

**КОМПЬЮТЕРНЫЙ
ЭКСПЕРИМЕНТ
В ИМИТАЦИОННОМ
МОДЕЛИРОВАНИИ**

Учебно-методическое пособие

УДК 519.8(075)
ББК 22.18я73
К63

Автор-составитель: доцент кафедры информатики и информационных технологий УО «ВГУ им. П.М. Машерова», кандидат биологических наук, доцент **А.А. Чиркина**

Рецензенты:

заведующий кафедрой прикладной математики УО «ВГУ им. П.М. Машерова»,
кандидат физико-математических наук, доцент *Л.В. Маркова*;
доцент кафедры технологии и оборудования машиностроительного производства УО «ВГТУ»
Ю.Е. Махаринский

Учебно-методическое пособие включает теоретические сведения по функциональным возможностям системы имитационного моделирования GPSS. Рассмотрены вопросы планирования и обработки результатов эксперимента с помощью пакета STATISTICA. Издание содержит примеры решения задач с помощью указанных пакетов и контрольные вопросы для самопроверки. Предназначено для студентов дневного отделения, обучающихся по специальности 1-31 03 03 «Прикладная математика».

УДК 519.8(075)
ББК 22.18я73

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Часть 1. Система моделирования GPSS.....	5
1.1. Краткая характеристика системы GPSS.....	5
1.2. Моделирование одноканальных устройств	8
1.3. Моделирование многоканальных устройств	16
1.4. Моделирование систем с использованием блоков передачи управления, работа с логическими ключами.....	18
1.5. Моделирование систем с использованием копий транзактов и синхронизации их движения	28
1.6. Моделирование систем с использованием блоков захвата и освобождения приборов.....	32
1.7. Моделирование недоступности устройств и накопителей.....	33
1.8. Организация циклов	37
1.9. Обработка результатов моделирования средствами GPSS	38
1.10. Моделирование последовательности значений случайных величин с заданным законом распределения.....	42
1.11. Контрольные вопросы по теме.....	48
Часть 2. Планирование и обработка эксперимента с помощью пакета STATISTICA.....	50
2.1. Вычисление основных статистических характеристик	50
2.2. Построение таблицы, графиков частот и гистограммы.....	51
2.3. Планирование и обработка результатов эксперимента	54
2.4. Автоматическое создание отчета	59
2.5. Контрольные вопросы по теме.....	60
Список рекомендуемой литературы	61
Приложение 1 Блоки GPSS	62
Приложение 2 Стандартные числовые атрибуты	64

ВВЕДЕНИЕ

Имитационное моделирование – это вид компьютерного моделирования, для которого характерно воспроизведение на ЭВМ (имитация) процесса функционирования исследуемой системы. Для создания имитационных моделей можно использовать как универсальные языки программирования, так и специализированные программные средства – языки имитационного моделирования (GPSS, SIMSCRIPT, SIMULA и прочие). Универсальные языки имеют следующие достоинства: доступность, возможность решения широкого класса задач. Специализированные языки моделирования по некоторым показателям имеют преимущество перед универсальными языками: наличие специальных средств описания структуры системы и ее поведения, статистической обработки результатов моделирования; средств отладки модели; генерации и обеспечения независимости случайных величин, используемых в модели.

За всю историю развития вычислительной техники было создано более 300 языков моделирования. Одним из первых языков, появившихся в 70-х годах прошлого века, был язык GPSS (General Purpose Simulation System – система моделирования общего назначения). Идеи, положенные в основу этой разработки, оказались настолько плодотворны, что использовались при создании многих программных средств аналогичного назначения в нашей стране и за рубежом.

Язык GPSS предназначен для написания имитационных моделей систем с дискретными событиями. Наиболее удобно в системе GPSS описываются модели систем массового обслуживания, для которых характерны относительно простые правила функционирования составляющих их элементов. GPSS поддерживает блочно-ориентированный подход, в рамках которого каждый моделирующий блок имеет свое функциональное назначение и представлен соответствующими функциональными объектами, имеющими аналоги с элементами систем массового обслуживания. В первом разделе учебно-методического пособия рассматриваются вопросы описания и формализации моделируемых систем, технология построения имитационных моделей, организация экспериментальных исследований с использованием языка GPSS.

Второй раздел учебного издания предназначен для формирования навыков использования компьютерных технологий в управлении экспериментом и обработке результатов исследований. Пакет STATISTICA фирмы StatSoft Inc. (США) представляет собой интегрированную систему статистического анализа и обработки данных. Он позволяет проводить всесторонний анализ данных, представлять результаты анализа в виде таблиц и графиков, автоматически создавать отчеты о проделанной работе. Рассмотрены возможности пакета, связанные с получением основных описательных статистик, построением плана эксперимента, регрессионным и дисперсионным анализом данных.

ЧАСТЬ 1. СИСТЕМА МОДЕЛИРОВАНИЯ GPSS

1.1. Краткая характеристика системы GPSS

Система имитационного моделирования GPSS (General Purpose Simulation System) предназначена для описания и исследования дискретных моделей систем массового обслуживания (СМО). Описание на языке GPSS представляет собой совокупность операторов (блоков), характеризующих процессы обработки заявок. В языке имеется более 40 блоков с различным функциональным назначением. Для детализации функций, выполняемых блоком, у каждого из них имеется набор параметров.

Организация модельного времени. Существуют два способа формирования конечного множества моментов модельного времени.

Первый способ – изменение модельного времени с постоянным шагом. В этом случае весь интервал времени моделирования разбивается на промежутки фиксированной длины Δt . Затем на каждом промежутке последовательно обрабатываются алгоритмы, имитирующие компоненты моделируемой системы, после чего происходит изменение модельного времени. Точность моделирования достигается ценой больших затрат машинного времени, находящихся в обратно пропорциональной зависимости от величины Δt . Обычно принцип фиксированных интервалов используется при построении моделей непрерывных динамических систем, описываемых системами дифференциальных уравнений различных типов.

Второй способ – моделирование по особым состояниям. Системное время меняется в моменты особых состояний, которые соответствуют наступлению событий в реальной системе (поступление заявки, освобождение канала и т.д.). Для реализации моделирования по особым состояниям требуется разработка специальной процедуры планирования событий.

Моделирование системы на GPSS представляет собой имитацию последовательности переходов системы из одного состояния в другое в особые моменты времени (событийное моделирование). Таймер модельного времени имеет следующие особенности:

- время отображается в условных значениях (целые числа), поэтому перед составлением модели необходимо провести временное масштабирование для всех временных параметров и характеристик модели системы; единица модельного времени выбирается разработчиком;
- система GPSS не анализирует состояние модели в каждый следующий момент модельного времени, а продвигает таймер к моменту времени, когда происходит ближайшее следующее событие.

Объекты GPSS подразделяются на 7 категорий.

Динамическими объектами в СМО являются *транзакты* (сообщения, заявки), которые представляют собой единицы исследуемых потоков и производят ряд определенных действий. Отличие одних транзактов от других состоит в назначаемом им наборе параметров. Для имитации процессов, протекающих в моделируемой системе, все транзакты, порождаемые в процессе моделирования, образуют списки, в которых транзакты отсортированы по времени, при равных временах у транзактов – по приоритетам. В процессе моделирования может быть образовано до пяти списков:

– *список текущих событий* содержит транзакты, которые должны перемещаться в текущий момент модельного времени; все они имеют одинаковое время, равное текущему и, если их приоритеты совпадают, то очередной транзакт для перемещения выбирается по правилу FIFO;

– *список будущих событий* содержит транзакты, которые будут перемещаться по модели в будущие моменты времени; в этот список попадают транзакты, вошедшие в блоки имитации задержки или заблокированные по каким-либо условиям;

– *списки прерываний, синхронизации и пользователя* содержат транзакты, обслуживание которых прервано по некоторым условиям, задаваемым блоками прерывания, синхронизации или находящимся под управлением пользователя.

В начальный момент времени в GPSS-модели нет ни одного транзакта. В процессе моделирования транзакты входят в модель (генерируются) в соответствии с потребностями, которые возникают в моделируемой системе. В общем случае в модели существует большое число транзактов, но в один момент времени двигается только один. Если транзакт начал свое движение, он перемещается от блока к блоку по пути, заданному алгоритмом моделирования. Продвижение текущего транзакта продолжается по блокам модели до тех пор, пока не произойдет одно из следующих событий:

- транзакт входит в блок задержки и переходит в список будущих событий;
- транзакт входит в один из блоков проверки условий, и условие не позволяет транзакту перемещаться дальше (наступает условие блокировки), тогда транзакт переводится в список будущих событий;
- транзакт входит в блок удаления транзакта из модели.

