Изменение приоритета транзактов. Блок PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ)

Блок PRIORITY присваивает или изменяет приоритет транзакта, если он был задан блокам GENERATE.

Его формат:

PRIORITY A

A целое число, СЧА, СЧА*<параметр> .

Новое значение приоритета может быть меньше, больше или равно текущему значению приоритета транзакта. Приоритет влияет на порядок выбора транзакта для обслуживания приборами и на порядок просмотра транзактов в списке текущих событий [7].

СЧА этого блока PR. Так как уровень приоритета транзакта может меняться от нуля до 127, то PR будет выдавать значения этого уровня в диапазоне $0 \div 127$.

Пример 14.5

PRIORITY 10

Вошедшему в этот блок транзакту присваивается приоритет 10.

14.8. Организация циклов. Блок LOOP (ЦИКЛ)

С помощью параметров транзактов можно в программе организовать циклы. Для этого используется блок LOOP. Он управляет количеством повторных прохождений транзактом определенной последовательности блоков модели.

Формат блока:

LOOP A,[B]

Операнд A определяет параметр транзакта, используемый для организации цикла . Может быть именем, положительным целым числом, СЧА, СЧА*<параметр>. По умолчанию – ошибка.

Операнд B задает имя блока (метку) начального блока цикла. По умолчанию – ошибка.

Когда транзакт входит в блок LOOP, в операнде A параметр уменьшается на единицу, а затем проверяется его значение на равенство нулю. Если значение

не равно нулю, то транзакт переходит в блок, указанный в операнде В. При равенстве значения параметра нулю транзакт переходит в следующий блок.

!

Блок LOOP может быть использован только для убывающего счета.

Пример 14.6

```
SIS SEIZE      PC
...
LOOP          1,SIS
PRIORITY      5
```

Цикл организован по первому параметру транзакта. Если этот параметр не равен нулю, то транзакт возвращается в блок, помеченный меткой SIS, т.е. занимает устройство с именем PC. При выходе из цикла транзакту назначается приоритет 5.

15. Примеры фрагментов GPSS - моделей с использованием СЧА и параметров транзактов

Пример 15.1

Определить функцию, значения которой зависели бы от текущего содержимого блока с меткой PPP. Вид зависимости задан в таблице 15.1.

Таблица 15.1

<i>Текущее содержимое блока с меткой PPP</i>	0	1,2 или 3	4 или 5	6	7 и более
<i>Значение функции</i>	1	4	2	4	5

```
FFF FUNCTION    W$PPP,D5
0,1/3,4/5,2/6,4/7,5
```

Пример 15.2

Определить функцию, значения которой были бы вдвое больше текущего содержимого очереди ALPHA для случаев 0,1,2,3,4. В остальных случаях значения функции должно быть равно 10. Сделать это двумя способами:

- а) с помощью дискретной функции, определяемой 6-ю значениями;
- б) с помощью непрерывной функции, определяемой 2-мя значениями.

Первый способ.

LONG FUNCTION Q\$ALPHA,D6

0,0/1,2/2,4/3,6/4,8/5,10

Второй способ.

SHORT FUNCTION Q\$ALPHA,C2

0,0/5,10

Пример 15.3

В систему массового обслуживания с одним прибором и очередью поступает пуассоновский поток требований с интенсивностью 12 приходов в час. Обслуживание является экспоненциальным, но среднее время обслуживания зависит от числа требований, находящихся в очереди к прибору. Эта зависимость представлена в таблице 15.2.

Таблица 15.2

<i>Длина очереди</i>	0	1 или 2	3,4 или 5	6 и более
<i>Среднее время обслуживания (мин)</i>	5.5	5.0	4.5	4.0

Здесь нас интересует включение в модель дискретной функции, в которой текущее содержимое очереди является аргументом. Это необходимо для учета длины очереди при определении интенсивности обслуживания и использования дискретной функции, а также для определения среднего значения интенсивности обслуживания.

1. Таблица определений.

Элементы GPSS	Интерпретация
Транзакты	Клиенты
Приборы:	
SURVR	Обслуживающий прибор
Функции:	
MEAN	Функция, определяющая среднее время обслуживания в зависимости от длины очереди
XPDIS	Функция розыгрыша случайных чисел по экспоненциальному закону со средним равным 1
Очереди:	
WAIT	Регистратор очереди для сбора статистики о состоянии очереди перед прибором

Единица модельного времени: 1 с.

II. GPSS-программа

```
MEAN      FUNCTION      Q$WAIT,D4
0,330/2,300/5,270/6,240
XPDIS     FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.38/8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/
.999,7/.9998,8
      GENERATE          30,FN$XPDIS
      QUEUE             WAIT
      SEIZE             SURVR
      DEPART            WAIT
      ADVANCE           FN$MEAN,FN$XPDIS
      RELEASE           SURVR
      TERMINATE        1
*
      START             500
```

Пример 15.4

Прибор с экспоненциальным обслуживанием имеет свойство уменьшать интенсивность своей работы в течение восьмичасового рабочего дня. В течение первых двух часов дня ему требуется в среднем 12 мин для выполнения обслуживания. В течение последующих 2 ч среднее время обслуживания составляет 15 мин. В течение 5-го, 6-го, 7-го часа - 17 мин. Обслуживание, начатое в течение 8-го часа, требует в среднем 20 мин. Предполагая, что единицей времени в модели 0.1 мин, определить функцию, значения которой давали бы среднее время, требуемое прибору для выполнения обслуживания. Показать, как эту функцию следует использовать в блоке ADVANCE. C1 - СЧА абсолютного текущего времени работы модели.

```
LAMDA     FUNCTION      C1,D4
1200,120/2400,150/4200,170/4800,200
XPDIS     FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
.75,1.38/8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/
.999,7/.9998,8
      ...
      ADVANCE          FN$LAMBDA,FN$XPDIS
```


Пример 15.5

Ситуация, описанная в примере 14.4., предполагает, что среднее время обслуживания увеличивается скачкообразно. Более реально полагать, что среднее время возрастает постепенно в течение дня. Показать, как определить непрерывную функцию, которая описывала бы увеличение времени обслуживания в соответствии с правилом: в момент времени 0 среднее время равно 12 мин; к концу 2-го часа оно увеличивается до 15 мин; к концу 4-го часа оно увеличивается до 17 мин; к концу 7-го - достигает 20 мин, а к концу 8-го - 25 мин. Считать, что увеличение среднего времени обслуживания непрерывно и равномерно на указанных интервалах времени.

```
MEAN      FUNCTION      C1,C5
0,120/1200,150/2400,170/4200,200/4800,250
XPDIS     FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2
.75,1.38/8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2
.999,7/.9998,8
...
ADVANCE   FNS$MEAN,FNS$XPDIS
```

16. Сохраняемые величины

В GPSS пользователю предоставляется возможность определить "свои" глобальные переменные, начальные значения которых могут быть определены перед моделированием и к которым можно обратиться из любого места модели во время прогона. Эти переменные называют **сохраняемыми величинами (ячейками)**.

Значениями сохраняемых величин могут быть только целые (положительные или отрицательные) числа. Значение сохраняемой величины является её **стандартным числовым атрибутом**:

X_j (X\$<имя сохраняемой величины>) - значение сохраняемой величины **j** (<имя сохраняемой величины>). Например, **X2** - значение сохраняемой величины 2; **X\$SAR** - значение сохраняемой величины SAR.

В отличие от параметров транзакта, приоритета и отметки времени, теряющихся в момент выхода транзакта из модели, сохраняемые величины не исчезают на протяжении всего процесса моделирования. Значения сохраняемых величин не подсчитываются интерпретатором как для СЧА, а изменяются программистом.

В качестве стандартных числовых атрибутов сохраняемые величины могут быть использованы для косвенного задания данных в поле операндов блока, а также как аргументы функций и таблиц.

Из сохраняемых величин могут быть организованы матрицы сохраняемых величин с помощью оператора MATRIX.

16.1. Оператор INITIAL (ИНИЦИАЛИЗИРОВАТЬ)

Если в процессе моделирования происходит обращение к сохраняемой величине, которая не была задана, то произойдет ошибка в процессе выполнения программы GPSS. Поэтому перед началом моделирования такие сохраняемые величины должны быть инициализированы с помощью оператора INITIAL.

Его формат:

<i>Поле</i>	<i>Информация в поле</i>
<i>Метка</i>	<имя сохраняемой величины>
<i>Операция</i>	INITIAL
<i>Операнд В</i>	<начальное значение>

Пример 15.2

INITIAL X\$TIMER,1000000

Сохраняемой величине TIMER присваивается начальное значение 1000000.

INITIAL X3,25

Сохраняемой величине с номером 3 присваивается начальное значение 25.

16.2. Блок SAVEVALUE (СОХРАНИТЬ ВЕЛИЧИНУ)

Значение сохраняемой величины изменяется при входе транзакта в блок SAVEVALUE (СОХРАНИТЬ ВЕЛИЧИНУ).

Его формат:

SAVEVALUE A[+,-],B,C

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Номер или символическое имя изменяемой сохраняемой величины	Ошибка

В	Величина, используемая для модификации (число или СЧА)	Ошибка
----------	--	--------

Подобно блоку ASSIGN, блок SAVEVALUE может быть использован как в режиме замещения величины, так и в режиме накопления и уменьшения. В режиме накопления предыдущее значение сохраняемой величины увеличивается на значение, стоящее в операнде **В**. В режиме уменьшения оно уменьшается на величину, стоящую в операнде **В**. Режимы накопления и уменьшения определяются введением соответственно знака плюс и минус перед запятой, разделяющей операнды **А** и **В**.

Пример 16.3.

SAVEVALUE P5,V\$ALPHA

При входе транзакта в блок SAVEVALUE прежде всего вычисляется величина переменной ALPHA. Полученный результат присваивается сохраняемой величине, номер которой записан в параметре P5. При этом старое значение сохраняемой величины уничтожается.

SAVEVALUE MARIE,TD\$TIME3,H

При входе транзакта в блок SAVEVALUE стандартное отклонение таблицы TIME3 заносится в полусловную сохраняемую величину с именем MARIE.

SAVEVALUE 5+,X2

При входе транзакта в блок величина X5 будет увеличена на значение величины X2.

SAVEVALUE DAVID-,FN\$HOLD

При входе транзакта в блок величина X\$DAVID будет уменьшена на величину значения функцию FN\$HOLD.

17. Косвенная адресация

Ранее были рассмотрены прямые способы адресации, когда:

- номер объекта задаётся константой

QUEUE 2 Стать в очередь 2

SEIZE 1 Занять прибор 1;

- номер объекта задаётся с помощью использования СЧА

LEAVE P1 Освободить МКУ с номером, задаваемым параметром 1.

Подобные ссылки не зависят от каких-либо свойств транзакта, обрабатываемого в данный момент. Использование прямой адресации может привести к введению большого числа дополнительных блоков только для того, чтобы записать номера объектов в стандартный числовой атрибут, т.е. к увеличению объема модели. Существенно сократить объем модели и

именам числовые значения, однако при этом пользователь не контролирует такое присвоение.

Назначение именам числовых значений осуществляется с помощью оператора EQU.

Пример 17.3

Необходимо занять МКУ, номер которого определяется в первом параметре транзакта. В зависимости от работы модели МКУ могут быть из двух или трех приборов обслуживания.

```

STAN1 EQU 1
STAN2 EQU 2
STAN1 STORAGE 2
STAN2 STORAGE 3

...
ASSIGEN 1,1
ENTER P1
...

```

18. Проверка числовых выражений. Блок TEST (ПРОВЕРИТЬ)

Соотношение между стандартными числовыми атрибутами может быть исследовано с помощью блока TEST.

Его формат:

TEST X A,B[,C]

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Имя стандартного числового атрибута	Ошибка
B	Имя стандартного числового атрибута	Ошибка
C	Имя блока, в который переходит проверяющий транзакт, если ответ на вопрос, подразумеваемый оператором отношения, отрицателен	При отсутствии операнда C проверку производят в режиме отказа
X	Вспомогательный оператор, представляющий собой оператор отношения, используемый при проверке. Ниже приведены значения, принимаемые оператором X :	
	<i>Оператор отношения</i>	<i>Вопрос, подразумеваемый в</i>

		<i>контексте блока TEST:</i>
	GE	А больше или равно В?
	E	А равно В?
	NE	А не равно В?
	LE	А меньше или равно В?
	L	А меньше В?

Операнды **А** и **В** являются именами сравниваемых стандартных числовых атрибутов. Вспомогательный оператор **X** указывает способ сравнения этих двух СЧА друг с другом.

Пример 18.1

Режим отказа.

TEST LE Q1,Q2

Проверяющий транзакт будет задержан в предыдущем блоке до тех пор, пока содержимое очереди 1 не станет меньше или равно содержимому очереди 2.

Пример 18.2

Режим условного перехода.

TEST LE Q1,Q2,BYPAS

Проверяющий транзакт перейдет в следующий по порядку блок, если содержимое очереди 1 меньше или равно содержимому очереди 2. Если это условие не выполнится, транзакт перейдет в блок с именем **BYPAS**.

Пример 18.3

TEST E F*5,0,FAC

Если свободен прибор, номер которого указан в параметре 5 транзакта, перейти к следующему блоку, в противном случае перейти в блок с меткой **FAC**.

19. Определение и использование таблиц

GPSS-таблицы используются для накопления выборочных значений случайных величин и статистической обработки этих выборок. Графическим аналогом GPSS-таблицы является гистограмма выборочных значений случайной величины.

Использование таблиц включает два действия:

- 1) определение таблицы, используемой в модели;
- 2) разделение входящих в выборку значений для каждой из описываемых таблиц.

1.1. *Оператор TABLE (ТАБЛИЦА)*

В модели может быть несколько таблиц. Таблицу определяют путем указания:

- имени таблицы (числового или символического);
 - имени случайной переменной, значения которой будут табулироваться;
- числа, являющегося первым граничным значением;
- значения выборки, меньшие или равные граничному числу, попадают в самый левый интервал (частотный класс) таблицы;
 - ширины интервала, общей для всех интервалов таблицы, за исключением левого (нижнего) и правого (высшего);
 - общего числа интервалов таблицы, включая низший и высший.

Формат оператора определения таблиц:

<i>Поле</i>	<i>Информация поля</i>
Метка	Имя таблицы
Операция	TABLE
Операнды:	
A	СЧА, значения которого должны учитываться в таблице
B	Первое граничное значение (целое)
C	Ширина всех промежуточных интервалов (целое положительное)
D	Общее число интервалов таблицы, включая левый и правый (целое положительное)

На рис. 19.1 показана графическая интерпретация оси действительных значений и ее разделение на ряд интервалов таблицы.

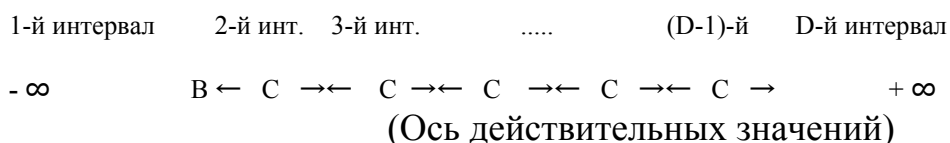


Рис. 19.1

1.2. **Блок *TABULATE* (ТАБУЛИРОВАТЬ)**

Значения выборки попадают в таблицу в те моменты времени, когда транзакты входят в блок *TABULATE*.

Формат блока:

TABULATE A

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Имя (символическое или числовое) таблицы, в которой используется соответствующее значение	Ошибка

Операнд **A** задает имя таблицы, в которую попадают значения выборки. Одну таблицу можно использовать в нескольких блоках *TABULATE* модели. Заметим, что сам стандартный числовой атрибут в блоке *TABULATE* не указывается, поскольку он уже записан в операторе *TABLE*.