Если возникло одно из описанных условий, то транзакт остается на месте, и начинается перемещение в модели другого транзакта, который выбирается из списка текущих событий. Если становится невозможным продвижение всех транзактов из списка текущих событий, то изменяется модельное время, то есть наступает время следующего события или группы событий, и алгоритм повторяется. Изменение по-

следовательного перемещения транзакта по модели может быть нарушено оператором, определяющим для данного транзакта номер следующего блока.

Объекты аппаратной категории – это устройства, памяти и логические ключи. Воздействуя на эти объекты, транзакты могут изменять их состояние и влиять на движение других транзактов. *Устройства* моделируют реальные объекты, в которых может происходить обработка транзактов (сообщений). Устройство является аналогом обслуживающего прибора системы массового обслуживания. В любой момент времени устройство может быть занято только одним транзактом. *Памяти* описывают такие устройства, которые обслуживают одновременно несколько транзактов. Количество одновременно обслуживаемых транзактов определяют емкость памяти. *Ключи* используются для блокировки или изменения движения транзактов в зависимости от ранее наступивших событий. Наличие в модели объектов аппаратной категории позволяет автоматически регистрировать статистическую информацию.

Операционные объекты, то есть *блоки*, задают логику функционирования модели системы и определяют пути движения транзактов между объектами аппаратной категории. Описание системы на GPSS представляет собой последовательность блоков, каждый из которых соответствует некоторому оператору. Каждый блок имеет определенное количество реквизитов, называемых полями, которые отделяются друг от друга запятой (это аналоги параметров процедур и функций в языках программирования), но положение полей строго фиксировано и отсутствие некоторого поля отмечается запятой. Формат записи блоков имеет следующий вид:

[метка] ОПЕРАТОР А,В,С,Д,...; Комментарии

Метка блока должна начинаться с буквы и содержать не более пяти алфавитно-цифровых символов. Для некоторых блоков метка не ставится. ОПЕРАТОР – это ключевое слово, указывающее конкретную функцию, выполняемую данным блоком. Далее после имени блока через пробел располагаются операнды, уточняющие и конкретизирующие выполнение функции, определенной в поле операции. Операнды разделяются запятыми. Количество операндов определяется типом блока и его функциональным назначением в данной программе. Некоторые операции вообще не имеют операндов, а в некоторых операнды могут быть пропущены. Если некоторое частичное поле остается пустым, т.е. не нужен соответствующий аргумент, это отмечается еще одной запятой. Комментарий можно поместить в рабочей строке после точки с запятой или в строке, в первой позиции которой будет находиться звездочка (*).

Объекты вычислительной категории описывают связи между элементами системы, задаваемые с помощью аналитических или логических соотношений. Они могут служить для задания вероятностных законов распределения случайных величин в имитационной модели, для численного или логического описания условий, определяющих движение транзактов. В качестве объектов вычислительной категории введены арифметические и булевские переменные и функции.

К **статистическим объектам** относятся *очереди* и *таблицы*, вводимые для оценки характеристик поведения системы. Транзакт помещается в очередь в том случае, когда некоторое устройство не в состоянии обслужить его немедленно (например, устройство занято, либо память переполнена). Статистические таблицы используются для получения характеристик и частотных распределений определенных аргументов, например, времени задержки транзакта в модели в целом или в отдельных ее частях, длин очередей, содержимого памяти и так далее. В конце эксперимента с моделью результаты в таблицах выводятся на печать.

Запоминающие объекты служат для задания условий моделирования, хранения, накопления и обработки информации, получение которой не предусмотрено стандартными средствами GPSS. К объектам запоминающей категории относятся ячейки и матрицы.

Объекты группирующей категории содержат информацию о транзактах, находящихся в модели. К ним относятся списки и группы.

Стандартные числовые атрибуты. Все элементы, используемые в GPSS, имеют специальные числовые характеристики, которые называются *стандартными числовыми атрибутами* (СЧА). Например, для транзактов можно применять такие СЧА, как приоритет или время пребывания в модели. В GPSS СЧА имеют следующие объекты: транзакты, блоки, устройства, многоканальные устройства, очереди, таблицы, ячейки и матрицы ячеек сохраняемых величин, списки и группы, вычислительные объекты (СЧА вычисленного значения функции, СЧА вычисленного значения переменной). С помощью СЧА пользователь получает доступ к характеристикам состояния системы в процессе моделирования.

Следует отметить, что символьные обозначения СЧА нельзя использовать в качестве меток операторов и блоков программы GPSS.

1.2. Моделирование одноканальных устройств

Пример 1. Заявки поступают в СМО через фиксированное время (8 минут). Время обработки (обслуживания) каждой заявки распределено равномерно в интервале времени от 5 до 15 минут. После обработки заявки покидают систему. Провести обработку 100 заявок.

Для создания нового файла используется команда **File>New>Model.**

Текст программы на GPSS:

```

GENERATE 8      ; генерация транзактов каждые 8 минут
SEIZE 1        ; занятие прибора обслуживания 1
ADVANCE 10,5   ; задержка транзакта на время от 5 до 15 минут
RELEASE 1      ; освобождение прибора 1
TERMINATE 1    ; уничтожение транзакта
START 100     ; запуск процесса моделирования для 100 транзак-
                тов

```

Рассмотрим назначение основных блоков модели.

Для создания транзактов (заявок), входящих в модель, служит блок **GENERATE** (генерировать), имеющий следующий формат:

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

В поле <A> задается среднее значение интервала времени между моментами поступления в модель двух последовательных транзактов. Если этот интервал постоянен, то поле не используется, иначе в поле указывается разброс интервала времени относительно среднего значения. Разброс может быть задан в виде модификатора-интервала или модификатора-функции. Модификатор-интервал используется, когда интервал поступления транзактов является случайной величиной с равномерным законом распределения (диапазон изменения интервала поступления транзактов имеет границы A-B, A+B). Модификатор-функция используется, если закон распределения интервала поступления отличен от равномерного. В поле <C> задается время появления первого транзакта. Если это поле пусто или равно 0, то момент появления первого транзакта определяется операндами <A> и . Поле <D> задает количество транзактов, которое должно быть сгенерировано, после чего генерация транзактов прекращается. Если это поле пусто, то блок генерирует неограниченное число транзактов до завершения моделирования. В поле <E> указывается значение приоритета генерируемых транзактов, которое может лежать в диапазоне от 0 до 127, причем большее значение соответствует более высокому приоритету. Если поле <E> пусто, то генерируемые транзакты имеют нулевой приоритет. СЧА с названием PR позволяет ссылаться на приоритет транзакта.

Вход транзактов в блок **GENERATE** не допускается. В этом случае происходит ошибка исполнения и прерывание работы программы.

Блок занятия устройства **SEIZE** имеет следующий формат:

SEIZE A

В поле <A> задается номер (имя) занимаемого устройства. Устройство остается занятым до тех пор, пока занимающее его сообщение не войдет в соответствующий парный блок **RELEASE**.

Блок освобождения устройства **RELEASE** имеет следующий формат:

RELEASE A

В поле <A> задается номер (имя) освобождаемого устройства.

Транзакты обслуживаются устройствами в течение некоторого промежутка времени. Для моделирования такого обслуживания, то есть для задержки транзактов на определенный отрезок модельного времени, служит блок задержки транзактов **ADVANCE**, имеющий следующий формат:

ADVANCE A,[B]

Операнды в полях <A> и задают среднее значение и разброс интервала времени, на которое транзакт задерживается в устройстве. Транзакты, входящие в блок **ADVANCE**, переводятся из списка текущих событий в список будущих событий, а по истечении вычисленного времени задержки возвращаются назад в список текущих событий и их продвижение в модели продолжается. Если вычисленное время задержки равно 0, то транзакт в тот же момент модельного времени переходит в следующий блок, оставаясь в списке текущих событий.

В примере 1 транзакты, поступающие в модель из блока **GENERATE** каждые 8 минут, попадают в блок **SEIZE** и занимают устройство с номером 1. Далее в блоке **ADVANCE** определяется случайное время задержки транзакта, имеющее равномерное распределение на отрезке [5;15], и транзакт переводится в список будущих событий. По истечении времени задержки транзакт возвращается в список текущих событий, входит в блок **RELEASE** и освобождает устройство 1.