Если в модели используются блоки *TABLE* или *QTABLE*, то в файле стандартной статистики будет представлена информация:

TABLE <i>Имя таблиц ы</i>	MEAN <i>Среднее</i>	STD. DEV. <i>Сред. кванд., откло-н ение</i>	RETRY <i>Ожидают условий</i>	RANGE <i>Границы</i>	FREQUE NCY <i>Частота</i>	CUM. % <i>Суммарная частота</i>
RTIME	4	0.04	0	10 - 15	9	1.37

СЧА таблицы:

ТВ<номер таблицы>, **ТВ**<имя таблицы> - вычисленное среднее значение соответствующего СЧА.

ТС<номер таблицы>, **ТС**<имя таблицы> - общее число входов в таблицу.

ТД<номер таблицы>, **ТД**<имя таблицы> - вычисленное среднеквадратическое отклонение соответствующего СЧА.

Пример 19.1

TYME2 TABLE MP3,10,5,6 Оператор определения таблицы
GENERATE 100,15

...
MARK **3**
 ...
TABULATE **TYME2**

В таблице TIME2 будет табулироваться частотное распределение случайной величины, равной времени пребывания транзакта между блоками MARK и TABULATE. В таблице 19.1 показаны интервалы регистрации попадания времени пребывания.

Пример 19.2

Необходимо собирать в таблицу TYME время пребывания транзакта в модели, которое принимает значения в интервале $100 \div 200$.

TYME TABLE M1,100,12,10

Таблица 19.1

Левый (низший) интервал	1-й интервал:	$-\infty, <10$
Промежуточные интервалы	2-й интервал:	$\geq 10, <15$
	3-й интервал:	$\geq 15, <20$

Продолжение таблицы 19.1

Промежуточные интервалы	4-й интервал:	$\geq 20, <25$
	5-й интервал:	$\geq 20, <30$
Правый (высший) интервал	6-й интервал:	$\geq 30, +\infty$

!

- Для того, чтобы табулируемые данные имели "нормальный" вид (не попадали в один-два интервала), рекомендуется: вначале сделать пробный прогон;
- * по нему определить диапазон значений, в котором может колебаться анализируемая случайная величина (перейти в окно таблиц с помощью клавиш [ALT+T]);
- * с учетом полученного диапазона откорректировать значения операндов **B**, **C** и **D** соответствующей таблицы.

20. СПИСКИ ПОЛЬЗОВАТЕЛЕЙ

1.1. Блок *LINK* (*ВНЕСТИ В СПИСОК*)

Блок *LINK* удаляет транзакты из списка текущих событий и помещает в список пользователя, а интерпретатор их не просматривает и не продвигает по блокам модели пока пользователь не возвратит их в модель.

Формат блок:

LINK A,B[,C]

Поле **A** задает номер или имя списка пользователя, в который будет помещен вошедший транзакт. Операнд **A** может быть положительным целым, СЧА, СЧА*СЧА.

Поле **B** задает алгоритм упорядочивания списка пользователя.

Операнд **B** может быть LIFO, FIFO, целым, СЧА, СЧА*<параметр>.

Допустимые операнды:

- **FIFO** - вошедший транзакт помещается в конец списка пользователя;
- **LIFO** - вошедший транзакт помещается в начало списка пользователя;
- номер параметра.

Если не используете LIFO или FIFO в операнде **B**, то можно использовать СЧА, такие как: PR – приоритет транзакта, M1 - время нахождения транзакта в модели или параметр транзакта. Если используется PR, то транзакт помещается в приоритетном порядке. Если задан номер параметра, то транзакт помещается в список пользователя за транзактом, у которого номер параметра меньше. Чем меньше значение параметра - тем ближе транзакт к началу списка, чем больше значение параметра - тем ближе транзакт к концу списка. Если значение параметра транзакта, помещаемого в список, равно значению параметра транзакта (или нескольких транзактов) уже находящегося в списке, то входящий транзакт помещается за последним из транзактов, имеющих одинаковые значения Pj.

Поле **C**, указывает альтернативный выход и используется при описании различных ситуаций, возникающих в очередях. Операнд **C** может быть именем, положительным целым, СЧА, СЧА*<параметр>.

Если поле **C** пусто – *безусловный режим*, индикатор, связанный с заданным списком пользователя, устанавливается в "1". Это приводит к тому, что все входящие транзакты обязательно заносятся в список пользователя, определенный полем **A**, в том порядке, какой задан полем **B**.

Если поле **C** не пустое – *условный режим*, то проверяется индикатор списка пользователя. Если индикатор списка установлен в "1", вошедшее транзакт заносится в список пользователя в порядке, заданном в поле **B**.

Если индикатор списка установлен в "0", он переводится в "1", и вошедшее транзакт переходит к блоку, заданному полем **C**.

Индикатор связи списка пользователя играет роль впереди смотрящего и имеет много общего со светофором. Он полезен при работе с приборами обслуживания.

Пример 20.1

LINK ONHOLD, FIFO

В этом примере вошедший транзакт помещается в конец списка пользователя с именем ONHOLD.

СЧА, связанные с описываемым оператором:

СА<номер списка> - среднее число транзактов в списке пользователя с именем <номер списка>.

СС<номер списка> - общее число транзактов в списке пользователя с именем <номер списка>.

СН<номер списка> - текущее число транзактов в списке пользователя с именем <номер списка>.

СМ<номер списка> - максимальное число транзактов в списке пользователя с именем <номер списка>.

СТ<номер списка> - среднее время пребывания сообщения в списке пользователя с именем <номер списка>, которое рассчитывается так:

$$\text{СТ<номер списка>} = \frac{\text{вычисленный временной интервал}}{\text{общее число входов}} .$$

20.2. Блок UNLINK (ВЫВЕСТИ ИЗ СПИСКА)

Блок UNLINK удаляет транзакты из списка пользователя.

Формат блока:

UNLINK [<X>] <A>,,<C>,<D>,<E>,<F>]

Интерпретатор GPSS возобновляет их движение по модели. Указатели отношения, записываемые во вспомогательном поле операции <X> определяют, какое условие следует рассматривать. Если указатель отношения не задан, предполагается отношение равенства 'E'.

Указатели отношения следующие:

- 'G' – больше; отношение истинно, если значение параметра, заданного полем D, больше значения, заданного полем E;

- 'GE' - больше или равно; отношение истинно, если значение параметра, заданного полем D, больше или равно значению, заданному полем E;

- '**L**' – меньше; отношение истинно, если значение параметра, заданного полем **D**, меньше значения, заданного полем **E**;

- '**LE**' - меньше или равно, если значение параметра, заданного полем **D**, меньше или равно значению, заданному полем **E**, то отношение истинно;

- '**E**' – равно; отношение истинно, если значение параметра, заданного полем **D**, равно значению, заданному полем **E**;

- '**NE**' - не равно; отношение истинно, если значение параметра, заданного полем **D**, не равно значению, заданному полем **E**.

Поле **A** задает номер списка пользователя, из которого удаляются один или несколько транзактов. Операнд **A** может быть именем, положительным целым, СЧА или СЧА*<параметр>.

В поле **B** указывается номер блока, к которому переходят удаленные сообщения. Операнд **B** может быть именем, положительным целым, СЧА, СЧА*<параметр>.

Поле **C** задает счетчик числа удаленных транзактов. Операнд **C** может быть "**ALL**", именем, положительным целым, СЧА, СЧА*<параметр>. Значение СЧА является числом удаляемых транзактов. Может быть задано "**ALL**", что означает удаление всех транзактов.

Поле **D** может быть именем, целым, СЧА, СЧА*<параметр>, "**BACK**". Оно может содержать:

1. Номер параметра. Если поле **E** пусто, значение заданного параметра вошедшего транзакта сравнивается со значением этого же параметра транзакта списка пользователя. Если поле **E** не пусто, значение заданного параметра транзакта из списка пользователя сравнивается со значением СЧА из поля **E**. В обоих случаях сообщения, удовлетворяющие заданному отношению, будут удалены из списка и направлены в блок, указанный в поле **B**;

2. Слово "**BACK**". Из указанного списка пользователя, начиная с конца списка, будет удалено столько транзактов, сколько задано счетчиком в поле **C**. Поле **E** в этом случае должно быть пустым;

3. Булевскую переменную BV_j , которая вычисляется отдельно для каждого сообщения из списка пользователя. Если булевская переменная BV_j имеет ссылки на какой-либо параметр, то эти ссылки относятся к параметрам транзактов из списка пользователя, а не к вошедшему транзакту. Транзакт удаляются из списка пользователя в соответствии со значением счётчика, заданный полем **C**, только в том случае, если значение $BV_j=1$. Если $BV_j=0$ для всех транзактов списка, вошедший транзакт пытается перейти к блоку, заданному полем **F**. Если поле **F** пусто, транзакт пытается перейти к следующему по номеру блоку. Когда в поле **D** задана булевская переменная, тогда поле должно быть пустым.

Поле **E** содержит стандартный числовой атрибут, значение которого сравнивается со значением параметра транзакта списка пользователя (номер параметра указан в поле **D**). Операнд **E** может быть именем, целым, СЧА, СЧА*<параметр>.

Поле **F** задаёт номер следующего блока для входящего в блок UNLINK транзакта в таких случаях:

- когда соответствующий список пользователя пуст;
- когда не выполнено заданное отношение или когда указанная в поле **D** переменная равна нулю.

Операнд **F** может быть именем, положительным целым, СЧА, СЧА*<параметр>.

Пример 20.2.

UNLINK ONHOLD, REENTRY, 1

Первый транзакт из списка пользователя с именем ONHOLD помещается в блок с именем REENTRY. Он заносится в список текущих событий за транзактом с таким же приоритетом. Входящий транзакт переходит к следующему блоку.

Рассмотрим последовательность операций, выполняемых при входе транзакта в блок UNLINK. В зависимости от того, какие из полей **A - F** блока UNLINK заполнены, можно выделить восемь основных вариантов.

Вариант 1. Поля **A, B, C** заполнены, поля **D, E, F** пусты. Вычисляется значение атрибута из поля **A** для определения номера списка пользователя. Производится проверка, есть ли в списке пользователя транзакты. Если в списке нет транзактов, соответствующий этому списку индикатор списка устанавливается в "0", и входящий транзакт переходит к следующему по номеру блоку. Если список не пуст, вычисляется значение атрибута из поля **C** для определения числа удаляемых из списка транзактов. Транзакты удаляются, начиная с первого транзакта списка до тех пор, пока содержимое счетчика не станет равным нулю или пока не будут исчерпаны транзакты в списке. Удаленные из списка пользователя транзакты будут помещены в список текущих событий и направлены к блоку, номер которого указан в поле **B**. Транзакт, вошедший в блок UNLINK, перейдет к следующему по номеру блоку.

Вариант 2. Поля **A, B, C** и **F** заполнены, поля **D** и **E** пусты. Этот вариант аналогичен варианту 1, за исключением случая, когда указанный список пользователя пуст. В этом случае индикатор списка устанавливается в "0", а вошедшее в блок UNLINK соотношение переходит к блоку, указанному в поле **F**, а не к следующему по номеру блоку.

Вариант 3. Поля **A, B, C** и **D** заполнены, поля **E** и **F** пусты. Этот вариант также аналогичен варианту 1. Однако из списка пользователя удаляются только те транзакты, у которых значение параметра P_j , заданного полем **D**, равно значению параметра P_j вошедшего в блок UNLINK транзакта. Удаление транзактов из списка производится до тех пор, пока содержимое счетчика в поле **C** не станет равным нулю или пока не будут просмотрены все транзакты списка. Все удаленные из списка пользователя транзакты переходят к блоку, указанному в поле **B** блока UNLINK. Транзакт, вошедший в блок UNLINK, переходит к следующему по номеру блоку.

Вариант 4. Поля **A, B, C** и **D** заполнены, поля **E** и **F** пусты, а полем **D** записано слово "BACK". Этот вариант аналогичен варианту 1, за исключением того, что транзакты удаляются, начиная с конца списка пользователя.

Вариант 5. Поля **A, B, C** и **D** заполнены, поля **E** и **F** пусты, в поле **D** задана булевская переменная BV_j . Этот вариант аналогичен варианту 3, за исключением того, что BV_j вычисляется отдельно для каждого транзакта списка пользователя. Обратите внимание, что если BV_j имеет ссылки к какому-либо параметру, то эти ссылки относятся к параметрам, связанным с транзактами из списка пользователя, а не с вошедшим транзактом. Из списка пользователя удаляются только те транзакты, для которых $BV_j=1$.

Вариант 6. Поля **A, B, C, D** и **F** заполнены, поле **E** пусто. В этом варианте поле **F** заполнено, а в варианте 3 нет. Эти варианты аналогичны, за исключением тех случаев, когда заданный список пользователя пуст ($CH_j=0$), или заданное отношение не выполняется ни для одного сообщения списка, или $BV_j=0$ для всех транзактов списка пользователя. В таких случаях транзакт, вошедший в блок UNLINK, переходит к блоку, номер которого задан в поле **F**, а не к следующему по номеру блоку. Индикатор списка устанавливается в "0" только в том случае, если список пуст.

!

В случаях, когда в поле **D** блока UNLINK записано BV_j или "BACK", поле **E** должно быть пустым, иначе происходит ошибка.

Вариант 7. Поля **A, B, C, D** и **E** заполнены, поле **F** пропущено.

Число удаляемых транзактов определяется значением атрибута из поля **C**. Номер блока, к которому направляются удаленные транзакты, определяется полем **B**. Транзакты, для которых значение P_j (поле **D**) равно СЧА из поля **E**, удаляются из списка пользователя. Транзакты просматриваются и удаляются (если это возможно), начиная с начала списка пользователя. Удаление продолжается до тех пор, пока значение счетчика не станет равным нулю или пока из списка пользователя не будут удалены все транзакты. Транзакт, вошедший в блок UNLINK, переходит к следующему по номеру блоку.

Вариант 8. Поля **A, B, C, D** и **F** заполнены. Этот вариант аналогичен варианту 5, за исключением тех случаев, когда список пользователя пуст (т.е. $CH_j=0$) или заданное отношение не выполнилось ни для одного из транзактов списка. Транзакт, вошедший в блок UNLINK, переходит к блоку, заданному в блоке **F**, а не к следующему по номеру блоку. Если заданное отношение выполняется для какого-нибудь транзакта из списка, вошедший транзакт переходит к следующему по номеру блоку. Индикатор списка устанавливается в "0", если список пользователя пуст.

Стандартными числовыми атрибутами, связанными с описываемым оператором, приведены в п. 20.2

Пример 20.3

Рассмотрим модель с обслуживанием очереди по алгоритму FIFO:

	GENERATE	1000, FN1
	QUEUE	QUE1
	LINK	CHAIN, FIFO, CAN
CAN	SEISE	FC1
	DEPART	QUE1
	ADVANCE	700, FN1
	RELEASE	FC1
	UNLINK	CHAIN, CAN, 1
	TERMINATE	

Транзакты, выходящие из блока **GENERATE**, поступают в блок **QUEUE**. После внесения изменений в статистику, собираемую в данной очереди, транзакт входит в блок **LINK**. Поскольку в данном блоке **LINK** есть альтернативный выход, проверяется состояние индикатора соответствующего списка. Если индикатор списка равен "0", транзакт немедленно устанавливает его в "1" и переходит к альтернативному блоку, т.е. в данной модели блоку **SEISE**. Если индикатор списка равен единице, то транзакт, входящий в блок **LINK**, без дальнейших проверок заносится в список пользователя, номер которого определяется полем **A** блока **LINK**. Поскольку в поле **B** задан алгоритм **FIFO**, транзакт помещается в конец списка пользователя.

Отметим, что транзакты, занесенные в список пользователя, не проходят дальнейшей обработки до тех пор, пока они не будут удалены из списка другими транзактами, входящими в блок **UNLINK**. После удаления из списка пользователя транзакт (или транзакты) переходит на блок, указанный в поле **B** блока **UNLINK**.

Когда транзакт входит в блок **SEISE**, производится корректировка статистики, связанной с заданным в этом блоке устройством.

Затем транзакт переходит к блоку **DEPART**, где корректируется статистика очереди. Транзакт затем переходит к блоку **ADVANCE**, в котором он остается в течение времени, определяемого средним значением времени задержки и модификатором блока **ADVANCE**.

После выхода из блока **ADVANCE** транзакт входит в блок **RELEASE**. Освобождается указанное в блоке устройство и корректируется связанная с ним статистика. После этого транзакт переходит к блоку **UNLINK**.

Когда транзакт переходит к блоку **UNLINK**, просматривается список пользователя, номер которого задан полем **A**. Если указанный список пуст, т.е. $CH_j = 0$, то индикатор списка устанавливается в "0", и транзакт переходит к следующему по номеру блоку, к блоку **TERMINATE**.

Если список пользователя не пуст, т.е. $CH_j \neq 0$, то первый транзакт из списка пользователя удаляется, помещается в список текущих событий и

направляется в блок, указанный полем **B** блока UNLINK. В данной модели это блок SEIZE, помеченный идентификатором CAN. Затем вошедший транзакт перейдет к следующему по номеру блоку. Хотя эта модель и не сложна, на ней можно проиллюстрировать несколько важных моментов:

1. В этой системе активными, т.е. находящимися в списках текущих и будущих событий, списках прерываний или списках задержки, являются только транзакты, которые выходят из блока GENERATE, и транзакт, который в данный момент занимает устройство. Все остальные транзакты находятся в списке пользователя CHAIN;

2. Поскольку все задержанные транзакты, т.е. транзакты, находящиеся в очереди к устройству FC1, будут помещены в список пользователя CHAIN, интерпретатор не будет тратить время на изменения индикаторов задержки всех этих транзактов при каждом изменении состояния устройства. Величина экономии времени зависит от длины очереди. Чем длиннее очередь, тем больше времени будет сэкономлено при помощи блоков LINK/UNLINK, применяемых для управления очередями различных объектов;

3. Пользователь имеет возможность формировать свои списки в динамике вне зависимости от списков задержки, которые управляются системой GPSS/PC;

4. Использование блоков LINK/UNLINK дает возможность синхронизировать движение разных транзактов в модели, например, задержать в списке транзакт, пока какой-то другой транзакт не выведет их из списка с помощью блока UNLINK.

Пример 20.4

Рассмотрим работу мультиплексора, который подключен к высокоскоростному каналу связи и работает в режиме разделения времени с четырьмя низкоскоростными каналами, опрашивая их циклически. На один опрос каналов мультиплексор затрачивает 50 мс, а время переключения между каналами 10 мс. Если в опрашиваемом канале есть сообщение, оно передается по высокоскоростному каналу в течение времени 500 ± 60 мс. За один опрос передается одно сообщение. Время возникновения сообщений в низкоскоростных каналах равномерно распределено в интервале 1500 ± 500 мс и может появляться равновероятно на любом канале.

Необходимо определить загрузку высокоскоростного канала связи при передаче 1000 сообщений.

OPROS FUNCTION P1, D4

*

определение № канала, где возникло сообщение

1,2/2,3/3,4/4,1

NUMBKS FUNCTION RN1, D4

функция опроса каналов

.25,1/.5,2/.75,3/.9999,4

* <i>Опрос каналов</i>			
	GENERATE	10,,,1	генерация опросного сообщения
	ASSIGN	1,4	последний опрашиваемый канал
POOL	ASSIGN	1,FN\$OPROS	№ опрашиваемого канала
	SAIZE	MPD	занять скоростной канал
	ADVANCE	50	время опроса
	TEST NE	CH*1,0,MREL	есть ли сообщение в канале?
	UNLINK	P1,MESS,1	разрешить передачу по каналу
MREL	RELEASE	MPD	освободить скоростной канал
	ADVANCE	10	время переключения
	TRANSFER	,POOL	продолжать опрос каналов
* <i>Передача сообщений по каналам</i>			
	GENERATE	1500,500	генерация сообщения
	ASSIGN	1,F N\$NUMBKS	№ канала, где возникло сообщение
	LINK	P1,FIFO	ждать опроса канала
MESS	SAIZE	MPD	занять канал для передачи
	ADVANCE	500,60	время передачи
	RELEASE	MPD	освободить канал
	TERMINATE	1	
	START	1000	передать 1000 сообщений

Пример 20.5

На вход СМО поступает пуассоновский поток заявок со средним временем 55 с, которые обслуживаются последовательно двумя приборами. Порядок выбора заявок из очередей для обслуживания первым и вторым приборами – LIFO. Причем, если в первой очереди больше 4-х заявок или во второй очереди время пребывания в ней больше 2 с, то заявки покидают систему не обслуживаясь. Время обслуживания первым прибором 30 с, а вторым – 20 с. Про моделировать прохождение 100 заявок.

10 EXP FUNCTION

RN1,C240,0/.100,.104/.200,.222/.300,.355/.400,.509.500,.690/.600,.915/.700,1.200/.750,1.380

.800,1.600/.840,1.830/.880,2.120/.900,2.300

.920,2.520/.940,2.810/.950,2.990/.960,3.200

.970,3.500/.980,3.900/.990,4.600/.995,5.300

.998,6.200/.999,7/1,8

20 GENERATE 55,FN\$EXP

30 TEST L CH\$SP1,4,TER1

В списке SP1 меньше 4-х транзактов?

40 QUEUE OCH1

Да, встать в очередь. Нет, покинуть систему

50 LINK SP1,LIFO,MUS1

60 MUS1 SEIZE USTR1

```

70  DEPART  OCH1
80  ADVANCE 30
90  RELEASE USTR1
100 UNLINK  SP1,MUS11
110 MARK    TOCH2           Запомнить время пребывания в очереди
120 TEST L  MP$TOCH2,2,TER1  Время нахождения в списке SP1 < 2 с ?
130 LINK    SP2,LIFO,MUS2    Да, идти на обслуживание. Нет, покинуть
135 *                               систему
140 MUS2 SEIZE  USTR2
150 DEPART  OCH2
160 ADVANCE 20
170 RELEASE USTR2
180 UNLINK  SP2,MUS2
190 TER1 TERMINATE 1
200 START   100

```

В модели используется индикатор связи списка пользователя вместе с приборами обслуживания, что обеспечивает просмотр вперед для определения состояния прибора. Если индикатор связи включен, транзакт ставится в начало списка SP1 или SP2 (LIFO) и индикатор остается включенным. Если индикатор связи был выключен, он включается, а транзакт переходит в блок MUS1 или MUS2. Время пребывания во второй очереди запоминается в параметре транзакта с именем TOCH.

Пример 20.6

Рассмотрим модель эвакуации пострадавших и раненых во время проведения боевых действий. Пострадавшие эвакуируются 5 - и 13 - местными автомобилями. Если на пункте эвакуации скапливается раненых больше чем мест в автомобиле, то вывозятся столько - сколько есть свободных мест. В противном случае забирают всех. Известны функции времени прибытия раненых в медицинский пункт полка (МПП). Всего в модели рассматривается четыре МПП. Раненые эвакуируются в военно-полевой передвижной госпиталь (ВППГ), где им оказывается необходимая помощь.

```

2 EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2
.75,1.38/8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81
.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2
.999,7/1,8
3 EXP1     FUNCTION RN1,C21           функция распределения интервалов
0,0/. 0078,0/. 2556,2/. 4544,4/. 7081,6/. 8023,8   времени прибытия раненых
8694,10/. 9151,12/. 9418,14/. 9625,16/. 9763,18   в 1-й МПП

```

9859,20/. 9896,22/. 9933,24/. 9955,26/. 9963,28

9974,30/. 9981,32/. 9989,34/. 9996,36/1,96

4 MEST_VP FUNCTION X\$CARVP, D2 работают 9 автомобилей 5-местных
9,5/18,13 и 9 - 13-местных

4 *

5 GENERATE ,, 30,18 эвакуация пострадавших 18 - ю автомобилями,
6 * момент выезда через 30 мин. после начала боя

7 SAVEVALUE CARVP+, 1 определение номера автомобиля

8 ASSIGN MESTOVP, FN\$MEST_VP определение количества мест в
9 * автомобиле

10 VOZVRPV TRANSFER .25,, POLK4 25% автомобилей едут в полк
11 * второго эшелона

12 ADVANCE 120,20 длительность рейса до полка 1-го эшелона от
13 * 1 часа 40 мин. до 2 часов 20 мин.

14 TRANSFER ,POLK выехать в МПП

15 POLK4 ADVANCE 40,10 длительность рейса до полка 2-го эшелона
16 * от 30 до 50 мин.

17 POLK TEST LE CH\$EVAC_POL, P\$MESTOVP, BOLVP если

17.1 * пострадавших

18 UNLINK EVAC_POL, VPPG, CH\$EVAC_POL меньше,

19 * чем мест в автомобиле, то эвакуируют всех

20 TRANSFER ,VOZVRPV новый рейс в МПП

21 BOLVP UNLINK EVAC_POL, VPPG, P\$MESTOVP эвакуируют

22 * пострадавших столько, сколько мест в автомобиле

23 TRANSFER ,VOZVRPV новый рейс в МПП

24 * Эвакуация пострадавших с медицинского пункта полка 1-го эшелона

25 GENERATE FN\$EXP1,, 100 поступление пострадавших в МПП

26 * 1-го эшелона, первый поступает не раньше 100 мин. после начала боя

27 ASSIGN 1,1 определение номера полка

28 ASSIGN TRAN, FN\$TPOLK время от момента ранения до

29 * подготовки в МПП для следующей эвакуации

30 SAVEVALUE SV_P1+, 1 подсчет санитарных потерь в 1-м полку

31 TRANSFER ,NAK направить на эвакуацию.

32 * Эвакуация пострадавших с медицинского пункта полка 1-го эшелона с

33 * других направлений

34 GENERATE 8, FN\$EXP, 100

35 ASSIGN 1,2

36 ASSIGN TRAN, FN\$TPOLK

37 SAVEVALUE SV_P2+, 1

38 TRANSFER ,NAK

39 * Эвакуация пострадавших с медицинского пункта полка 1-го эшелона с

40 * других направлений

41 GENERATE 10, FN\$EXP, 100

42	ASSIGN	1,3	
43	ASSIGN	TRAN, FN\$TPOLK	
44	SAVEVALUE	SV_P3+, 1	
45	TRANSFER	,NAK	
46 *	<i>Эвакуация пострадавших с медицинского пункта полка 1-го эшелона с</i>		
47 *	<i>других направлений</i>		
48	GENERATE	10, FN\$EXP, 100	
49	ASSIGN	1,4	
50	ASSIGN	TRAN, FN\$TPOLK	
51	SAVEVALUE	SV_P4+, 1	
52	NAK LINK	EVAC_POL, FIFO	ожидание эвакуации с МПП
53	VPPG TEST L	P1,4, MPOL4	это не полк 2-го эшелона?
54	ADVANCE	60,10	если полк 1-го эшелона, то время
54.1*			эвакуации 50-70 мин.
55	TRANSFER	,MVPPG	эвакуация в ВППГ
56	MPOL4 ADVANCE	20,5	если полк 2-го эшелона, то время
56.1 *			эвакуации 15-25 мин.
57 *	<i>Прибытие медицинского транспорта на сортировочный пост ВППГ</i>		
58	MVPPG ADVANCE	2,1	длительность сортировки 1-3 мин.
	...		

21. Статистическая обработка результатов моделирования

В языке моделирования GPSS статистика о работе модели автоматически выводится в файл результатов или по желанию пользователя. Однако при этом не гарантируется надежность получаемых оценок. Этим вопросом должен заниматься сам пользователь. Для имитационных стохастических моделей, работающих в переходном режиме, необходимо провести несколько прогонов модели, каждый из которых должен отличаться своей последовательностью псевдослучайных чисел. Для этого используется последовательность команд (подробнее о командах см. раздел 22):

```

RMULT  7
START  1
CLEAR

...
RMULT  17
START  1

```

Команда RMULT позволяет устанавливать начальные значения множителей для генераторов случайных чисел. Команда CLEAR осуществляет сброс

собранный статистики по предыдущему прогону модели, но не устанавливает множители генераторов случайных чисел в начальные значения.

Получаемые результаты моделирования могут быть записаны командой RESULT в специальный файл с целью проведения простейшего дисперсионного статистического анализа. Команды ANOVA определяет доверительный интервал и значимость факторов по критерию Фишера (F - критерий).

При моделировании стохастических систем, работающих в стационарном режиме, может быть использован регенеративный анализ, если эти системы регенерируют. Для СМО моменты регенерации будут определяться номерами тех требований, которые будут, например, заставлять прибор обслуживания свободным. Класс регенерирующих систем достаточно большой. К нему относятся стохастические сети СМО, система управления запасами и др. Поэтому приведем алгоритм построения доверительного интервала с использованием этого метода [7]:

1. Провести n циклов регенерации.

2. Вычислить Y_j и a_j для каждого j -го цикла, где Y_j сумма выходных значений исследуемой переменной, полученной на имитационной модели на j -м цикле регенерации, и a_j - длина j -го цикла.

3. Вычислить выборочные статистики:

$$\bar{Y} = 1/n \sum_{j=1}^n Y_j, \quad \bar{a} = 1/n \sum_{j=1}^n a_j, \quad r = \bar{Y} / \bar{a};$$

$$S_{11} = 1/(n-1) \sum_{j=1}^n Y_j^2 - n/n(n-1) \left(\sum_{j=1}^n Y_j \right)^2;$$

$$S_{22} = 1/(n-1) \sum_{j=1}^n a_j^2 - n/n(n-1) \left(\sum_{j=1}^n a_j \right)^2;$$

$$S_{12} = 1/(n-1) \sum_{j=1}^n Y_j a_j - n/n(n-1) \left(\sum_{j=1}^n Y_j \right) \left(\sum_{j=1}^n a_j \right);$$

$$S^2 = S_{11} - 2 r S_{12} - r^2 S_{22},$$

где r оцениваемое значение выходной переменной имитационной модели; S_{11} , S_{22} , S_{12} - обозначают соответственно выборочную дисперсию от значений Y_j , выборочную дисперсию a_j и выборочный второй смешанный момент от значений (Y_j, a_j) .

4. Сформировать доверительный интервал:

$$r \pm \frac{z_d^* S}{\alpha n^{1/2}},$$

где $z_d^* = \Phi^{-1}(1 - d/2)$; Φ - функция стандартизированного нормального распределения.

Если первый цикл не начинается одновременно с моделированием, то данные, предшествующие первому циклу, необходимо отбросить.