Для удаления транзактов из модели и подсчета удаленных транзактов служит блок **TERMINATE** (завершить), имеющий следующий формат:

TERMINATE [A]

Значение поля <A> указывает, на сколько единиц уменьшается содержимое счетчика завершений при входе транзакта в данный блок **TERMINATE**. Каждый транзакт, проходящий через блок **TERMINATE**, вычитает из счетчика A единиц. Если поле <A> не определено, то оно считается равным 0, и транзакты, проходящие через такой блок, не уменьшают содержимого счетчика завершений.

Начальное значение счетчика завершений устанавливается в поле <A> управляющего оператора **START**, предназначенного для запуска прогона модели. Прогон модели заканчивается, когда содержимое счетчика завершений обращается в 0. Счетчик завершения задает число транзактов, которые выйдут из модели, в отличие от значения поля <D> блока **GENERATE**, определяющего число транзактов, которые войдут в модель. По окончании моделирования транзакты могут оставаться в модели.

Участок модели, связанный с парой блоков GENERATE-TERMINATE, называется сегментом. Простые модели могут состоять из одного сегмента, в сложных моделях может быть несколько сегментов.

Для запуска программы на выполнение необходимо выбрать пункт меню *Command>Create Simulation*. В процессе выполнения программы собирается стандартная статистическая информация, которая автоматически выводится на печать по окончании моделирования. Рассмотрим результаты моделирования по стандартному отчету примера 1 (основные обозначения приведены в таблице 1.1):

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES				
	0.000	1023.219	5	1	0				
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY				
	1	GENERATE	127	26	0				
	2	SEIZE	101	1	0				
	3	ADVANCE	100	0	0				
	4	RELEASE	100	0	0				
	5	TERMINATE	100	0	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	101	0.992	10.052	1	101	0	0	0	26
CEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
101	0	808.000	101	2	3				
FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
128	0	1024.000	128	0	1				

Таблица 1.1

Обозначение	Описание
Общая статистика	
START TIME	время начала моделирования
END TIME	время окончания моделирования
BLOCKS	количество блоков, используемых в программе
FACILITIES	количество устройств, используемых в программе
STORAGES	количество многоканальных устройств
LOC	номер блока, назначенный системой
BLOCK TYPE	название блока
ENTRY COUNT	общее количество транзактов, прошедших через блок за время моделирования
CURRENT COUNT	количество транзактов, задержанных в блоке на момент конца моделирования
RETRY	количество транзактов, ожидающих специальных условий для прохождения через данный блок
Отчет о работе устройств	
FACILITY	название устройства

Обозначение	Описание
ENTRIES	количество транзактов, прошедших через устройство
UTIL	вероятность загрузки устройства (часть периода моделирования, когда устройство было занято)
AVE. TIME	среднее время обработки одного транзакта устройством
AVAIL	состояние готовности устройства на момент конца моделирования (1 – готово к обслуживанию очередной заявки; 0 – не готово)
OWNER	номер последнего транзакта, занимающего устройство (если не занималось, то значение 0)
PEND	количество транзактов, находящихся в режиме прерывания, ожидающих устройство
INTER	количество транзактов, прерывающих устройство в данный момент
RETRY	количество транзактов, ожидающих специальных условий, зависящих от состояния устройства
DELAY	количество транзактов, ожидающих занятия или освобождения устройства
<i>Статистические данные об очередях</i>	
QUEUE	имя очереди
MAX	максимальная длина очереди
CONT.	текущее значение длины очереди
ENTRY	общее количество входов в очередь
ENTRY(0)	общее количество входов в очередь без последующего ожидания («нулевые» входы)
AVE.CONT.	средняя длина очереди
AVE.TIME	среднее время пребывания транзактов в очереди
AVE.(0)	среднее время пребывания в очереди с учетом всех входов и без учета «нулевых» входов
RETRY	количество транзактов, ожидающих специальных условий
<i>Информация о списке текущих событий CEC (Current Events Chain)</i>	
XN	номер транзакта
PRI	приоритет транзакта (по умолчанию – 0)
M1	время пребывания транзакта в системе с момента начала моделирования
ASSEM	номер семейства транзактов
CURRENT	номер блока, в котором находится транзакт
NEXT	номер блока, в который перейдет транзакт далее

Обозначение	Описание
PARAMETER	номер или имя параметра транзакта
VALUE	значение параметра
<i>Информация о списке будущих событий FEC (Future Events Chain)</i>	
XN	номер транзакта
PRI	приоритет транзакта
BDT	таблица модельных событий – абсолютное модельное время выхода транзакта из списка будущих событий (и перехода транзакта в список текущих событий)
ASSEM	номер семейства транзактов
CURRENT	номер блока, в котором находится транзакт (0 – если транзакт не вошел в модель)
NEXT	номер блока, в который перейдет транзакт далее
PARAMETER	номер или имя параметра транзакта
VALUE	значение параметра

В рассмотренном примере количество сгенерированных транзактов равно 127. Так как время генерирования транзактов меньше, чем среднее время их обработки, обработано и выведено из системы только 100 транзактов. На момент окончания моделирования в устройство поступил 101-й по счету транзакт. Это число показано в поле OWNER статистики устройства. 26 транзактов ждут освобождения устройства. Среднее время обработки одного транзакта устройством 10.052, вероятность загрузки устройства близка к единице.

Статистические данные об очередях могут быть получены с помощью блоков занятия и освобождения очереди QUEUE (вход в очередь) и DEPART (выход из очереди). Формат записи блока QUEUE:

QUEUE A,[B]

В поле <A> задается номер или имя очереди. Поле определяет количество мест в очереди, занимаемое транзактом (по умолчанию 1). Поскольку очередь обычно используется для измерения времени ожидания, за блоком QUEUE обычно следует такой блок, как SEIZE, который может задержать сообщение. Формат блока DEPART:

DEPART A,[B]

В поле <A> задается номер или имя очереди. Поле определяет количество мест в очереди, освобождаемое транзактом. Это число не должно превышать текущую длину очереди. Если поле пусто, длина очереди уменьшается на единицу.

Внесите следующие изменения в программу примера 1:

GENERATE 8 ; генерация транзактов каждые 8 минут
 QUEUE OCH1 ; занятие очереди och1
 SEIZE 1 ; занятие прибора обслуживания 1
 DEPART OCH1 ; освобождение очереди och1
 ADVANCE 10,5 ; задержка транзакта на время от 5 до 15 минут
 RELEASE 1 ; освобождение прибора 1
 TERMINATE 1 ; уничтожение транзакта
 START 100 ; запуск процесса моделирования для 100 транзактов

В стандартном отчете должен появиться блок статистики по очередям:

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OCH1	27	27	127	1	12.795	103.089	103.908	0

В приведенном примере точное значение таймера в момент завершения прогона модели неизвестно, так как продолжительность прогона устанавливается не по модельному времени, а по количеству транзактов, прошедших через модель. Если необходимо управлять продолжительностью прогона по модельному времени, то в модели используется специальный сегмент, называемый сегментом таймера.

Пример 2. В систему массового обслуживания поступают заявки, распределенные по равномерному закону в интервале от 3 до 7 минут. Обработка поступивших заявок осуществляется также по равномерному закону распределения в интервале от 5 до 9 минут. Необходимо смоделировать работу системы обслуживания в течение 120 минут.

GENERATE 5,2 ; генерация транзактов
 SEIZE 1 ; занятие прибора обслуживания 1
 ADVANCE 7,2 ; задержка транзакта на время от 5 до 9 минут
 RELEASE ; освобождение прибора 1
 TERMINATE ; уничтожение обработанных транзактов без уменьшения ; счетчика
 GENERATE 120 ; сегмент таймера: генерация транзакта через 120 мин
 TERMINATE 1 ; сегмент таймера: уничтожение транзакта
 START 1 ; задание числа счетчика завершений

Стандартный отчет:

START TIME		END TIME		BLOCKS	FACILITIES	STORAGES	
0.000		120.000		7	1	0	
LABEL	LOC	BLOCK TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE		25		9	0
	2	SEIZE		16		0	0
	3	ADVANCE		16		1	0

4	RELEASE	15	0	0
5	TERMINATE	15	0	0
6	GENERATE	1	0	0
7	TERMINATE	1	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	16	0.947	7.102	1	17	0	0	0	9

Как следует из отчета, за 120 минут было сгенерировано 25 транзактов в условном вложенном цикле, который отражает основную работу моделируемой системы. Обработано и выведено из системы 15 транзактов. На момент окончания моделирования в устройство поступил 17-й по счету транзакт внутреннего цикла. Это число показано в поле OWNER статистики устройства. Среднее время занятости устройства составляет 7,102 минут. Коэффициент использования устройства равен 0,947. Число транзактов, ожидающих специальных условий для устройства, равно 9. Поле DELAY статистики устройства заполнено (число 9) потому, что работа внутреннего цикла была прервана внешним циклом путем назначенного времени работы (число 120).

В приведенной модели из двух сегментов, первый (основной) сегмент выполняет те же функции, что и в предыдущем примере. Однако поле <A> блока TERMINATE в первом сегменте пусто, т.е. уничтожаемые транзакты не уменьшают содержимого счетчика завершенных. Во втором сегменте блок GENERATE создаст первый транзакт в момент модельного времени, равный 120. Но этот транзакт окажется и последним в данном сегменте, так как, войдя в блок TERMINATE, он обратит в 0 содержимое счетчика завершенных, установленное оператором START равным 1. Таким образом, в этой модели гарантируется завершение прогона в определенный момент модельного времени, а точное количество транзактов, прошедших через модель, не определено. Если по условию задачи требуется, чтобы все сгенерированные транзакты были обработаны, используется поле <D> блока GENERATE. Подобное условие задачи часто требуется для уточнения характера работы системы, когда обслуживание заявок осуществляется медленнее, чем поступление этих заявок в систему.

Пример 3. В систему массового обслуживания поступает фиксированное число заявок через интервалы времени, распределенные равномерно от 3 до 7 минут. Обработка заявок осуществляется также по равномерному закону в интервале времени от 5 до 9 минут. Необходимо смоделировать генерирование и обработку в системе 100 заявок.

GENERATE 5,2,,100 ; генерация 100 транзактов
 SEIZE 1 ; занятие прибора обслуживания 1
 ADVANCE 7,2 ; задержка транзакта на время от 5 до 9 минут
 RELEASE 1 ; освобождение прибора 1

TERMINATE 1 ; уничтожение транзактов без уменьшения счетчика
 START 100 ; задание числа счетчика завершений

Стандартный отчет:

START TIME		END TIME	BLOCKS	FACILITIES	STORAGES
0.000		715.916	5	1	0

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

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	100	0.991	7.095	1	0	0	0	0	0

Число 100 установлено в поле <D> блока GENERATE и в поле <A> оператора START. Несмотря на то, что время генерации транзактов в примере меньше времени обработки, поле CURRENT_COUNT файла стандартного отчета нулевое, так как число сгенерированных транзактов равно числу счетчика завершений. Моделирование системы будет происходить до тех пор, пока не будут сгенерированы и выведены все 100 транзактов.

1.3. Моделирование многоканальных устройств

Пример 4. Пусть в вычислительной системе имеются два компьютера (интенсивность обработки заданий одинаковая). В вычислительную систему на обработку поступают задания. Интервал времени между двумя последовательными поступлениями заданий подчиняется равномерному закону распределения в интервале 1–11 минут. Перед вычислительной системой допустима очередь заданий, длина которой не ограничена. Время выполнения задания также равномерно распределено в интервале 1–19 минут.

Смоделировать обработку 100 заданий.

VS STORAGE 2 ; задать МКУ (емкость 2)
 GENERATE 6,5 ; генерация транзактов каждые 1–11 минут
 QUEUE OCH1 ; занятие очереди och1
 ENTER VS ; занятие МКУ
 DEPART OCH1 ; освобождение очереди och1
 ADVANCE 10,9 ; задержка транзакта на время от 1 до 19 минут
 LEAVE VS ; освобождение МКУ
 TERMINATE 1 ; уничтожение транзакта
 START 100 ; запуск процесса моделирования для 100 транзактов

Если параллельно работающие устройства являются одинаковыми, то для их моделирования может использоваться объект – мно-

гоканальное устройство (МКУ) или накопитель, память. Формат оператора:

Метка STORAGE A

Метка – имя МКУ; <A> – емкость МКУ (количество однотипных устройств, входящих в МКУ). Если память имеет емкость равную единице, то она аналогична устройству, осуществляющему одноканальное обслуживание. Различие: обслуживание в устройстве может быть прервано транзактом с более высоким приоритетом, чего не допускается при работе с памятью.

Блоки SEIZE – RELEASE заменены, соответственно, на блоки ENTER – LEAVE, моделирующие работу многоканального устройства. Блок входа в устройство ENTER имеет следующий формат записи:

ENTER A,[B]

В поле <A> указывается номер или имя МКУ, поле задает емкость МКУ (по умолчанию 1). Если значение поля равно нулю, то транзакт не задерживается на входе, а блок рассматривается как нерабочий. Также транзакт может быть задержан на входе в блок, если многоканальное устройство заполнено или имеющейся емкости недостаточно, или устройство в данный момент недоступно.

Блок выхода из устройства LEAVE имеет следующий формат:

LEAVE A,[B]

Поле <A> определяет номер (имя) МКУ. Блок LEAVE освобождает заданное число единиц МКУ, определенное в поле (по умолчанию 1). Занятый объем МКУ уменьшается на число освобождаемых единиц, оставшаяся емкость увеличивается на ту же величину. Число освобождаемых единиц не должно превышать текущее содержимое МКУ. Счетчик числа входов не изменяется.

Стандартный отчет (обозначения приведены в таблице 1.2):

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES				
0.000	599.856	7	0	1				
NAME	VALUE							
OCH1	10001.000							
VS	10000.000							
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY			
	1	GENERATE	100	0	0			
	2	QUEUE	100	0	0			
	3	ENTER	100	0	0			
	4	DEPART	100	0	0			
	5	ADVANCE	100	0	0			
	6	LEAVE	100	0	0			
	7	TERMINATE	100	0	0			
QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY	
OCH1	4	0	100	28	0.727	4.359	6.054	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
VS	2	2	0	2	100	1	1.756	0.878	0	0

Таблица 1.2

Обозначение	Описание
STORAGE	имя МКУ
CAP.	емкость МКУ, заданная оператором STORAGE
REM.	количество единиц свободной емкости в конце периода моделирования
MIN.	минимальное количество емкости за используемый период
MAX.	максимальное количество емкости за используемый период
ENTRIES	количество входов в МКУ за период моделирования
AVL.	состояние готовности МКУ в конце периода моделирования (1 – МКУ готов, 0 – нет)
AVE.C.	среднее значение занятой емкости за период моделирования
UTIL.	средний коэффициент использования всех устройств, входящих в МКУ
RETRY	количество транзактов, ожидающих специальных условий, зависящих от состояния МКУ
DELAY	количество транзактов, ожидающих занятия или освобождения устройства МКУ

1.4. Моделирование систем с использованием блоков передачи управления, работа с логическими ключами

Блок передачи транзактов TRANSFER служит для изменения маршрута движения транзакта. Блок TRANSFER имеет следующий формат:

TRANSFER [A],B,[C],[D]

Поле <A> задает режим выбора следующего блока, к которому должно перейти сообщение. Поля и <C> задают возможные значения номеров следующих блоков или их положение. Поле <D> представляет собой константу, используемую для вычисления возможных адресов движения транзактов. Оператор имеет пять основных режимов работы.

Безусловный режим выбора

Если операнд <A> отсутствует, то блок TRANSFER работает в безусловном режиме. Входящее в блок TRANSFER сообщение переходит к блоку, указанному в поле . Если сообщение в этот блок войти не может, попытка направить сообщение к какому-либо другому блоку не производится.