Пример 21.1

Покажем как можно использовать данный алгоритм в GPSS программе. Ниже приведена GPSS программа для моделирования СМО вида М/М/1, для которой оценивается значение времени пребывания заявки в системе и строится доверительный интервал с 90% -м уровнем доверия.

```

31 EXP    FUNCTION    RN1,C24
0,0/.100,.104/.200,.222/.300,.355/.400,.509
.500,.690/.600,.915/.700,1.200/.750,1.380
.800,1.600/.840,1.830/.880,2.120/.900,2.300
.920,2.520/.940,2.810/.950,2.990/.960,3.200
.970,3.500/.980,3.900/.990,4.600/.995,5.300
.998,6.200/.999,7/1,8
41 BAD2   TABLE     M1,0,500,30 ; таблица времени пребывания
51 KBA    VARIABLE    X2^2        ; расчет  $\alpha$ 
61 KBY    VARIABLE    X1^2        ; расчет  $\bar{Y}$ 
71 ALY    VARIABLE    X1#X2       ; расчет  $\alpha \bar{Y}$ 
81 YSR    FVARIABLE    X3/X4       ; расчет  $\bar{Y}$ 
91 NNN    FVARIABLE    1/(X8-1)    ; расчет 1/(n-1)
101 NN1    FVARIABLE    X8/V$NNN    ; расчет n/(n-1)
111 SS11   FVARIABLE    X6/(X8-1)-X3/X8#X3/(1-X8) ; расчет  $S_{11}$ 
121 SS22   FVARIABLE    X5/(X8-1)-X4/X8#X4/(1-X8) ; расчет  $S_{22}$ 
131 SS12   FVARIABLE    X7/(X8-1)-X3/X8#X4/(X8-1) ; расчет  $S_{12}$ 
141 SKB    FVARIABLE    V$SS11-2#V$YSR#V$SS12+V$YSR^2#V$SS22
;расчет  $S^2$ 
151        GENERATE    200,FN$EXP    ; генерация потока заявок
161 INP    QUEUE      BAD1          ; постановка в очередь
171        SEIZE       BAD1          ; захват прибора
181        DEPART      BAD1          ; освобождение очереди
191        ADVANCE     180,FN$EXP    ; обслуживание
201 OUT    RELEASE     BAD1          ; освобождение прибора
211        TEST NE     N$OUT,N$INP,CIKL ; проверка начала цикла

```

221	SAVEVALUE	2+,1	; подсчет j в цикле
231	SAVEVALUE	1+,M1	; подсчет α_j в цикле
241	TABULATE	BAD2	; табулирование времени пребывания
251	TERMINATE		
261	CIKL	SAVEVALUE 2+,1	; подсчет последнего j в цикле
271	SAVEVALUE	1+,M1	; подсчет последнего α_j в цикле
281	SAVEVALUE	3+,X1	; $\sum Y_j$
291	SAVEVALUE	4+,X2	; $\sum \alpha_j$
301	SAVEVALUE	5+,V\$KBA	; $\sum \alpha_j^2$
311	SAVEVALUE	6+,V\$KBY	; $\sum Y_j^2$
321	SAVEVALUE	7+,V\$ALY	; $\sum \alpha_j * Y_j$
331	SAVEVALUE	1,0	; начало нового цикла
341	REG	SAVEVALUE 2,0	
351	TEST NE	X8,N\$REG,ENDREG	; закончить моделирование?
361	SAVEVALUE	8+,1	; подсчет числа циклов
371	TABULATE	BAD2	; табулирование последнего значения
*			времени пребывания в цикле
381	TERMINATE	1	
391	ENDREG	SAVEVALUE 8+,1	; учет последнего цикла
401	SAVEVALUE	9,V\$SKB	; запомнить значение S^2
411	TERMINATE	1	
421	START	1000,NP	; закончить моделирование через
422*			1000 циклов
431	SHOW	X3/X4	; вывести на экран среднее время
441	SHOW	1.645#SQR(V\$SKB)#X8/X4/SQR(X8)	; и доверительный
442*			интервал с 90% уровнем доверия

22. КОМАНДЫ GPSS/PS

Команды GPSS/PC необходимы, как для построения программы модели, так и для интерактивного взаимодействия с моделью. В противоположность операторам описания данных и контроля управления или блокам программы, команды не являются частью языка GPSS/PC. Для спецификации команды достаточно указать несколько первых символов этой команды, обеспечивающих ее уникальность.

22.1. Команда @<FILESPEC>

Эта команда считывает текст модели из файла с именем FILESPEC в рабочий буфер GPSS/PC.

Формат команды:

@<FILESPEC>

GPSS/PC открывает текстовый файл с именем FILESPEC и начинает записывать его в рабочий буфер системы. Если оператор GPSS/PC начинается с некоторого номера, то он вставляется в соответствующее место в рабочем буфере программы. При этом старые операторы заменяются новыми операторами с одинаковыми номерами. Если в тексте модели встречается ошибка, то издается звуковой сигнал, текст диагностического сообщения отображается в окне данных (Data Window), и продолжается запись модели в рабочий буфер GPSS/PC. При этом ошибочная строка не включается в текст считываемой модели. Можно прервать процесс считывания модели, нажав клавишу [HOME] или [ESC]. Можно временно приостановить процесс считывания файла, нажав любую клавишу. Вторичное нажатие клавиши приводит к возобновлению считывания файла. Можно отключить прокрутку текста выводимых операторов программы на экран дисплея, установив соответствующее значение строки

#26 в файле SETTINGS.GPS.

22.2. Команда ANOVA

Команда ANOVA вычисляет доверительные интервалы и выполняет анализ изменения значений, хранящихся в файле результатов (Result File).

Формат команды:

ANOVA <A>,[],[<C>]

A - файловая спецификация для DOS-файла, содержащего результаты моделирования и уровни обработки, организованного в виде столбцов. Такой файл называется файлом результатов (Result File) и создается командой RESULT.

B - колонка результата. Номер колонки в файле результатов, которая содержит зависимые результаты для анализа изменений. По умолчанию - 1. Операнд может быть именем или положительным целым.

C - колонка обработки. Номер колонки в файле результатов, которая содержит уровни обработки, связанные с каждым результатом. По умолчанию - 2. Операнд может быть именем или положительным целым.

Команда ANOVA читает файл результатов, определяемый операндом A, пытается вычислить доверительные интервалы и сделать единым способом анализ изменений величин, определяемых операндом B, согласно уровням обработки, определяемым операндом C. Таблица ANOVA или любые ошибочные сообщения выводятся в окно данных (Data Window) и в рабочий журнал (Session Journal), если он используется. Это выполняется вплоть до удаления любой неправильной записи в файле результатов. Например, можно удалить любые результаты, получившиеся в ходе вмешательства в моделирование способом ручного моделирования или изменения блоков. Это легко сделать, используя текстовый редактор.

Операнд В оценивается и используется как номер колонки в файле результатов, где результирующая величина может быть найдена. Если операнд В опущен или если его значение превышает число 6, то GPSS/PC предполагает, что результат находится в первой или левой колонке файла результатов.

Операнд С оценивается и используется как номер колонки в файле результатов, где могут быть найдены уровни обработки. Если операнд С опущен или если его значение превышает число 6, то GPSS/PC предполагает, что величины уровня обработки находятся во втором столбце файла результатов.

Пример 22.1

ANOVA RESULTS.GPS

Это простейший способ использования команды ANOVA. Он предполагает, что колонка 1 в файле RESULTS.GPS содержит результаты, и колонка 2 содержит уровни обработки. Это обе числовые величины. GPSS/PC выполняет анализ изменений с одним способом классификации и затем выводит результаты в окно данных (Data Window).

Предположим, что содержимое файла результатов следующее:


```
192 2 0 0 0 0 18:19:46 11-18-1989 Replication #1 Treatment A
189 2 0 0 0 0 18:21:14 11-18-1989 Replication #2 Treatment A
191 2 0 0 0 0 18:26:20 11-18-1989 Replication #3 Treatment A
800 3 0 0 0 0 18:30:05 11-18-1989 Replication #1 Treatment B
790 3 0 0 0 0 18:39:55 11-18-1989 Replication #2 Treatment B
794 3 0 0 0 0 18:45:55 11-18-1989 Replication #3 Treatment B
```

По команде ANOVA на экране будет изображение, показанное на рис. 22.1.

ANOVA						
<hr/>						
Source of Variance		Sum of Squares		Degrees of Freedom	Mean Square	F
<hr/>						
Treatments		546016.667		1	546016.667	1294.961
Error		1686.667		4	421.667	1294.961
Error		547703.333		5	421.667	1294.961
<hr/>						
Treatment	Count	Mean	Std.Dev.	Minimum	Maximum	95% Conf.
2	3	189.67	13.05	176	202	32.4
3	3	793.00	25.94	772	822	64.4

Рис. 22.1 Результаты выполнения команды ANOVA

Команда ANOVA выводит описательную статистику в нижней части экрана, даже если не достаточно данных для таблицы ANOVA. Эти величины вычисляются для каждого уровня обработки и включают:

- Treatment - числовой уровень обработки;
- Count - число результатов в уровне обработки;
- Mean - значение результата в уровне обработки;
- Std.Dev. - стандартное отклонение результата в уровне обработки;
- Minimum - наименьшее значение результата в уровне обработки;
- Maximum - наибольшее значение результата в уровне обработки;
- 95% Conf. - аппроксимация половинного размаха для 95% доверительного интервала для значения уровня обработки. Например, приближенный 95% доверительный интервал для величины результата второго уровне (189.67-32.4, 189.67+32.4) или (157.3  222.1).

Необходимо проверить корректность счетчиков, минимумов и максимумов. Если необходимо, следует удалить любые лишние данные в файле результатов с помощью текстового редактора DOS.

Когда файл результатов содержит правильные величины, можно использовать команду ANOVA для вычисления 95% доверительного интервала и для создания таблицы ANOVA. Можно использовать величину F, записанную в таблицу ANOVA, для установления смысла результатов.

Для этого можно использовать таблицу 22.1 критических значений F-распределения с 95% уровнем доверия.

В таблице ANOVA (таблица 22.1) определены данные о свободных ошибках и уровнях (Degrees of Freedom of the Error and Treatment), которые соответственно равны 4 и 1. Если взять критическую F величину из ряда 4 колонки 1 таблицы 22.1, то критическая величина F с 95% уровнем равна 7.71.

Поскольку вычисленная F-величина 1294 больше чем 7.71, делаем вывод, что различия в уровнях обработках являются значимыми.

Если результаты не значимы, следует предусмотреть увеличение длины прогонов моделирования, либо изменить источники случайных чисел. Это часто уменьшает уровень ошибки и является достаточным для установления действительных результатов обработки.

Если нужно исключить результаты стартовых условий из вашего последнего моделирования, вы должны использовать команду RESET до начала периода сбора статистики. Команда PLOT полезна для наблюдения сходимости моделирования к устойчивому состоянию.

Таблица 22.1

Degrees of Freedom of Treatment						
Degrees of Freedom of Error	1	2	3	4	5	6
1	161	200	216	225	230	234
2	18.5	19.0	19.2	19.2	19.3	19.3
3	10.1	9.55	9.28	9.12	9.01	8.94
4	7.71	6.94	6.59	6.39	6.26	6.16
5	6.61	5.79	5.41	5.19	5.05	4.95
6	6.00	5.14	4.76	4.53	4.39	4.28
7	5.59	4.74	4.35	4.12	3.97	3.87
8	5.32	4.46	4.07	3.83	3.69	3.58
9	5.12	4.27	3.86	3.63	3.48	3.37
10	4.97	4.40	3.71	3.48	3.33	3.22
11	4.84	3.98	3.59	3.36	3.20	3.10
12	4.74	3.89	3.49	3.26	3.10	3.00
13	4.67	3.81	3.41	3.18	3.03	2.92
14	4.60	3.74	3.34	3.11	2.96	2.85
15	4.54	3.68	3.29	3.06	2.90	2.79
16	4.49	3.63	3.24	3.01	2.85	2.74
17	4.45	3.59	3.20	2.97	2.81	2.70
18	4.41	3.56	3.16	2.93	2.77	2.66
19	4.38	3.52	3.13	2.90	2.74	2.63
20	4.35	3.49	3.10	2.87	2.71	2.60

22.3. Команда CONTINUE

Эта команда предназначена для возобновления прерванного процесса моделирования. Формат команды:

CONTINUE

Команда возобновляет процесс моделирования, который был ранее прерван или остановлен командой STOP. Моделирование считается остановленным, когда встречается условие, установленное командой STOP или STEP. Команда CONTINUE продолжает процесс моделирования, но не исключает условие STOP, т.е., если это условие встретится вновь в модели, то оно снова сработает. Условие STOP может быть исключено из модели опцией OFF команды STOP.

Моделирование считается прерванным, если была нажата клавиша [Home] или [Esc]. В этом случае команда CONTINUE продолжает процесс моделирования.

Команда **CONTINUE** продолжает процесс моделирования, если счетчик завершения является положительным числом, или, в противном случае, прекращает процесс моделирования. При возобновлении процесса моделирования осуществляется перенумерация блоков модели, если были вставлены новые блоки или удалены некоторые из них; переопределяется размещение блоков и затем начинается процесс планирования транзактов. Генераторы случайных чисел не сбрасываются.

22.4. Команда DELETE

Команда **DELETE** удаляет блоки модели из рабочего буфера GPSS/PC. Формат команды:

DELETE <A>,[]

A - номер первого удаляемого блока;

B - номер последнего удаляемого блока.

Операнды **A** и **B** могут быть положительными десятичными числами.

Пример 22.2

DELETE 122,145

Эта команда удаляет все блоки, номера которых содержатся в интервале 122-145.

Если операнд **B** отсутствует, то удаляется только один блок с номером **A**.

Для удаления текста всей модели из рабочего буфера GPSS/PC можно указать номера границ блоков, заведомо большие реально существующих.

22.5. Команда DISPLAY

Команда **DISPLAY** выводит в окне данных весь текст модели или его часть, находящуюся в рабочем буфере GPSS/PC.

Формат команды:

DISPLAY [<A>],[]

A - номер первой отображаемой строки программы;

B - номер последней отображаемой строки программы.

Операнды **A** и **B** могут быть положительным десятичным числом.

Для отображения всей программы применяется команда **DISPLAY** без операндов.

Если отсутствует операнд **A**, но имеется операнд **B**, то отображение начинается с первого блока программы и до блока с номером **B**. Если используется только операнд **A**, то отображается только одна строка программы с номером **A**.

Для приостановки процесса прокрутки текста на экране дисплея необходимо нажать любую клавишу. Повторное нажатие клавиши вызывает

продолжение прокрутки. Для постраничной выдачи информации необходимо изменить строку #26 файла SETTINGS.GPS.

22.6. Команда DOS

Эта команда осуществляет временное прерывание процесса моделирования, выход в DOS с возможностью возврата в соответствующую точку прерванного процесса по команде EXIT.

Формат команды:

DOS

Команда DOS сначала пытается найти файл DOS COMMAND.COM на устройстве для свопинга прерванной модели. Необходимо подготовить устройство для свопинга модели, разместив копию файла COMMAND.COM в корневом каталоге устройства для свопинга модели. Если GPSS/PC находит COMMAND.COM, то состояние текущей модели записывается на устройство для свопинга. Можно специфицировать расположение устройства для свопинга корректировкой строки #41 в файле SETTINGS.GPS.

Файл выгрузки (свопинга) - это временный файл с именем SWAPGPSS. Файл также содержит один символ расширения, который изменяется в зависимости от имеющейся в распоряжении памяти. Файл выгрузки будет содержать часть текущей модели, пока происходит работа с MS DOS. Файл SWAPGPSS уничтожается после возврата в GPSS/PC.

Размер файла выгрузки может быть ограничен установкой строки #42 в файле SETTINGS.GPS. Это дает возможность использовать дискеты в качестве устройства выгрузки.

22.7. Команда EDIT

Команда EDIT позволяет модифицировать программу, находящуюся в рабочем буфере модели.

Формат команды:

EDIT <A>

A - номер строки программы. Операнд может быть положительным десятичным числом.