Пример 5. На вход СМО поступают три потока заявок: заявки первого потока имеют наивысший приоритет 3 и поступают с интервалом 5–15 минут; заявки второго потока – приоритет 2 и интервал поступления 15–25 минут; третьего потока – наименьший приоритет 1 и интервал поступления 25–35 минут. Время обработки заявки от 4 до 10 минут. Интервалы времени между двумя последовательными поступлениями заявок и время обработки заявки подчиняются равномерному закону распределения. Требуется смоделировать процесс обработки 1000 заявок.

```

GENERATE 10,5,,3 ; генерация первого потока заявок с высшим
                  ; приоритетом
TRANSFER, MET1 ; безусловная передача управления на метку
MET1
GENERATE 20,5,,2 ; генерация второго потока заявок с приоритетом 2
TRANSFER ,MET1 ; безусловная передача управления на метку
MET1
GENERATE 30,5,,1 ; генерация третьего потока с низшим приоритетом
MET1 ASSIGN 1,PR ; присваивание параметру 1 транзакта значения,
                  ; соответствующего его приоритету
QUEUE P1 ; занятие очереди, имеющей номер приоритета
          ; транзакта
SEIZE 1 ; занятие устройства 1
DEPART P1 ; освобождение соответствующей очереди
ADVANCE 7,3 ; обработка транзакта
RELEASE 1 ; освобождение устройства 1
TERMINATE 1 ; удаление транзакта
START 1000 ; моделирование процесса прохождения 1000 заявок

```

Стандартный отчет показывает, что на момент окончания моделирования 272 сгенерированных заявки остались в очереди. Из них только одна заявка с наивысшим приоритетом не была обработана. Фрагмент отчета:

START TIME		END TIME	BLOCKS	FACILITIES	STORAGES	
0.000		6953.277	12	1	0	
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	696		0	0
	2	TRANSFER	696		0	0
	3	GENERATE	346		0	0
	4	TRANSFER	346		0	0
	5	GENERATE	231		0	0
MET1	6	ASSIGN	1273		0	0
	7	QUEUE	1273		272	0
	8	SEIZE	1001		1	0
	9	DEPART	1000		0	0
	10	ADVANCE	1000		0	0
	11	RELEASE	1000		0	0

	12	TERMINATE		1000		0	0		
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	1001	0.998	6.932	1	1273	0	0	0	272
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY	
1	230	230	231	0	114.302	3440.574	3440.574	0	
2	42	42	346	0	26.288	528.296	528.296	0	
3	2	1	696	2	0.427	4.268	4.281	0	

Статистический режим выбора

Значение операнда <A> задает вероятность перехода транзакта к блоку, указанному в поле <C>. Остальные сообщения направляются по альтернативному адресу к блоку, указанному в поле , или к следующему по номеру блоку, если операнд пропущен. Если вычисленное значение аргумента меньше или равно нулю, то произойдет безусловная передача сообщения к блоку, указанному в поле . Если же значение аргумента больше или равно 1000, произойдет безусловная передача сообщений к блоку, указанному в поле <C>.

Пример 6. Интервал времени поступления заявок в СМО распределен равномерно в диапазоне 3–7 минут. Обработка заявок осуществляется двумя обслуживающими каналами. Поступление заявок в первый или второй канал происходит с вероятностью 0,3 и 0,7, соответственно. Время обслуживания заявок каждым каналом распределено равномерно в интервале 5–9 минут. Необходимо осуществить обработку 100 заявок.

```

GENERATE 5,2,,100 ; генерация потока 100 заявок
TRANSFER .7,MET1,MET2 ; передача 70% заявок в устройство 2
MET1 SEIZE 1 ; занятие устройства 1
ADVANCE 7,2 ; обработка транзакта
RELEASE 1 ; освобождение устройства 1
TRANSFER ,EX ; безусловный переход на метку EX
MET2 SEIZE 2 ; занятие устройства 2
ADVANCE 7,2 ; обработка транзакта
RELEASE 2 ; освобождение устройства 2
EX TERMINATE 1 ; удаление транзакта
START 100 ; моделирование процесса прохождения
100 заявок

```

Стандартный отчет показывает, что 66 заявок из 100 обработано вторым каналом, а 34 – первым. Фрагмент отчета:

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	34	0.497	7.330	1	0	0	0	0	0
2	66	0.917	6.972	1	0	0	0	0	0

При исследовании различных систем методом имитационного моделирования часто возникает необходимость выполнения много-

кратных прогонов модели. Для того чтобы выполнить несколько прогонов модели, можно использовать оператор **CLEAR**. В примере 6 после оператора **START 100** нужно добавить операторы

CLEAR

START 100 ; моделирование еще 100 заявок (второй прогон)

Фрагмент отчета второго прогона:

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	31	0.399	6.819	1	0	0	0	0	0
2	69	0.891	6.833	1	0	0	0	0	0

Режим BOTH

Если в поле <A> стоит зарезервированное слово **BOTH**, блок **TRANSFER** работает в режиме условного перехода с одним альтернативным адресом. В этом режиме в поле указывается блок, к которому направляется транзакт (основной адрес). Если это сделать не удастся, сообщение пытается перейти к блоку, указанному в поле <C> (альтернативный адрес). Если сообщение не сможет перейти ни к одному, ни к другому блоку, оно остается в блоке **TRANSFER** и будет повторять в том же порядке попытки перехода при каждом просмотре списка текущих событий до тех пор, пока не сможет выйти из блока **TRANSFER**. Ниже приведен фрагмент программы, в котором сообщение сначала пытается перейти к блоку **TRY1**; если оно не может войти в этот блок, оно пытается войти в блок **TRY2**. Иначе сообщение остается в списке текущих событий и повторяет эти попытки до тех пор, пока не выйдет из блока **TRANSFER**.

```
TRANSFER BOTH, TRY1, TRY2
```

```
TRY1 SEIZE 1
```

```
TRY2 SEIZE 2
```

Режим ALL

Если в поле <A> стоит зарезервированное слово **ALL**, блок **TRANSFER** работает в режиме условного перехода со многими альтернативами. В этом режиме каждое входящее сообщение прежде всего пытается перейти к блоку, указанному в поле (основной адрес). Если сообщение в этот блок войти не может, то последовательно проверяются блоки в поисках первого, способного принять это сообщение, включая блок, указанный операндом <C> (последний адрес). Номер каждого проверяемого блока вычисляется как сумма номера предыдущего блока и шага, заданного операндом <D>:

$$n + m, n + 2m, n + 3m, \dots l,$$

где n – номер блока, указанного в поле ; m – значение шага, заданного в поле <D>; l – номер блока, указанного в поле <C>. Этот номер должен быть больше номера блока, указанного в поле , на величину, кратную шагу m . Как только первый блок, способный при-

нять сообщение, будет найден, сообщение входит в этот блок и оттуда продолжает свое дальнейшее движение. Если сообщение не может перейти ни к одному из указанных блоков, оно остается в блоке TRANSFER и повторяет данную процедуру при каждом просмотре списка текущих событий до тех пор, пока не выйдет из блока.

Если устройства в примере 6 равнозначны (вероятность поступления заявки в первый или второй канал равна 0,5), оператор, передающий заявку в свободное устройство, должен выглядеть следующим образом:

TRANSFER ALL,MET1,MET2,4

Блоки следует располагать таким образом, чтобы разность между номерами блоков, указанных в полях и <C>, была кратна шагу, указанному в поле <D>. Например, при использовании оператора

TRANSFER ALL, 60, 120, 25

режим ALL недопустим, потому что разность между номерами блоков, записанных в полях и <C>, не является кратной шагу, указанному в поле <D>.

Следует отметить, что каждый раз, когда интерпретатор при просмотре списка текущих событий обнаруживает сообщение, задержанное в блоках TRANSFER BOTH или TRANSFER ALL, он пытается продвинуть сообщение, начиная с блока, указанного в поле . Следовательно, в режиме BOTH в тех случаях, когда возможен переход к обоим блокам (и <C>), блок имеет некоторое преимущество. Аналогично, в режиме ALL в случае, когда возможен переход к нескольким блокам, блоки с меньшими номерами имеют некоторое преимущество перед блоками с большими номерами.

Режим PICK

Если в поле <A> стоит зарезервированное слово PICK, блок TRANSFER случайным образом выбирает один блок, к которому должно быть направлено сообщение из последовательности блоков с номерами $n, n+1, n+2, \dots, m$ (n - номер блока, указанного в поле , а m - номер блока, указанного в поле <C>). Все блоки, включая указанные в полях и <C>, выбираются с одинаковой вероятностью, равной $1/((m-n)+1)$. Если сообщение не может сразу перейти к следующему блоку, то оно будет ждать в блоке TRANSFER до тех пор, пока не будет снято блокирующее условие. Номер блока в поле <C> должен быть больше или равен $n+1$. В примере:

TRANSFER PICK,10,19

сообщение, вошедшее в блок TRANSFER, пытается войти в один из 10 блоков (10,11,...19) с равной вероятностью: 1/10.

Помимо блока TRANSFER, потоком сообщений может управлять блок TEST, который определяет направление движения транзакта

в зависимости от выполнения некоторого условия (алгебраического соотношения). Блок TEST имеет следующий формат:

TEST X A,B,[C]

Операнды <A> и – сравниваемые величины, которые могут быть именем, любым целым числом или СЧА. Во вспомогательном поле операции <X> оператора описания блока TEST записывается знак логической операции – один из шести условных операторов:

L – меньше; отношение истинное, если значение аргумента поля <A> меньше значения аргумента поля ;

LE – меньше или равно;

E – равно;

NE – не равно;

G – больше;

GE – больше или равно.

Если отношение СЧА, заданных в полях <A> и , истинно, сообщение переходит к следующему блоку. Если отношение ложно, сообщение переходит к блоку, номер которого задан полем <C>. Если поле <C> пусто, транзакт задерживается в блоке TEST до выполнения условия.

Например:

TEST G M1,500,MET

SEIZE 1

...

MET SEIZE 2

Если значение времени пребывания транзакта в модели больше 500, то переходим к следующему по номеру блоку, ложно – к метке MET.

Пример 7. Интервал времени поступления заявок в СМО распределен равномерно в диапазоне от 1 до 11 минут. Обработка заявок длится от 1 до 19 минут. Так как обслуживающее устройство не успевает обрабатывать заявки, образуется очередь. Если длина очереди больше трех, то заявка выводится из системы без обработки. Смоделировать прохождение 100 заявок. Определить число заявок, покинувших систему необслуженными.

GENERATE 6,5 ; генерация заявок

TEST L Q\$OCH,3,EX ; если условие «длина очереди меньше 3» ложно,
; удалить заявку

QUEUE OCH ; занятие очереди

SEIZE 1 ; занятие устройства 1

DEPART OCH ; освобождение очереди

ADVANCE 10,9 ; обслуживание заявки

RELEASE 1	; освобождение устройства 1
EX TERMINATE 1	; удаление транзакта
START 100	; моделирование процесса прохождения 100 заявок

Стандартный отчет показывает, что при полной загрузке устройства было обслужено (прошло через блок RELEASE) 66 заявок, 34 получили отказ из-за заполнения очереди. Моделирование было остановлено, как только из системы было удалено 100 транзактов.

Блок проверки состояния элементов GATE разрешает движение транзактов при определенном состоянии устройств, накопителей, ключей. Формат записи блока GATE:

GATE X A, [B]

Поле <A> содержит имя или номер объекта, для которого проводится проверка во вспомогательном поле операции <X>. Поле содержит номер или имя блока, в который переходит транзакт, если проверяемое условие не выполняется. Мнемонические обозначения проверяемого условия записываются непосредственно после поля GATE.

Состояние устройства описывается следующими условиями:

- U** – устройство используется, занято;
- NU** – устройство не используется, свободно;
- I** – устройство обслуживает прерывание;
- NI** – устройство работает без прерывания;
- FV** – устройство доступно;
- FNV** – устройство недоступно.

Состояние накопителя (памяти) описывается следующими условиями:

- SE** – накопитель пуст;
- SNE** – накопитель не пуст;
- SF** – накопитель заполнен;
- SNF** – накопитель не заполнен;
- SV** – накопитель доступен;
- SNV** – накопитель не доступен.

Состояние логического ключа описывается следующими условиями:

- LR** – ключ в состоянии «выключен»;
- LS** – ключ в состоянии «включен».

Существует два режима работы блока GATE: режим отказа, или условного входа и режим перехода, или безусловного входа. При работе в режиме отказа блок GATE блокирует транзакт, если соответствующий объект не находится в требуемом состоянии. Если же поставленное условие удовлетворяется, блок разрешает вход транзактов. Если в поле указано имя (номер) блока, то вместо отказа блок GATE будет направлять транзакт на указанный блок.

Пример 8. Поток заявок поступает в СМО с двумя обслуживающими каналами равномерно с интервалом 4–6 минут. Если первый канал занят, то заявки поступают на обработку во второй канал. Время обработки первого канала распределено равномерно в промежутке 8–10 минут, второго – 12–14 минут. Смоделировать процесс обработки 100 заявок.

GENERATE 5,1 ; генерация потока 100 заявок
 GATE NU 1,MET2 ; если устройство 1 занято, отправить заявку на
 ; устройство 2
 SEIZE 1 ; занятие устройства 1
 ADVANCE 9,1 ; обработка транзакта
 RELEASE 1 ; освобождение устройства 1
 TRANSFER ,EX ; безусловный переход на метку EX
 MET2 SEIZE 2 ; занятие устройства 2
 ADVANCE 13,1 ; обработка транзакта
 RELEASE 2 ; освобождение устройства 2
 EX TERMINATE 1 ; удаление транзакта
 START 100 ; моделирование процесса прохождения 100 заявок

Для иллюстрации работы блока проверки состояния элементов GATE с накопителем (памятью) изменим условие предыдущего примера.

Пример 9. Поток заявок поступает в накопитель с допустимой емкостью, равной 3 единицам, равномерно с интервалом 4–6 минут. После накопителя заявки поступают на обработку в первый обслуживающий канал. Если накопитель заполнен, то заявки поступают на обработку во второй канал. Смоделировать процесс обработки 100 заявок.

NAK STORAGE 3 ; накопитель емкостью 3 единицы
 GENERATE 5,1 ; генерация потока 100 заявок
 GATE SNF NAK,MET2 ; если накопитель заполнен, отправить заявку на
 ; устройство 2
 ENTER NAK ; добавление заявки в накопитель
 QUEUE 1 ; занятие очереди к устройству 1
 SEIZE 1 ; занятие устройства 1
 DEPART 1 ; освобождение очереди к устройству 1
 LEAVE NAK ; удаление заявки из накопителя
 ADVANCE 9,1 ; обработка транзакта
 RELEASE 1 ; освобождение устройства 1
 TRANSFER ,EX ; безусловный переход на метку EX
 MET2 QUEUE 2 ; занятие очереди к устройству 2
 SEIZE 2 ; занятие устройства 2
 DEPART 2 ; освобождение очереди к устройству 2
 ADVANCE 13,1 ; обработка транзакта

RELEASE 2 ; освобождение устройства 2
 EX TERMINATE 1 ; удаление транзакта
 START 100 ; моделирование процесса прохождения 100 заявок

Анализ стандартного отчета показывает, что использование накопителя уменьшает общее время моделирования процесса прохождения заявок и увеличивает загрузку обслуживающих каналов.

Логические ключи

Логические ключи служат для моделирования объектов с двумя логическими состояниями «включено», «выключено». Управление ключами осуществляется транзактами, продвигающимися по модели. Транзакт может установить ключ в состояние «включено», сбросить его – перевести в состояние «выключено» или инвертировать, т.е. изменить состояние ключа на противоположное значение.

Оператор установки начальных значений INITIAL записывается вне тела основной программы. Формат записи оператора INITIAL:

INITIAL A,B

В поле <A> вводится имя устанавливаемой величины, в поле – начальное значение. Для инициализации логического ключа в поле <A> записывается выражение LS\$имя_ключа или LSj, где j – номер ключа; в поле записывается 0 или 1. В начале моделирования все ключи находятся в состоянии «выключено». Для включения, выключения и инвертирования ключей служит оператор LOGIC.

Формат записи блока LOGIC имеет вид:

LOGIC X A

Во вспомогательном поле операций <X> задается мнемоническое обозначение операции. В поле <A> задается номер или имя логического ключа, ссылка на имя ключа может происходить в любом месте программы. В поле <X> могут быть заданы следующие мнемонические обозначения:

S – установить в состояние «включен» логический переключатель;

R – сбросить логический переключатель;

I – инвертировать логический переключатель.

Формат записи блока GATE для проверки логических ключей имеет вид:

GATE X A,B

В поле <X> задается один из следующих логических операторов:

LS – логический переключатель «включен»;

LR – логический переключатель «сброшен».

В поле <A> задается номер или имя логического ключа, состояние которого проверяется. В поле задается номер блока, к которому переходит транзакт, если логический оператор вспомогательного поля <X>

имеет значение «ложь». Если значение логического оператора «истина», транзакт переходит к следующему по номеру блоку.