Команда EDIT сначала восстанавливает копию указанного оператора из программы, находящейся в рабочем буфере GPSS/PC. Можно сделать изменения в этой копии, включая и номер блока. После нажатия клавиши [Enter] копия блока помещается в соответствующее место текста модели, находящейся в рабочем буфере. В ходе модификации блока модели одновременно осуществляется синтаксический анализ модифицируемого блока. При выявлении ошибки выдается звуковой сигнал.

Программа EDIT запрещает ввод ошибочного блока программы модели. В режиме TYPE-OVER MODE осуществляется изменение строки программы без

вставки новых символов. Если нажать клавишу [Ins], то EDIT переходит в режим INSERT MODE, что дает возможность вставлять новые символы в модифицируемую строку.

EDIT использует специальные клавиши или комбинации клавиш:

[Ins] - переключить режим вставки (INSERT MODE) и обратно (TYPE-OVER MODE);

[<-] - переместить курсор влево на один символ;

[->] - переместить курсор вправо на один символ;

[Ctrl],[<-] - переместить курсор влево на одно слово;

[Ctrl],[->] - переместить курсор вправо на одно слово;

[Del] - удалить один символ;

[End] - переместить курсор к концу строки;

[Home] - выйти из EDIT без изменения строки.

22.8. Команда EVENTS

Команда EVENTS отображает содержимое списка будущих и текущих событий в окне данных.

Формат команды:

EVENTS

Команда EVENTS переходит к окну данных и высвечивает транзакты с содержимым их параметров в таком порядке:

- 1) в списке текущих событий (CEC) активный транзакт с наивысшим приоритетом;
- 2) транзакты с более низкими приоритетами ;
- 3) все транзакты из списка будущих событий (FEC) в порядке их расположения в таблице моделируемых событий (BDT).

Для приостановки просмотра списков CEC и FEC достаточно нажать любую клавишу. Вторичное нажатие клавиши приводит к продолжению просмотра списков CEC и FEC. Для постраничного просмотра информации необходимо скорректировать строку с номером #26 файла SETTINGS.GPS.

В списке CEC имеется информация:

- XACT NUMBER - номер сообщения;

- PRI - приоритет сообщения;

- M1 - время появления транзакта в блоке GENERATE, или время прохождения сообщения или его "родителя" через блок MARK без параметров;

- CURRENT - номер блока, где находится транзакт; 0 - если транзакт еще не вошел ни в один блок;

- NEXT - номер блока, в который должно войти сообщение;

- PARAMETER - номер или имя параметра сообщения;

- VALUE - значение параметра.

В списке FEC дополнительно появляется атрибут:

BDT - таблица моделируемых событий - абсолютное модельное время выхода транзакта из FEC (и переход транзакта в CEC).

22.9. Команда GROUPS

Эта команда выводит содержимое группы в виде номеров транзактов или численных величин. Формат команды:

GROUPS

Команда GROUPS переходит в окно данных, где выводит содержимое группы сообщений в виде номеров транзактов или содержимое числовых групп в виде соответствующих чисел.

22.10. Команда MICROWINDOW

Эта команда открывает или закрывает микроокна.

Формат команды:

MICROWINDOW <A>,[],[<C>] ;COMMENT

A - номер микроокна от 1 до 4 включительно.

B - аргумент микроокна. Операнд может быть СЧА или СЧА*<параметр>.

C - состояние микроокна. Операнд может быть ON или OFF. По умолчанию ON.

COMMENT - комментарий к микроокну, до 8 символов. Поле является необязательным.

Пример 22.3

MICROWINDOW 4,AC1 ; CLOCK

Команда открывает микроокно 4, которое высвечивает абсолютное модельное время. Микроокно комментируется строкой CLOCK.

Значения СЧА, специфицированные операндом **B** команды MICROWINDOW, постоянно обновляются в ходе моделирования, за исключением случая, когда открыто окно данных. В качестве СЧА нельзя использовать параметры транзактов. Рекомендуется значения параметров транзактов заносить в сохраняемые величины, которые можно выводить в MICROWINDOW.

22.11. Команда PLOT

Команда PLOT предназначена для отображения в окне данных диаграммы изменения значений выбранного СЧА. Диаграмма располагается в правом

верхнем квадранте Декартовой системы координат. На горизонтальной оси системы координат отображается интервал времени отслеживания изменений значений выбранного СЧА. На вертикальной оси - значения, принимаемые СЧА.

Формат команды:

PLOT <A>,,<C>,<D> ;COMMENT

A - аргумент отображения. Операнд - СЧА.

B - максимальная величина значений аргумента. Наибольшее значение СЧА, появляющееся при отображении.

C - начальное время отображения. Операнд - 0 или положительное целое.

D - конечное время отображения. Операнд - положительное целое.

COMMENT - комментарий, являющийся заголовком отображения. Может быть любым текстом. Поле комментария может содержать до 34 символов.

Команда **PLOT** инициализирует оси отображения и строит диаграмму изменения значений СЧА в процессе моделирования на интервале времени от **C** до **D**.

Поле комментария используется для заголовка отображения.

Одно или два отображения (действия команды **PLOT**) могут быть активны в любой момент времени. Если поступает вторая команда **PLOT**, первое отображение перемещается вверх в окне данных. Содержимое окна может быть стерто нажатием клавиши [Esc] дважды.

Если величина СЧА выходит за пределы отображения, она определяется как 0 или как максимальное значение Y.

Если системное время выходит за пределы, определенные в команде **PLOT**, значение СЧА игнорируется.

Если СЧА неправильно определены, они могут вызвать ошибку выполнения.

Первоначально, величины СЧА принимают значение 0. Можно определить другое начальное значение через переменные.

Пример 22.4

```
20    GENERATE    300,100           ;Create next customer.
30    QUEUE      BARBER             ;Begin queue time.
40    SEIZE      BARBER             ;Own or wait for barber.
50    DEPART     BARBER             ;End queue time.
60    ADVANCE    600,200           ;Haircut takes a few min.
70    ALTDEST    RELEASE BARBER     ;Haircut done.
80    TERMINATE  1                  ;Customer leaves.
90    PLOT    q$barber,200,0,80000  ;Waiting line for Facil.
100   START      200
```

22.12. Команда RENUMBER

Эта команда перенумеровывает все блоки программы модели, находящиеся в рабочем буфере GPSS/PC.

Формат команды:

RENUMBER [

A - номер первого блока. Может быть положительным десятичным числом, содержащим не более 6 знаков.

B - шаг(приращение) нумерации. Может быть положительным десятичным числом, содержащим не более 6 знаков.

Пример 22.5

RENUMBER 10,1

Эта команда нумерует блоки в следующем порядке: 10.1, 10.2,...

Если операнды A и B не используются, то по умолчанию нумерация начинается с номера 10 с шагом 10.

22.13. Команда REPORT

Эта команда устанавливает имя текстового файла DOS для записи в него стандартной выходной статистической информации GPSS/PC в специальном формате.

Формат команды:

REPORT <A>,[] ;COMMENT

A - имя файла.

B - признак немедленной записи информации в файл. Может иметь значение NOW.

COMMENT - строка комментариев в файле.

Пример 22.6

REPORT B:PRICE20 ;Sales of Widgets \$19.95

По этой команде стандартная статистика GPSS/PC должна быть выведена в файл с именем PRICE20, который находится на устройстве B. Указанная статистическая информация будет озаглавлена строкой SALES OF WIDGETS \$19.957.

Команда REPORT только подготавливает файл для вывода выходной информации GPSS/PC в специальном (не отформатированном) формате.

Вывод информации осуществляется непосредственно в файл в ходе моделирования только, если операнд **B** определен как NOW. В противном случае выходная информация будет выведена в файл лишь при нулевом либо отрицательном значении счетчика завершения моделирования. При отсутствии команды REPORT выходная статистическая информация после завершения

моделирования будет автоматически выведена в файл REPORT.GPS. Имя стандартного выходного файла может быть изменено установкой соответствующего имени в файле SETTINGS.GPS. Каждый новый выходной отчет заменяет в файле старый отчет. При необходимости сохранения нескольких выходных отчетов в ходе моделирования в команде REPORT специфицируют различные имена выходных файлов. Статистическая информация находится в выходном файле в нечитабельном виде. Для представления этой информации в формате, удобном для просмотра, используется программа с именем GPSSREPT.

Программа GPSSREPT должна быть запущена либо после завершения работы в GPSS/PC (команда END), либо после временного выхода из GPSS/PC по команде DOS. Для отказа от вывода выходной статистической информации необходимо специфицировать операнд **B** оператора START как NP.

22.14. Команда RESULT

Команда RESULT подготавливает результаты моделирования в файле результатов (Result File) для последующей их обработки командой ANOVA.

Формат команды:

RESULT <A>,,[<C>],[<D>],[<E>],[<F>],[<G>] ;COMMENT

A - файловая спецификация для DOS-файла, содержащего результирующие величины и уровни их обработки. Такой файл называется файлом результатов (Result File).

B - имя или номер ячейки сохраняемых величин, содержащей некоторую отслеживаемую характеристику. Операнд имя или положительное целое.

C - уровень обработки для класса 1, который помещается во 2-ой столбец файла результатов. Операнд имя или положительное целое.

D - уровень обработки для класса 2, который помещается в 3-ий столбец файла результатов. Операнд имя или положительное целое.

E - уровень обработки для класса 3, который помещается в 4-ый столбец файла результатов. Операнд имя или положительное целое.

F - уровень обработки для класса 4, который помещается в 5-ый столбец файла результатов. Операнд имя или положительное целое.

G - уровень обработки для класса 5, который помещается в 6-ой столбец файла результатов. Операнд имя или положительное целое.

COMMENT - комментарий записи в файле результатов.

Пример 22.7

RESULT RESULTS.GPS,AVERAGE,3 ;Replication #1 Treatment 3

Это простейший вариант использования команды RESULT. Он предполагает, что определен только один класс обработки и столбец 2 записи файла результатов содержит 3-ий уровень обработки. GPSS/PC определяет результат

путем взятия величины, связанной с ячейкой сохраняемых величин с именем AVERAGE. Если такая ячейка не существует, она создается с нулевым значением. Затем GPSS/PC создает запись с результатом в первом столбце и уровнем обработки во втором.

Если ячейка AVERAGE содержит значение 176, то GPSS/PC к DOS-файлу RESULTS.GPS добавляет строку, аналогичную следующей:

176 3 0 0 0 0 19:19:46 11-18-1989 Replication #1 Treatment 3

Если файл RESULTS.GPS не существует в текущей директории, то GPSS/PC создает его.

Команда RESULT оценивает операнд В и использует результирующее значение как номер ячейки. Она выбирает значение ячейки и затем строит одну строку-запись для добавления к файлу результатов, определяемого операндом А. Временная метка и комментарий автоматически включаются в запись.

Операнды от С до G позволяют определить уровень обработки для пяти классификаций обработок. Необходимо определить уровень обработки по крайней мере для одного класса обработки. Для определения уровня обработки можно использовать числовое значение или имя. Рекомендаций по использованию числовых уровней обработки нет.

Каждый уровень обработки должен представляться более чем одним, предпочтительнее несколькими, откликами. Отклики получают за счет повторных прогонов модели, изменяя только источники для генератора случайных чисел. Оператор RMULT может использоваться для контроля откликов.

Для анализа изменений в файле результатов можно использовать команду ANOVA GPSS/PC. Перед анализом файла результатов следует удалить любые лишние или неверные данные с помощью текстового редактора. Файл ответов можно редактировать в середине прогона, используя DOS команду GPSS/PC.

22.15. Команда SAVE

Команда переписывает текст модели, содержащийся в рабочем буфере GPSS/PC в указанный текстовый файл MS DOS.

Формат команды:

SAVE <A>,[],[<C>]

A - имя файла, в который записывается программа модели.

B - номер блока модели, начиная с которого осуществляется запись программы в файл.

C - номер блока модели, по какой включительно осуществляется запись программы в файл.

При отсутствии операндов В и С в файл записывается весь текст программы модели. При отсутствии операнда В и наличии операнда С в файл записывается текст модели начиная с первого оператора и по оператор,

указанный операндом **C**. При отсутствии операнда **C** и наличии операнда **B** в файл записывается только один оператор программы модели с номером, указанным операндом **B**. Если будет использована команда **END** с не сохраненной рабочей моделью, то GPSS/PC выдает предупреждающее сообщение с возможностью предотвратить выход из системы. При отсутствии достаточного места на диске рабочая программа может быть сохранена в нескольких файлах частями. В дальнейшем каждая часть текста может быть восстановлена по отдельности использованием оператора **@<filespec>**.

22.16. Команда **SHOW**

Команда отображает значение выражения на экране дисплея.

Формат команды:

SHOW X

X - выражение.

Пример 22.8

SHOW LOG(Q\$BARBER)

Указанная команда вычисляет натуральный логарифм от СЧА **Q\$BARBER** и выводит результат на экран дисплея.

Выражения в команде **SHOW** могут быть арифметического и логического типа. Арифметические выражения оцениваются с двойной точностью.

22.17. Команда **STEP**

Команда задает возможность прерывания процесса моделирования при прохождении указанного количества блоков.

Формат команды:

STEP <A>

A - счетчик прохождения блоков. Операнд может быть положительным целым.

Пример 22.9

STEP 1

Процесс моделирования прерывается при прохождении одного блока модели. Когда процесс моделирования прерывается, то GPSS/PC переходит в состояние, вызываемое клавишами **[Home]** или **[Esc]**. В верхней части экрана появляется сообщение трассировки. Для использования команды **STEP** в модели должно быть активное сообщение. Модель, которая стартовала с использованием команды **STEP**, не может быть завершена при нулевом счетчике завершения.

Моделирование завершится при прохождении указанного в операнде А команды STEP количества блоков.

Когда команда STEP выполняется, то

- счетчик завершения не устанавливается;
- если необходимо, перенумеровываются блоки;
- все блоки, генерирующие сообщения, приводятся в исходное состояние;
- генераторы случайных чисел не сбрасываются.

22.18. Команда STOP

Команда устанавливает или снимает условие прерывания моделирования.

Формат команды:

STOP [<A>],[],[<C>]

A - номер транзакта. Операнд - положительным целым.

B - номер блока. Операнд - положительным целым.

C - флаг состояния команды. Операнд - ON или OFF. По умолчанию ON.

Пример 22.10

STOP 100,52

Эта команда устанавливает условие прерывания модели при входе транзакта с номером 100 в блок с номером 52.

Команда STOP с опцией ON устанавливает условие прерывания моделирования, но не стартует модель. Для запуска моделирования должна использоваться последовательность команд START, STEP, CONTINUE.

Когда выполняется прерывание по условию команды STOP, то система переходит в состояние, вызываемое нажатием клавиш [Home] или [Esc].

В верхней части экрана высвечивается трассируемое сообщение. Команда CONTINUE позволяет выйти из прерывания и продолжить моделирование, однако условие прерывания, введенное ранее командой STOP, остается включенным. Для отключения условия прерывания необходимо войти в команду STOP с флагом OFF.

Если отсутствует операнд А команды STOP, то любой транзакт может вызвать условие прерывания. Если отсутствует операнд В команды STOP, то любой блок модели удовлетворяет условию прерывания. Команда STOP без операндов вызывает немедленное прерывание процесса моделирования.

22.19. Команда USERCHAINS

Команда отображает содержимое списков пользователя на экране дисплея.

Формат команды:

USERCHAINS

Команда USERCHAINS переходит в окно данных и отображает в нем все члены списков пользователя.

22.20. Команда WINDOW

Команда WINDOW предназначена для открытия виртуальных окон GPSS/PC.