Пример 10. На обработку в СМО поступает поток заявок с интервалом времени 9–11 минут. Нечетные по номеру заявки обрабатываются на первом канале, время обработки 15 минут, четные – на втором канале со временем 17 минут. Смоделировать процесс прохождения 101 заявки.

```
INITIAL LS$KEY1,0 ; инициализация ключа с именем KEY1 (эту
; строку можно опустить)
GENERATE 10,1,,101 ; генерация потока заявок
LOGIC I KEY1 ; инвертирование ключа
GATE LS KEY1,MET2 ; разделение потока четных и нечетных заявок
MET1 SEIZE 1 ; занятие устройства 1
ADVANCE 15 ; обработка заявки
RELEASE 1 ; освобождение устройства 1
TRANSFER ,EX ; безусловная передача управления
MET2 SEIZE 2 ; занятие устройства 2
ADVANCE 17 ; обработка заявки
RELEASE 2 ; освобождение устройства 2
EX TERMINATE 1 ; удаление транзакта
START 101 ; моделирование прохождения 101 заявки
```

Фрагмент стандартного отчета:

FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	51	0.744	15.000	1	0	0	0	0	0
2	50	0.827	17.000	1	0	0	0	0	0

LOGICSWITCH	VALUE	RETRY
KEY1	1	0

Формирование потока четных и нечетных по номеру заявок осуществляется блоками LOGIC и GATE, логический переключатель «включен». Через первое устройство проходит 51 нечетная по номеру заявка, а через второе – 50 четных по номеру заявок. Если в примере вместо блока GATE с полем LS поставить блок GATE с полем LR, то четный и нечетный потоки поменяются местами: первое устройство будет обслуживать четные заявки, а второе устройство – нечетные. Если же в поле <X> блока LOGIC поставить S или R, то обработка всех поступающих заявок будет осуществляться одним из устройств. В стандартном отчете появилось поле LOGICSWITCH, которое определяет заданное имя логического ключа KEY1. Поле VALUE определяет значение логического переключателя в конце периода моделирования. Поле RETRY определяет количество транзактов, ожидающих наступления специальных условий, зависящих от состояния логического переключателя.

1.5. Моделирование систем с использованием копий транзактов и синхронизации их движения

Достаточно часто встречаются задачи, связанные с моделированием параллельных процессов, синхронизируемых в определенные моменты времени. Примером может служить производственный процесс изготовления изделия из нескольких деталей. Прежде чем поступить к месту сборки изделия, детали проходят обработку в соответствии с заданными технологическими маршрутами. При этом каждая деталь движется по своему маршруту независимо от других, но процесс сборки может начаться, только когда все комплектующие поступят к месту сборки, то есть будет осуществлена синхронизация движения деталей.

Для моделирования таких систем используются семейства транзактов – множества транзактов, являющихся копиями одного и того же исходного транзакта, либо его копий. Транзакты, принадлежащие одному семейству, могут быть синхронизированы.

Блок **SPLIT** так же, как и блок **GENERATE**, предназначен для создания транзактов. Но в отличие от блока **GENERATE** он не создает самостоятельных транзактов, а лишь генерирует заданное число копий, идентичных входящему в него транзакту. Блок **SPLIT** имеет следующий формат:

SPLIT A,[B],[C]

Блок **SPLIT** выполняет функцию копирования входящего в него транзакта, который называется исходным или порождающим. Все копии формируются в момент входа порождающего транзакта в блок **SPLIT**. В поле **<A>** задается число создаваемых копий. Исходный транзакт и копии являются равноправными и могут проходить снова через любое число блоков **SPLIT**. Если значение поля **<A>** равно нулю, то блок **SPLIT** не выполняет никаких операций. Поле **** задает номер следующего блока, к которому переходят копии исходного транзакта. Исходный транзакт поступает к следующему за блоком **SPLIT** блоку. В поле **<C>** может быть задан номер параметра, используемого для присвоения копиям последовательных номеров. Например:

SPLIT 3, MET, 1

Создаются три копии, которые передаются по метке **MET**. Номер копии записан в параметре **P1** транзакта.

Все транзакты, полученные копированием, а также копии копий принадлежат к одному ансамблю и далее к этому ансамблю можно применять специальные блоки, осуществляющие обработку ансамблей транзактов (**ASSEMBLE**, **MATCH**, **GATHER**).

Для объединения заданного числа транзактов одного семейства используется блок **ASSEMBLE**, имеющий следующий формат:

ASSEMBLE A

Блок ASSEMBLE осуществляет сборку заданного числа транзактов. Поле <A> задает число транзактов, участвующих в сборке. Первый из транзактов, принадлежащих некоторому семейству, вошедший в блок ASSEMBLE, задерживается до прихода в блок последнего необходимого для сборки транзакта из данного семейства. После сборки из блока ASSEMBLE выходит только один транзакт, который переходит в следующий по номеру блок, остальные транзакты уничтожаются. В одном и том же блоке ASSEMBLE возможна одновременная сборка транзактов нескольких семейств. Транзакты одного семейства могут накапливаться одновременно в различных блоках ASSEMBLE.

Пример 11. Интервал поступления деталей подчиняется равномерному закону в диапазоне 6–10 минут. Обработка производится параллельно двумя рабочими, каждый из которых выполняет свою операцию независимо от другого. Время выполнения операции распределено равномерно в интервале 2–8 минут. Смоделировать процесс обработки 100 деталей, определить коэффициент занятости рабочих.

GENERATE 8,2 ; генерация деталей
 SPLIT 1,RAB2 ; начало обработки детали
 SEIZE 1 ; обработка первым рабочим
 ADVANCE 5,3
 RELEASE 1
 TRANSFER ,OUT ; передача транзакта на этап сборки ансамбля
 RAB2 SEIZE 2 ; обработка вторым рабочим
 ADVANCE 5,3
 RELEASE 2
 OUT ASSEMBLE 2 ;окончание обработки детали (этап сборки)
 TERMINATE 1
 START 100

Блок SPLIT создает одну копию транзакта и направляет ее по метке RAB2 на блок SEIZE 2. При этом через блок SPLIT проходит транзакт-родитель на следующий по номеру блок. В блоке ASSEMBLE собираются два транзакта, выходит из него только один. За полный цикл моделирования в блоке ASSEMBLE собираются 200 транзактов, выходит из него только 100 транзактов. Формально блок ASSEMBLE уничтожает 100 транзактов. Коэффициенты занятости рабочих равны 0,618 и 0,621, соответственно.

Действие блока **GATHER** аналогично действию блока ASSEMBLE. Отличие состоит в том, что после накопления в блоке числа транзактов, указанного в поле <A>, они все передаются в следующий блок. Блок GATHER позволяет синхронизировать движение

транзактов одного ансамбля при их движении по одному пути. Например: GATHER 3 – после прихода трех транзактов они поступают в следующий блок.

Пример 12. На производственный участок сборки подшипников поступают обоймы и шарики с интервалом времени 25 ± 4 единицы. На контроль обоймы затрачивается 4 ± 1 единицы времени; контроль шариков производится последовательно со временем 2 ± 1 единицы времени на шарик. Операция сборки требует одновременного поступления обоймы и всех шариков и производится со временем 4 ± 2 единицы. Все процессы подчиняются равномерному закону. Построить программу модели процесса прохождения 80 деталей. В результате моделирования требуется определить коэффициенты занятости рабочих участка.