Формат команды:

WINDOW <A>,[],[<C>],[<D>]

Операнд **A** позволяет выбрать окно из набора окон GPSS/PC.

Операнд может быть: BLOCKS, DATA, FACILITIES, MATRICES, POSITIONS, STORAGES, TABLES.

Операнд **B** определяет объект, наиболее близко расположенный к левому верхнему углу окна в случае одновременного отображения в окно нескольких объектов одного типа.

В окнах таблиц и матриц, как правило, отображается один объект за один раз. При наличии нескольких объектов типа таблиц и матриц полный просмотр можно осуществить с помощью управляющих клавиш [PgUp], [PgDn], [End]. Операнд **B** не используется для окон DATA и POSITIONS.

Операнд **C** определяет номер верхней строки отображаемой области для визуализации в окне MATRICES или в окне POSITIONS. Этот операнд не используется в окнах BLOCKS, DATA, FACILITIES, STORAGE, TABLES. Если операнд **C** не указан, то принимается значение по умолчанию: 1 - в окне MATRICES или 0 - в окне POSITIONS.

Операнд **D** определяет номер левой колонки отображаемой области для визуализации в окнах MATRICES, TABLES, POSITIONS. Операнд не используется в окнах DATA, FACILITIES, STORAGES. Если операнд **D** не указан, то по умолчанию принимается значение: 1 - в окне MATRICES или TABLES, 0 - в окне POSITIONS.

Все окна, включая окно DATA, позволяют использовать микроокна. Если одно или более микроокон остаются открытыми, они остаются в правой части виртуального окна, до тех пор, пока не будут закрыты очередной командой MICROWINDOW.

Использование управляющих клавиш позволяет осуществлять просмотр информации, не уместящейся в одном окне. Так, клавиша [PgUp] осуществляет листание информации в окне назад, клавиша [PgDn] - листание информации в окне вперед. Клавиша [End] высвечивает в окне данных последнюю порцию информации.

Окно POSITIONS может перемещаться подобно кинокамере по виртуальному экрану. Для этого используется одновременное нажатие следующих комбинаций клавиш:

- [Ctrl],[PgUp] - двигает окно вверх на величину, равную высоте окна;
- [Ctrl] [PgDn] - двигает окно вниз на величину, равную высоте окна;

- [Ctrl],[>-] - двигает окно вправо на величину, равную ширине окна;
- [Ctrl],[<-] - двигает окно влево на величину, равную ширине окна.

1. Диалоговые возможности

Взаимодействие пользователя с системой осуществляется в режиме активного диалога. Для этого в системе предусмотрен ряд средств, к которым относятся:

- команды GPSS/PC;
- виртуальные окна;
- редактор исходных текстов моделей на языке GPSS;
- микроокна.

Под виртуальным окном (имеется шесть графических окон и одно псевдографическое окно) понимается отображение информации о состоянии отдельных объектов на экране дисплея. Информация может отображаться как статически, представляя собой снимок состояния объекта на момент ее получения, так и динамически, когда изменяется состояние объекта, в процессе его использования при моделировании.

Пользователь может активно вмешиваться в процесс отображения информации в виртуальном окне, выдавая различные команды из набора команд GPSS/PC и/или используя специальные поля команд, имеющиеся в ряде виртуальных окон GPSS/PC.

Микроокна - это небольшие графические окна, в которых отображается текущее значение любой переменной состояния системы и заголовков. В пределах графического виртуального окна может быть открыто до четырех микроокон. Каждое из четырех микроокон имеет фиксированную позицию и размеры, которые невозможно изменить.

Меняется только содержимое микроокон. Микроокна открываются командой MICROWINDOW. Возможно многократное использование одного и того же микроокна для получения различной информации. В процессе моделирования содержимое каждого микроокна меняется динамически, при изменении значений связанных с микроокнами переменных.

Для использования виртуальных графических окон в полном объеме необходимо наличие у персональной ЭВМ (ПЭВМ) пользователя цветного графического дисплея с адаптером VGA или его аналогом. Предусмотрена работа в двух режимах адаптера VGA: с высокой разрешающей способностью дисплея (350 линий) и со средней разрешающей способностью дисплея (200 линий). Для задания используемого в ПЭВМ режима работы адаптера необходимо установить признак режима в строке файла SETTINGS.GPS с ключом #32. Комментарии приводятся в самом файле SETTINGS.GPS. В этом же файле устанавливаются атрибуты цветности различных полей виртуальных окон GPSS/PC.

GPSS/PC представляет в распоряжение пользователя семь виртуальных окон:

- псевдографическое окно данных (DATA WINDOW);
- графическое окно блоков (BLOCKS WINDOW);
- графическое окно устройств (FACILITIES WINDOW);
- графическое окно памяти (STORAGES WINDOW);
- графическое окно матриц (MATRICES WINDOW);
- графическое окно позиций (POSITIONS WINDOW);
- графическое окно таблиц (TABLES WINDOW);

В левой части каждого из графических окон отводится место для четырех микроокон, для случая, если пользователь пожелает их использовать. Микроокна имеют фиксированные номера позиций 1-4. Номера увеличиваются снизу вверх.

Все виртуальные окна GPSS/PC могут быть открыты с помощью команды WINDOWS или одновременным нажатием на клавиатуре клавиши [Alt] и клавиши, соответствующей первому символу в имени окна.

При наличии информации об объектах, которая не уместилась в одном графическом окне, возможно "листание" в окне этой информации, с использованием клавиш [PgDn], [PgUp], [End] дополнительной клавиатуры. Клавиша [Home] служит для отказа от работы в текущем графическом окне, а при ее использовании в процессе моделирования (также как и клавиш [Ins] и [Esc]), происходит прерывание процесса имитации.

Нажатие клавиши [Home] при прерванном, завершенном, либо не начатом процессе моделирования, вызывает очистку экрана и возврат курсора в начальную позицию командной строки.

Работая с графическими окнами, пользователь может одновременно нажатием клавиш [Alt + L] запросить трассировку прохождения транзактов по блокам модели в процессе имитации. Это вызовет при имитации появление в верхней части графического окна строки трассировки, в которой динамически отображается информация о текущем модельном времени (Time), номере участвующего в трассировке транзакта (XACT), номере пройденного им блока (leaves), номере блока, в который он входит (enters) и типе последнего блока. Повторным нажатием комбинации клавиш [Alt + L] пользователь может отказаться от дальнейшей трассировки модели. Строка трассировки, при этом, исчезает из окна.

Пользователь, находясь в одном из графических окон, взаимодействует с системой путем набора команд GPSS/PC в командной строке и ввода их клавишей [CR] ([Enter]), либо путем выбора одной из команд, связанных с окном, высвеченных в поле команд окна, и ввода выбранной команды путем нажатия клавиши [Ins] ([Insert]). Для выбора команд из поля команд окна используется символ "+" - курсор окна (указатель точки в окне). Курсором окна можно управлять, перемещая его в любую позицию окна с помощью клавиш управления движением курсора (стрелки вверх, вниз, влево, вправо). При

работе в окне блоков и окне позиций с помощью курсора окна можно осуществлять выбор объектов модели для последующего манипулирования ими.

При вводе команд, не ориентированных на работу в текущем окне, система автоматически открывает нужное окно, меняя картинку на экране дисплея.

24. Организация связи системы GPSS/PC с другими программами

Блок HELP предназначен для связи с пользовательскими программами, написанными на языке Фортран.

Когда транзакт входит в блок HELP, любая сохраняемая величина, определяемая в операндах блока, передается в COMMON-область HELP-программы на Фортране. Программа на Фортране выполняется, и значения (возможно измененные) возвращаются в сохраняемую величину GPSS.

Существует много возможных видов применения HELP-блока.

Он может:

- быть использован для подключения статистической обработке результатов моделирования, организации аналитико-имитационных моделей, чтения данных в среде моделирования и записи возвращаемых данных и команд;
- собирать информацию для запуска того или иного режима мультипликации, при этом состояние реальной системы может отображаться на экране;
- создать диалог для менее опытного пользователя и дать высокое разрешение отображения.

Существует и другие возможности организации связи GPSS программы с другими программами и пакетами статистической обработки. Для этого можно использовать запись результатов моделирования в специальный файл с помощью команды RESULT.

Команда RESULT подготавливает результаты моделирования в файле результатов (Result File), для последующей их обработки командой ANOVA.

Для формирования временного ряда в GPSS-программу вставляют специальные строки для запоминания ряда и вывода его в файл GPSS-отчета. Ниже приведена программа модели СМО вида М/М/1, в которую вставлены строки 41,82,83,84,121-124. Начало цикла регенерации отмечается флажком 1111, если циклы регенерации существуют. В результате работы программы формируется столбец из ячеек сохранения, который может читаться пакетом статистики. Это позволяет использовать эти данные для построения параметрической модели [7], для формирования функции распределения, а также для передачи результатов моделирования в статистические пакеты. Для этого необходимо написать программу, читающую формируемый столбец сохраняемых величин, или использовать текстовый редактор для работы с файлом статистического отчета.

```

30 EXP    FUNCTION    RN1,C24
0,0/.100,.104/.200,.222/.300,.355/.400,.509
.500,.690/.600,.915/.700,1.200/.750,1.380
.800,1.600/.840,1.830/.880,2.120/.900,2.300
.920,2.520/.940,2.810/.950,2.990/.960,3.200
.970,3.500/.980,3.900/.990,4.600/.995,5.300
.998,6.200/.999,7/1,8
39 BAD2   TABLE     M1,0,500,30
41 *
42        INITIAL     X1,1                      ;N первого значения ряда
50        GENERATE     200,FN$EXP
51 INP    QUEUE       BAD1
60        SEIZE       BAD1
61        DEPART      BAD1
70        ADVANCE     150,FN$EXP
80 OUT    RELEASE     BAD1
82        TEST NE     N$OUT,N$INP,CIKL          ; есть ли циклы?
83        SAVEVALUE   1+,1                      ; N следующего значения
84        SAVEVALUE   X1,M1                     ; исследуемая переменная
102       TABULATE    BAD2
120       TERMINATE
121 CIKL  SAVEVALUE   1+,1                      ; первое значение в цикле
122       SAVEVALUE   X1,M1                     ; первое значение в цикле
123       SAVEVALUE   1+,1                      ; N для флажка
124       SAVEVALUE   X1,1111                   ; флажок начала цикла
125       TABULATE    BAD2
126       TERMINATE   1

```

25. Задания для самостоятельной работы

Задание 1. Моделирование разливочной линии.

Пусть имеется некоторая автоматизированная линия по выпуску баночек фруктового сока. Пустые баночки для фруктового сока поступают в накопитель 1 автоматизированной линии каждые $A \pm B$ сек. После этого в них автоматически заливается сок (одновременно может заливаться одна банка, на что тратится F сек.). Затем банки поступают в накопитель 2 для операции закупорки и приклеивания этикеток (C сек. на банку, в один момент времени может обрабатываться одна банка) и попадают в накопитель 3 для следующей операции. Структура модели изображена на рис. 25.1.

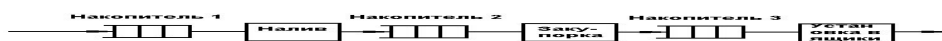


Рис. 25.1

В конце конвейера осуществляется установка банок в ящики. Время установки одной банки - равномерно распределенная случайная величина на интервале $D \pm E$ секунд. В один момент времени может устанавливаться в ящик не более двух банок.

Начальные условия: в начале смены в накопителе 2 находится G бутылок, в накопителе 3 - K бутылок..

Оценить какие размеры должны иметь накопители с номерами, указанными в таблице 25.1 для каждого варианта. Промоделировать работу линии в течение одной смены - N часов.

Таблица 25.1

<i>Параметры</i>	<i>Значения</i>		
	<i>Вариант 1</i>	<i>Вариант 2</i>	<i>Вариант 3</i>
$A \pm B$	3.5 ± 1.1	4.5 ± 2.0	3.2 ± 1.3
F	1.5	1.2	2.3
C	1.6	1.3	2.4
$D \pm E$	2.0 ± 0.8	1.7 ± 0.5	2.6 ± 0.4
G	20	26	35
K	36	36	30
N	8	8	7
№ анализируемого накопителя	1,3	2,3	1,2

Задание 2. Моделирование контроля и настройки телевизоров.

Собранные телевизоры после сборки проходят серию испытаний на станции технического контроля. Если оказывается, что функционирование телевизора ненормально, то отбракованный телевизор переправляют в цех наладки, где заменяют неисправные блоки. После наладки телевизор возвращают на станцию контроля и снова проверяют. Телевизионные приемники уходят с последней

станции после одной или нескольких проверок в цех упаковки. Структурная схема участка показана на рис. 25.2.

Телевизоры попадают на станцию технического контроля каждые $A \times B$ мин. На станции работают C контролеров одинаковой квалификации. Операция контроля одного телевизора состоит из двух этапов:

1. Проверка типа 1. На нее каждому из контролеров требуется $D \times E$ мин.
2. Проверка типа 2. Для ее осуществления на всех C контролеров имеется один прибор, длительность тестирования телевизора этим прибором занимает F мин.

Примерно $G\%$ телевизоров проходят проверку успешно и попадают в цех упаковки. Остальные $K\%$ попадают в цех наладки, в котором находится 1 рабочий - наладчик. Время наладки блока распределено по экспоненциальному закону со средним N мин.

В таблице 25.2 приведены варианты заданий.

Написать на GPSS модель функционирования этого подразделения производственной линии. Моделирование провести для 8 часов работы.

Оценить сколько мест на стеллажах необходимо предусмотреть на входе станции контроля и в цехе наладки.

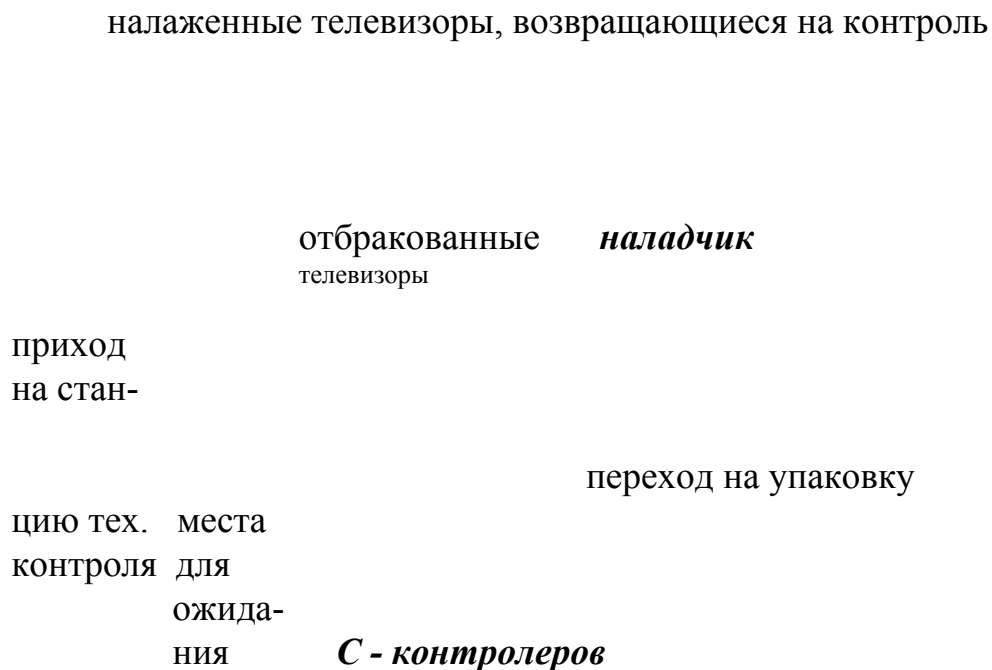


Рис. 25.2

Таблица 25.2

<i>Параметры</i>		<i>Значения</i>	
	<i>Вариант 1</i>	<i>Вариант 2</i>	<i>Вариант 3</i>
A ± B	5.5 ± 2.0	7.0 ± 1.5	6.5 ± 2.0
D ± E	9.0 ± 3.0	5.0 ± 6.0	12.0 ± 3.0
F	1.2	1.5	1.0
G	85	95	85
K	15	5	15
N	30	35	30

Задание 3. Моделирование работы кафе.

В небольшом кафе работают 2 официантки (А и Б), обслуживая по N четырёхместных столиков. Официантка А пользуется большей популярностью, чем Б (приходя в кафе, клиент сядет за столик Б только в том случае, если все места у А заняты). Клиенты приходят в кафе через $a \pm b$ мин и если не застанут свободных мест, становятся в очередь.

Когда клиент садится за освободившееся место, он ждет, пока к нему не подойдет официантка. Официантка принимает у него заказ. Время приёма заказа у А занимает $c \pm d$ с, у Б $e \pm f$ с. Время выполнения заказа обеими официантками составляет $g \pm h$ с. После чего клиент обедает в течение $k \pm m$ мин. и уходит. Официантки обслуживают клиентов по принципу FIFO. В каждый момент времени могут обслуживать не более одного клиента. Официантки приняв заказ, сразу же его выполняют.

В таблице 25.3 приведены варианты заданий.

Оценить время ожидания в очереди и время, которое клиент проводит за столиком кафе. Промоделировать работу кафе в течение 10 часов.

Таблица 25.3

		<i>Значения</i>	
<i>Параметры</i>	<i>Вариант 1</i>	<i>Вариант 2</i>	<i>Вариант 3</i>
N	5	7	6
$a \pm b$	3 ± 1	4 ± 2	5 ± 2
$c \pm d$	15 ± 5	20 ± 6	30 ± 10
$e \pm f$	17 ± 4	22 ± 6	35 ± 8
$g \pm h$	160 ± 20	180 ± 30	200 ± 50
$k \pm m$	6 ± 2	8 ± 3	9 ± 3

Задание 4. Моделирование работы обрабатывающего цеха.

В обрабатывающий цех через $a \pm b$ мин поступают детали двух типов: с вероятностью p_1 - типа 1, p_2 - типа 2.

Детали типа 1 обрабатываются станком 1 (время обработки $c \pm d$ мин, в каждый момент времени может обрабатываться только одна деталь). Если деталь не соответствует требованиям качества, то с вероятностью p_3 она возвращается для повторной обработки на станок 1, в противном случае она поступает на станок 3.

Детали типа 2 сначала обрабатываются станком 2 (время обработки $e \pm f$ мин, в каждый момент времени может обрабатываться только одна деталь). Если деталь не соответствует требованиям качества, то с вероятностью p_3 она возвращается для повторной обработки на станок 2, в противном случае она поступает на станок 3.

Станок 3 может обрабатывать до g деталей одновременно, время обслуживания $k \pm m$ мин.

Оценить время нахождения детали на обработке в цеху. Промоделировать работу цеха в течение N часов. В таблице 25.4 приведены варианты заданий.

Таблица 25.4

<i>Параметры</i>	<i>Значения</i>		
	<i>Вариант 1</i>	<i>Вариант 2</i>	<i>Вариант 3</i>
$a \pm b$	5 ± 1	6 ± 2	7 ± 2
p_1	0.4	0.5	0.7
p_2	0.6	0.5	0.3
p_3	0.1	0.05	0.075
$c \pm d$	15 ± 5	16 ± 6	14 ± 10
$e \pm f$	8 ± 4	12 ± 6	16 ± 8
G	5	4	3
$k \pm m$	6 ± 2	8 ± 3	9 ± 3
N	10	11	8

Задание 5. Моделирование работы цеха обработки.

На вход некоторого обрабатывающего цеха, состоящего из трех участков, поступает случайный поток деталей. Интервалы поступления распределены экспоненциально со средним значением времени 4 мин. Поступающая деталь с

вероятностью 0,65 имеет тип 1, с вероятностью 0,35 - тип 2. Сначала детали поступают на участок 1, где обрабатываются последовательно одна за другой, время обработки распределено равномерно на интервале от 2 до 5 мин. Детали второго типа имеют больший приоритет при обработке, чем детали первого типа.

Далее, после обработки на участке 1, детали 1-го типа поступают на участок 2, а детали 2-го типа - на участок 3. На участке 2 имеется 3 идентичных станка, время обработки детали станком распределено экспоненциально со средним значением 11 мин. На участке 3 имеется 2 станка (время обработки на каждом из них распределено экспоненциально со средним значением 7 мин). Промоделировать работу цеха в течение 40 часов со сбором статистики об очереди деталей перед третьим участком.

Задание 6. Моделирование работы цеха.

В цех поступает пуассоновский поток деталей с интенсивностью 20 деталей в час. С вероятностью 0,4 деталь поступает на участок 1, а с вероятностью 0,6 - на участок 2. На первом участке детали могут обрабатываться равновероятно на одном из двух станков. Время обслуживания - экспоненциальное со средним значением 48 мин. На втором участке детали обрабатывают одним станком, за время, равномерно распределенное на интервале 2 ± 1 мин. После обработки на одном из участков 1 или 2 детали направляются на участок 3 с одним станком, время обработки - экспоненциально распределенное со средним значением 2 мин.

Промоделировать обработку 1000 деталей. **Определить** количество деталей, прошедших через участок 1 и максимальную длину очереди перед участком 3.

Задание 7. Моделирование работы СМО.

На вход одноканальной обслуживающей системы поступает поток заявок, интервал поступления равномерно распределен на интервале от А до В единиц времени. С вероятностью p_1 заявка имеет тип 1, с вероятностью p_2 - тип 2. Заявки типа 2 при выборе из очереди имеют больший приоритет, чем заявки типа 1. Время обслуживания заявки прибором - экспоненциально распределенное со средним t_1 единиц времени для заявки типа 1, t_2 - для заявки типа 2. Промоделировать обслуживание К заявок. **Оценить** длину очереди. В таблице 25.5 приведены варианты заданий.

Таблица 25.5

<i>N</i> <i>варианта</i>	A	B	p_1	p_2	t_1	t_2	K
1	10	20	0,4	0,6	12	16	100
2	25	30	0,2	0,8	28	26	200
3	100	200	0,3	0,7	100	190	300

4	150	180	0,65	0,35	70	200	400
---	-----	-----	------	------	----	-----	-----

Задание 8.

1. Задать дискретную GPSS-функцию, представленную в таблице 25.6. Построить график функции.

Таблица 25.6

Значение	2	3	4	8	10
вероятность	0.1	0.2	0.1	0.3	0.3

2. Задать кусочно-непрерывную GPSS-функцию, состоящую из трёх стыкующихся отрезков для значений, указанных в таблице 25.7. Аргументом функции является случайное число, равномерно распределенное на интервале от 0 до 1. Построить график функции.

Таблица 25.7

номер отрезка	1	2	3
вероятность того, что функция примет значение из этого отрезка	0.4	0.4	0.2
равновероятные значения отрезка	$2 \div 4$	$5 \div 12$	$13 \div 40$

3. Задать пуассоновский поток заявок:

- 1) с параметром 0,25 1/мин;
- 2) со средним значением времени поступления 5 мин.

Задание 9.

1. Задать дискретную GPSS-функцию, представленную в таблице 25.8. Построить график функции.

Таблица 25.8

Значение	3	2	1	4	8	5
Вероятность	0.05	0.05	0.1	0.2	0.3	0.3

2. Задать кусочно-непрерывную GPSS-функцию, состоящую из четырёх стыкующихся отрезков для значений, указанных в таблице 25.9. Аргументом функции является случайное число, равномерно распределенное на интервале от 0 до 1. Построить график функции.

Таблица 25.9

<i>номер отрезка</i>	1	2	3	4
<i>вероятность того, что функция примет значение из этого отрезка</i>	0.5	0.2	0.2	0.1
<i>равновероятные значения отрезка</i>	$3 \div 8$	$9 \div 13$	$14 \div 40$	$41 \div 50$

3. Задать пуассоновский поток заявок:

1) с параметром 0,33 1/час.;

2) со средним значением времени поступления 25 мин.

Задание 10.

1. Задать дискретную GPSS-функцию, представленную в таблице 25.10. Построить график функции.

Таблица 25.10

<i>Значение</i>	6	8	12	14	20	25	30
<i>Вероятность</i>	0.1	0.05	0.05	0.2	0.2	0.3	0.1

2. Задать кусочно-непрерывную GPSS-функцию, состоящую из четырёх стыкующихся отрезков для значений, указанных в таблице 25.11. Аргументом функции является случайное число, равномерно распределенное на интервале от 0 до 1. Построить график функции.

Таблица 25.11

<i>номер отрезка</i>	1	2	3	4
<i>вероятность того, что функция примет значение из этого отрезка</i>	0.2	0.3	0.15	0.35
<i>равновероятные значения отрезка</i>	$100 \div 150$	$151 \div 200$	$201 \div 225$	$226 \div 250$

3. Задать пуассоновский поток заявок:

1) с параметром 0,2 1/год;

2) со средним значением времени поступления 4 года.

Задание 11.

1. Задать дискретную GPSS - функцию, представленную в таблице 25.12. Построить график функции.

Таблица 25.12

Значение	2	3	6	5	8	10
Вероятность	0.1	0.2	0.2	0.3	0.1	0.1

2. Задать кусочно-непрерывную GPSS-функцию, состоящую из четырёх стыкующихся отрезков для значений, указанных в таблице 25.13. Аргументом функции является случайное число, равномерно распределенное на интервале от 0 до 1. Построить график функции.

Таблица 25.13

номер отрезка	1	2	3	4
вероятность того, что функция примет значение из этого отрезка	0.38	0.12	0.25	0.25
равновероятные значения отрезка	$1 \div 5$	$6 \div 10$	$11 \div 15$	$16 \div 20$

3. Задать пуассоновский поток заявок:
- 1) с параметром 0,5 1/сутки;
 - 1) со средним значением времени поступления 4 дня.

Задание 12. Моделирование системы обслуживания.

На вход одноканальной системы обслуживания поступает два потока требований. Первый из них - пуассоновский с интенсивностью λ 1/мин., во втором - интервалы поступления распределены равномерно на отрезке $a \pm b$ мин. Интенсивность обслуживания требования прибором зависит от длины очереди на обслуживание. Если длина очереди меньше либо равна d (больше d), то время обслуживания распределено экспоненциально со средним t_1 (t_2) мин. Промоделировать работу системы в течение K часов. В таблице 25.14 приведены варианты заданий.

Таблица 25.14

№ варианта	λ	a	B	d	t_1	t_2	K
1	0,2	6	10	3	6	7	100
2	0,25	4	6	4	3	5	250
3	0,5	3	5	2	2	4	350
4	0,75	2	5	5	3	5	400

Самостоятельно задать функциональные зависимости двумя способами: через дискретную и непрерывную GPSS-функции с числом отрезков не меньше трёх когда:

- 1) интенсивность обслуживания требования прибором зависит от времени функционирования системы;
- 2) интенсивность обслуживания требования прибором зависит от числа нулевых входов в очередь;
- 3) интенсивность обслуживания требования прибором зависит от среднего времени пребывания в очереди;
- 4) интенсивность обслуживания требования прибором зависит от коэффициента загрузки прибора.

Задание 19. Моделирование система автоматизации проектирования.

Система автоматизации проектирования состоит из ЭВМ и трех терминалов. Каждый проектировщик формирует задание на расчет в интерактивном режиме. Набор строки задания занимает 10 ± 5 с. Получение ответа на строку требует 3 с работы ЭВМ и 5 с работы терминала. После набора десяти строк задание считается сформированным и поступает на решение, при этом в течение 10 ± 3 с ЭВМ прекращает выработку ответов на вводимые строки. Вывод результата требует 8 с работы терминала. Анализ результата занимает у проектировщика 30 с, после чего цикл повторяется.

Промоделировать работу системы в течение 6 ч. Определить вероятность простоя проектировщика из-за занятости ЭВМ и коэффициент загрузки ЭВМ.

Указания. Считать транзактом проектировщика, работающего за терминалом.

Задание 20. Моделирование транспортного цеха.

Транспортный цех объединения обслуживает три филиала А, В и С. Грузовики перевозят изделия из А в В и из В в С, возвращаясь затем в А без груза. Погрузка в А занимает 20 мин, переезд из А в В длится 30 мин, разгрузка и погрузка в В - по 20 мин, переезд в С - 30 мин, разгрузка в С - 20 мин и переезд в А - 20 мин. Если к моменту погрузки в А и В отсутствуют изделия, грузовики уходят дальше по маршруту. Изделия в А выпускаются партиями по 1000 шт. через 20 ± 3 мин, в В - такими же партиями через 20 ± 5 мин. На линии работает 8 грузовиков, каждый перевозит 1000 изделий. В начальный момент все грузовики находятся в А.

Промоделировать работу транспортного цеха объединения в течение 1000 часов. Определить частоту пустых перегонов грузовиков между А и В, В и С.

Задание 21. Моделирование системы передачи речи.

В системе передачи цифровой информации передается речь в цифровом виде. Речевые пакеты передаются через два транзитных канала, буферизируясь в накопителях перед каждым каналом. Время передачи пакета по каждому из каналов является экспоненциально распределенной величиной со средним

значением 5 мс. Пакеты поступают через 6 ± 3 мс. Пакеты, передававшиеся более 10 мс, на выходе системы уничтожаются, так как это значительно снижает качество передаваемой речи. Уничтожение более 30 % пакетов недопустимо. При достижении такого уровня система за счет ресурсов ускоряет передачу в каналах до среднего значения 4 мс. При снижении уровня до приемлемого происходит отключение ресурсов.

Промоделировать 10 с работы системы. Определить частоту уничтожения пакетов и частоту подключения ресурсов.

Задание 22. Моделирование системы передачи данных.

Система передачи данных обеспечивает передачу пакетов данных из пункта **A** в пункт **C** через транзитный пункт **B**. В пункт **A** пакеты поступают через 10 ± 5 мс. Здесь они буферизуются в накопителе с максимальной емкостью 20 пакетов и передаются по одной из двух линий с равной вероятностью: **AB1** - за 20 мс; **AB2** - за 20 ± 5 мс.

В пункте **B** пакеты снова буферизируются в накопителе с максимальной емкостью 25 пакетов и далее передаются по линии **BC1** за 20 ± 3 мс и **BC2** за 25 мс. Причем пакеты из **AB1** поступают в **BC1**, а из **AB2** - в **BC2**. При заполнении буферов поступившие в это время пакеты получают отказ.

Промоделировать работу системы в течении 1 мин. Оценить вероятности отказов пакетам в результате переполнения буферов.

Задание 23. Моделирование узла коммутации сообщений.

В узел коммутации сообщений, состоящий из входного буфера, процессора, двух исходящих буферов и двух выходных линий, поступают сообщения с двух направлений. Сообщения с первого направления поступают во входной буфер, обрабатываются в процессоре, накапливаются в выходном буфере первой линии и передаются по первой выходной линии. Сообщения со второго направления обрабатываются аналогично, но передаются по второй линии. Применяемый метод контроля потоков позволяет одновременное присутствие в системе не более трех сообщений с каждого направления. Сообщения поступают через интервалы времени 15 ± 7 мс. Время обработки в процессоре равно 7 мс на транзакт, время передачи по выходной линии равно 15 ± 5 мс.

Если транзакт поступает при наличии в системе трех сообщений с данного направления, то оно получает отказ.

Промоделировать работу узла коммутации в течение 10 с. Определить загрузку устройств и вероятность отказов в обслуживании из-за переполнения буферов направлений.

Задание 24. Моделирование процесса сборки. На сборочный участок цеха предприятия через интервалы времени, распределенные экспоненциально со средним значением 10 мин, поступают от трех независимых источников детали. Половина всех поступающих деталей перед сборкой должна пройти

предварительную обработку в течение 7 мин. На сборку подаются обработанная и необработанная детали. Процесс сборки занимает 6 мин. Затем изделие поступает на регулировку, продолжающуюся в среднем 8 мин и имеющую экспоненциальное распределение.

Промоделировать процесс сборки в течение 24 час. Оценить загрузку операций.

Задание 25. Моделирование работы цеха,

Детали, необходимые для работы цеха, находятся на цеховом и центральном складах. На цеховом складе может храниться до 20 комплектов деталей, потребность в которых составляет один комплект и возникает через 60 ± 10 мин. В случае снижения запасов до трех комплектов, формируется заявка на пополнение запасов цехового склада до полного объема в 20 комплектов в течение 60 мин, которая посылается на центральный склад, где происходит комплектование в течение 60 ± 20 мин и доставка деталей в цех за 60 ± 5 мин.

Промоделировать работу цеха в течение 400 час. Определить вероятность простоя цеха из-за отсутствия деталей.

Задание 26. Моделирование работы ЭВМ.

ЭВМ предоставляет каждому терминалу циклически 30 с времени для обработки задач. Если за это время задача обрабатывается, то обслуживание завершается; если нет, то задача становится в специальную очередь для завершения выполнения. В этом случае используются свободные циклы терминалов для дообслуживания задачи, т.е. она обслуживается, если на терминалах нет заявок. Заявки от терминалов поступают через 30 ± 5 с и имеют длину 300 ± 50 знаков. Скорость обработки заданий ЭВМ равна 10 знаков/с.

Промоделировать 5 ч работы ЭВМ. Определить загрузку ЭВМ, параметры очереди неоконченных заданий.

Задание 27. Моделирование системы передачи данных.

В системе передачи цифровой информации передается речь в цифровом виде. Речевые пакеты передаются через два транзитных канала, буферизируются в накопителе перед каждым каналом. Время передачи пакета по каждому из каналов составляет 5 мс и является экспоненциально распределенной величиной. Пакеты поступают через 6 ± 3 мс. Пакеты, передающиеся более 10 мс, на выходе системы уничтожаются, так как это значительно снижает качество передаваемой речи. Уничтожение более 30% пакетов недопустимо. При достижении такого уровня система за счет ресурсов ускоряет передачу в каналах до среднего значения 4 мс. При снижении уровня приемлемого происходит отключения ресурса.

Промоделировать 10 с работы системы. Определить частоту уничтожения пакетов и частоту подключения ресурсов.

Задание 28. Моделирование производственного процесса.

Рассмотрим производственный технологический процесс, реализуемый линией из трех агрегатов : А, Б и В. Выходной поток продукции, производимой первым агрегатом (А), является пуассоновским со средней нормой выработки, равной 10 изделиям в час. Второй агрегат (Б) функционирует с равномерным распределением длительностей обработки в интервале 4 ± 6 мин. каждого изделия, поступающего от агрегата А. Закон обслуживания изделий агрегатом В представлен в таблице 25.15.

Таблица 25.15

Вероятность	0,1	0,2	0,3	0,2	0,2
Длительность обслуживания (мин.)	2	3	4	5	6

При скоплении на входе агрегата В двух или более изделий, в технологической линии возникает затор.

Промоделировать функционирование линии в течении 100 час. Определить долю продолжительности всего технологического процесса, в течение которого на входе агрегата В имеет место затор. Построить гистограмму распределения продолжительности заторов.

Задание 29. Моделирование работы заправки

На заправке имеется три вида топлива для автомобилей: низко – и высокооктановый бензин и дизельное топливо. Для каждого вида топлива имеются свои раздаточные колонки. Характеристики заправки приведены в таблице 25.16.

Таблица 25.16

Вид топлива	Количество колонок	% заправляемых автомобилей	Количество Заправляемого топлива, Л	Скорость отпуска 10 литров топлива, мин.	Стоимость топлива за литр, грн.
Низкооктановый бензин	1	30	Равновероятно от 5 до 60 л через 5 л	1,2	0,55

Высокооктановый бензин	2	50	Равновероятно от 5 до 40 л через 5 л	1,5	0,65
Дизтопливо	1	20	Равновероятно от 10 до 60 л через 5 л	1,8	0,5

Прибытие автомобилей на заправку распределено по закону Эрланга второго порядка со средним значением 2,2 мин.

После заправки 10 % автомобилей доливают от 0,5 до 2 л масла. Доливка 0,5 л масла занимает 2 мин. Стоимость одного литра масла составляет 2 грн.

Определить среднее время обслуживания автомобилей и выручку за пять дней работы заправки.

Задание 30. Моделирование работы станции технического обслуживания (СТО)

На СТО прибывают автомобили по закону Эрланга второго порядка со средним временем прибытия 14 мин. для проведения технического обслуживания (ТО) (36 % автомобилей) и ремонта (64 % автомобилей). На СТО имеется два бокса для проведения ТО и три бокса для ремонта. Выполнение простого, средней сложности и сложного ремонтов равновероятно.

Время для выполнения работ по ТО и ремонту зависит от категории выполняемых работ и приведено в таблице 25.17.

После ТО 12 % автомобилей поступают для выполнения ремонта средней сложности.

Определить среднее время обслуживания автомобилей и выручку за пять дней работы СТО.

Таблица 25.17

Категория ремонта	Время ремонта	Стоимость ремонта, грн.
ТО	Равномерно распределено в интервале 10 – 55 мин.	Равномерно распределена в интервале 10 - 40
Простой	Равномерно распределено в интервале 12 – 45 мин.	Равномерно распределена в интервале 5 - 45

Средней сложности	Нормально распределено со средним 45 мин. и среднеквадратичным отклонением 5 мин.	Равномерно распределена в интервале 10 - 140
Сложный	Равномерно распределено в интервале 80 – 150 мин.	Равномерно распределена в интервале 35 - 255

Задание 31. Моделирование работы станции скорой помощи.

На станцию скорой помощи поступают вызовы по телефону. У станции имеется 5 каналов для одновременного приема вызовов. Абоненты затрачивают 35 с для набора номера и если застают все каналы станции занятыми, то через 20 с повторяют вызов до тех пор пока не дозвонятся. Среднее время между попытками вызова скорой помощи распределено по закону Эрланга второго порядка. Время записи вызова составляет 1 мин.

На станции скорой помощи имеется 15 машин для обслуживания вызовов. Время, затраченное для проезда к больному, зависит от расстояния и вероятности поступления вызова с этого расстояния (таблица 25.18). Скорость движения автомобилей равномерно распределена в интервале 35 – 55 км/час.

Таблица 25.18

Вероятность	0,15	0,22	0,17	0,28	0.18
Расстояние, км	5	8	12	15	20

Время оказания помощи больному распределено по нормальному закону со средним значением 15 мин. и среднеквадратичным отклонением 3 мин.

Определить среднее время от момента первоначального вызова скорой помощи до окончания помощи больному и средний пробег автомобиля за пять дней круглосуточной работы.

Задание 32. Моделирование работы госпиталя

В госпиталь в течении суток поступают раненые и пострадавшие от катастрофы, которых доставляют на пяти - (30 %) и трехместных (70 %) автомобилях. Время приезда автомобилей распределено по закону Эрланга третьего порядка со средним значением 25 мин.

В госпитале две бригады из трех терапевтов и одного хирурга проводит осмотр раненых и потерпевших в течении времени 4 ± 2 мин., определяют требуемый вид оказания медицинской помощи и направляют пациентов в соответствующую палату (таблица 25.19).

Таблица 25.19

Вероятность направления	0,15	0,25	0,35	0,15	0,1
Палата	Интенсивной терапии	Операционная	Реанимационная	Хирургическая	Терапии
Количество мест	20	6	20	25	30
Время оказания помощи	Равномерно распределено в интервале 1440 – 2060 мин.	Равномерно распределено в интервале 20 – 120 мин.	Равномерно распределено в интервале 2880 – 3660 мин.	Нормально распределено со средним 1800 мин. и среднеквадратичным отклонением 60 мин.	Равномерно распределено в интервале 1200 – 2200 мин.

После операционной 55 % больных направляются в палату реанимации, а 45 % в палату интенсивной терапии.

Промоделировать работу госпиталя в течении 10 суток и определить среднее время пребывания пострадавших в нем.

Список литературы

1. Beasley J. E. and Whitchurch G. O. R. education — a survey of young O. R. workers. // *Journal of the Operational Research Society*. - 1984. - № 35. - P. 281 — 288.
1. Ray J. Paul. Recent Developments in Simulation Modelling. // *Journal of the Operational Research Society*. - 1991. - № 35. - P. 217 — 226.
3. Crookes J. G. Generators, generic models and methodology. // *Journal of the Operational Research Society*. - 1987. - № 38. - P. 765 — 768.
4. Crookes J. G., Balmer D. W., Chew S. T. and Paul R. J. A three-phase simulation system written in Pascal. // *Journal of the Operational Research Society*. - 1986. - № 37. - P. 603 — 618.
1. Balmer D. W. and Paul R. J. CASM— the right environment for simulation. // *Journal of the Operational Research Society*. - № 37. - 1986. - P. 443 — 452.
6. Tjo Jay. Simulation modelling tools. // *Computer Week-Moscow*. - № 35. - 1995. - PP. 35 — 36, 55.
7. Томашевський В.М. Імітаційне моделювання систем і процесів. – К: ІСДО, 1994. — 124 с.

8. O'Keefe Robert and Roach Joan W. Artificial Intelligence Approach to Simulation. // *Journal of the Operational Research Society*. – 1987. - № 38. - P. 713 – 722.
1. O'Keefe Robert. Simulation and expert systems: a taxonomy and some examples. // *J. Simulation*. – 1986. - № 46. P. 10 — 16.
2. Doukidis G. and Paul R. J. Research into expert systems to aid simulation modelling formulation. // *Journal of the Operational Research Society*. - 1985. – 36. – P. 319 — 325.
11. Crookes J. G. Simulation in 1981. // *J. Eur. Journal of the Operational Research Society* 1982 9, 1 — 7.
12. Шрайбер Т. Дж. Моделирование на GPSS. – М: Машиностроение, 1980. – 592 с.

СОДЕРЖАНИЕ

вВедение	4
1. ОБЪЕКТЫ GPSS	5
1.1. Блоки и транзакты	5
1.2. Объекты типа "ресурсы"	7
1.3. Переменные	8
1.4. Функции	8
1.5. Ячейки и матрицы ячеек сохраняемых величин	8
1.6. Очереди	8
1.7. Таблицы	9
2. Таймер модельного времени	9
Особенности таймера GPSS	9
3. Типы операторов GPSS	..10
Общие сведения о формате блоков GPSS	10
4. Внесение транзактов в модель. Блок (ГЕНЕРИРОВАТЬ)	11

GENERATE

5. Удаление транзактов из модели. Блок TERMINATE (ЗАВЕРШИТЬ)	14
Управление продолжительностью процесса моделирования	14
6. Элементы, отображающие одноканальные обслуживающие устройства	.15
7. задержки во времени. Блок ADVANCE (ЗАДЕРЖАТЬ)	
Примеры использования блока ADVANCE.....	19
8. Сбор статистики при ожидании. Блоки QUEUE (СТАТЬ В ОЧЕРЕДЬ), DEPART (ПОКИНУТЬ ОЧЕРЕДЬ)	
9. Переход транзакта в блок, отличный от последующего. Блок TRANSFER (ПЕРЕДАТЬ)	
9.1. Блок TRANSFER в режиме безусловной передачи	
9.2. Статистический режим	.22
9.3. Режим BOTH	.23
10. Моделирование многоканальных устройств	24
10.1. Блоки ENTER (ВОЙТИ) и LEAVE (ВЫЙТИ)	25
10.2. Определение ёмкости МКУ	26
11. Примеры построения GPSS-моделей	27
12. Переменные	37
12.1. Общая характеристика переменных	37
12.2. Арифметические переменные	38
12.3. Арифметические переменные с плавающей точкой	40
13. Определение функции в GPSS	40

13.1. Особенности вычисления дискретных и непрерывных GPSS-функций.....	44
13.2. Моделирование непрерывных случайных величин. Использование функций в блоках GENERATE и ADVANCE	45
13.3. Непрерывные случайные переменные, рассматриваемые как дискретные	47
13.4. Функции распределения случайных величин	47
13.4.1. Моделирование пуассоновского потока	48
13.4.2. Моделирование эрланговского потока	49
13.4.3. Моделирование гипер - и гипоекспоненциальных распределений	49
13.4.4. Моделирование нормального закона распределения	51
13.4.5. Моделирование других законов распределений	52
14. Стандартные числовые атрибуты и параметры транзактов	53
14.1. Стандартные числовые атрибуты	54
14.2. СЧА блоков и системные числовые атрибуты	55
14.3. Параметры транзактов	56
14.4. Изменение значений параметров. Блок ASSIGN (НАЗНАЧИТЬ)	58
14.5. Отметка времени	59
14.6. Транзитное время. Блок MARK (ОТМЕТИТЬ)	59
14.7. Изменение приоритета транзактов. Блок PRIORITY (НАЗНАЧИТЬ ПРИОРИТЕТ).....	60
14.8. Организация циклов. Блок LOOP (ЦИКЛ)	60

15. Примеры фрагментов GPSS - моделей использованием СЧА и параметров транзактов	с 61
16. Сохраняемые величины	64
16.1. Оператор INITIAL (ИНИЦИАЛИЗИРОВАТЬ)	64
16.2. Блок SAVEVALUE (СОХРАНИТЬ ВЕЛИЧИНУ)	65
17. Косвенная адресация	66
18. Проверка числовых выражений. Блок) TEST (ПРОВЕРИТЬ	67
19. Определение и использование таблиц	69
19.1. Оператор TABLE (ТАБЛИЦА)	69
19.2. Блок TABULATE (ТАБУЛИРОВАТЬ)	70
20. СПИСКИ ПОЛЬЗОВАТЕЛЕЙ	72
20.1. Блок LINK (ВНЕСТИ В СПИСОК)	72
20.2. Блок UNLINK (ВЫВЕСТИ ИЗ СПИСКА)	74
21. Статистическая обработка результатов моделирования	82
22. КОМАНДЫ GPSS/PS	85
22.1. Команда @<FILESPEC>.....	86
22.2. Команда ANOVA.....	86
22.3. Команда CONTINUE.....	89
22.4. Команда DELETE.....	90
22.6. Команда DOS.....	91
22.7. Команда EDIT.....	91

22.8. Команда EVENTS.....	92
22.9. Команда GROUPS.....	93
22.10. Команда MICROWINDOW.....	93
22.11. Команда PLOT.....	93
22.12. Команда RENUMBER.....	95
22.13. Команда REPORT.....	95
22.14. Команда RESULT.....	96
22.15. Команда SAVE.....	97
22.16. Команда SHOW.....	98
22.17. Команда STEP.....	98
22.18. Команда STOP.....	99
22.19. Команда USERCHAINS.....	100
22.20. Команда WINDOW.....	101
 23. Диалоговые возможности	 82
 24. Организация связи системы GPSS/PC с другими программами	 103
 25. Задания для самостоятельной работы	 104
 Список литературы.....	 118