GENERATE 25,4 ; генерация деталей
 SPLIT 8,RAB2 ; разделение обоймы и шариков
 SEIZE 1 ; контроль обоймы
 ADVANCE 4,1
 RELEASE 1
 TRANSFER ,OUT ; передача транзакта на этап сборки ансамбля
 RAB2 SEIZE 2 ; контроль шариков
 ADVANCE 2,1
 RELEASE 2
 GATHER 8 ; накопление 8 шариков
 OUT ASSEMBLE 9 ; комплектация детали из 8 шариков и одной обоймы
 SEIZE 3 ; сборка
 ADVANCE 4,2
 RELEASE 3
 TERMINATE 1
 START 80

Стандартный отчет показывает, что наибольший коэффициент занятости, равный 0,624 у рабочего участка, производящего контроль шариков.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	80	0	0
	2	SPLIT	80	0	0
	3	SEIZE	80	0	0
	4	ADVANCE	80	0	0
	5	RELEASE	80	0	0
	6	TRANSFER	80	0	0
RAB2	7	SEIZE	640	0	0
	8	ADVANCE	640	0	0
	9	RELEASE	640	0	0
	10	GATHER	640	0	0
OUT	11	ASSEMBLE	720	0	0
	12	SEIZE	80	0	0

13	ADVANCE	80	0	0
14	RELEASE	80	0	0
15	TERMINATE	80	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	80	0.159	4.016	1	0	0	0	0	0
2	640	0.624	1.973	1	0	0	0	0	0
3	80	0.163	4.122	1	0	0	0	0	0

Для синхронизации движения двух транзактов, принадлежащих к одному семейству и движущихся по разным путям, используется два сопряженных блока MATCH, имеющих следующий формат:

MATCH A

Поле <A> задает имя или номер другого блока MATCH, называемого «сопряженным блоком MATCH». Если такого блока нет, происходит останов по ошибке. Блоки MATCH не объединяют синхронизируемые транзакты. Синхронизация осуществляется путем подбора пар транзактов из одного семейства и задержки этих транзактов до тех пор, пока оба транзакта из одной пары не поступят в заданные точки модели. Транзакты, для которых выполнилось условие синхронизации, переходят к следующему по номеру блоку. В одной паре блоков MATCH могут одновременно находиться в состоянии синхронизации пары транзактов из различных семейств. Возможна также одновременная синхронизация пар транзактов из одного семейства в нескольких блоках MATCH. Следует отметить, что блок MATCH может быть сопряжен сам себе. При этом его действие будет аналогично действию блока GATHER с параметром 2 в поле <A>.

Пример 13. Детали поступают с интервалом времени 300 ± 50 единиц. Обработку производят двое рабочих, которые выполняют по две операции. После первой операции, выполненной первым рабочим со временем 70 ± 20 единиц и вторым со временем 60 ± 30 единиц, производится операция сверки, время выполнения которой принимается равным нулю. После операции сверки выполняется вторая операция первым рабочим со временем 20 ± 10 единиц и вторым – 50 ± 20 единиц. Затем третий рабочий производит сборку изделия из этих деталей со временем 50 ± 20 единиц. Сборка начинается только после поступления всех деталей. Все процессы подчиняются равномерному закону распределения. Построить имитационную программу прохождения 500 деталей. В результате моделирования необходимо определить коэффициенты занятости рабочих.

```

GENERATE 300,50      ; поступление деталей
SPLIT 1,RAB2        ; разделение операций
RAB1 SEIZE 1        ; выполнение первой операции первым рабочим
ADVANCE 70,20
SVER1 MATCH SVER2   ; сверка

```

ADVANCE 20,10 ; выполнение второй операции первым рабочим
 RELEASE 1
 TRANSFER ,RAB3 ; передача детали третьему рабочему для сборки
 RAB2 SEIZE 2 ; выполнение первой операции вторым рабочим
 ADVANCE 60,30
 SVER2 MATCH SVER1 ; сверка
 ADVANCE 30,20 ; выполнение второй операции вторым рабочим
 RELEASE 2
 RAB3 ASSEMBLE 2 ; сборка деталей
 SEIZE 3 ; выполнение сборки третьим рабочим
 ADVANCE 50,20
 RELEASE 3
 TERMINATE 1
 START 500 ; моделирование обработки 500 деталей

Стандартный отчет показывает, что наименьший коэффициент занятости, равный 0,167 у третьего рабочего, производящего сборку деталей. Остальные рабочие заняты 0,314 и 0,348 процентов рабочего времени.

1.6. Моделирование систем с использованием блоков захвата и освобождения приборов

Блок PREEMPT приостанавливает обслуживание транзакта, ранее занявшего устройство, и дает возможность захватить устройство прерывающему транзакту. Блок PREEMPT имеет следующий формат :

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

Поле <A> определяет номер или имя устройства, работа которого прерывается входящим в блок транзактом. Поле задает прерывание по приоритету (PR) или режим прерывания, если операнд опущен. Поле <C> задает номер или имя блока, куда должен попытаться войти прерванный транзакт в этот же момент условного времени. Прерванный транзакт теряет управление устройством, но претендует на право его использования, если только не задан аргумент поля <E>. Поле <D> задает номер параметра, связанного с прерванным транзактом. Поле <E> задает один из следующих режимов:

- режим удаления (RE); задание этого режима означает, что прерванный транзакт более не претендует на пользование устройством. Прерванный транзакт попытается войти в блок, заданный полем <C>.
- если режим RE не задан, т.е. поле <E> пусто, то прерванный транзакт будет вновь пытаться занять устройство.

Блок RETURN предназначен для освобождения ранее захваченного устройства и имеет следующий формат:

RETURN A

В поле <A> задается номер устройства, с которого снимается прерывание.

Пример 14. На обработку в СМО поступают два потока деталей. Интервалы времени поступления распределены равномерно в диапазоне 9–11 минут для первого потока и 23–27 минут для второго потока. Причем второй поток прерывает изготовление деталей первого потока. Время обработки деталей первого потока 4–10 минут, второго потока – 14–16 минут. Смоделировать процесс обработки 100 деталей.

GENERATE 10,1	; генерация первого потока заявок
SEIZE UST	; занятие устройства
ADVANCE 7,3	; обработка заявки первого потока
RELEASE UST	; освобождение устройства
TERMINATE 1	; удаление транзакта
GENERATE 25,2	; генерация второго потока заявок
PREEMPT UST	; прерывание первого потока заявок без удаления
ADVANCE 15,1	; обработка заявки второго потока
RETURN UST	; освобождение захваченного устройства
TERMINATE 1	; удаление транзакта
START 100	; моделирование процесса прохождения 100 заявок

1.7. Моделирование недоступности устройств и накопителей

Иногда требуется моделировать ситуацию, когда в течение определенного времени устройство обслуживания становится недоступным. В этом случае используется блок FUNAVAIL, который делает недоступным устройство до тех пор, пока не вступит в действие блок FAVAIL (эти блоки действуют совместно). Блок FUNAVAIL имеет следующий формат:

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

В поле <A> задается имя или номер устройства, которое должно стать недоступным. В поле задаются режимы работы с транзактами (заявками, сообщениями), использующими устройство в момент, когда оно становится недоступным:

- режим **RE** (режим удаления REMOVE) означает, что занимавший устройство транзакт больше не может претендовать на устройство. Удаленный транзакт попытается войти в альтернативный блок, заданный в поле <C>;
- режим **CO** (режим продолжения CONTINUE) позволяет транзакту использовать устройство в период недоступности;

- если режим не задан (режим по умолчанию), то это означает, что транзакт, использующий устройство, прерывается до тех пор, пока устройство вновь не станет доступным.

В поле <C> задается номер или имя блока, к которому направляется транзакт, использующий устройство в момент, когда оно становится недоступным. В поле <D> задается номер параметра управляющего транзакта. Информацию по полю <D> можно использовать для вычисления времени, которое управляющий транзакт должен провести в списке будущих событий, когда устройство становится недоступным.

Поле <E> можно использовать в следующих режимах:

- режим **RE** указывает на то, что прерванные транзакты (например, блоком **PREEMPT**) больше не могут претендовать на устройство. Эти транзакты пытаются войти в альтернативный блок, заданный в поле <F>;
- режим **CO** допускает использование устройства всеми транзактами во время периода недоступности. Транзакты занимают устройство в порядке их приоритета;
- если режим не задан (режим по умолчанию – пустое поле), то прерванный транзакт покидает список прерывания устройства, и не допускается использование устройства во время периода недоступности.

В поле <F> задается номер или имя альтернативного блока, к которому направляется прерванный транзакт. Поле <F> используется, если в поле <E> задан режим удаления. Когда устройство становится недоступным, поле <G> определяет действия с транзактами, пытающимися занять устройство или сгенерировать на нем прерывание (например, блоком **PREEMPT**) в момент, когда это устройство становится недоступным. Допускаются следующие режимы поля <G>:

- режим **RE** указывает на то, что задержанные транзакты больше не должны претендовать на устройство;
- режим **CO** предполагает, что всем транзактам будет разрешено использовать устройство во время периода его недоступности; транзакты занимают устройство в порядке их приоритета;
- если режим не задан (режим по умолчанию), то это означает, что задержанные транзакты покидают список задержки или список ожидания устройства и не могут больше претендовать на устройство во время периода недоступности.

Поле <H> задает номер или имя блока, к которому направляются задержанные транзакты в режиме **RE** поля <G>.

Блок **FAVAIL** делает доступным заданное устройство после периода недоступности. Блок **FAVAIL** имеет следующий формат :

FAVAIL A

В поле <A> блока FAVAIL задается номер или имя устройства, с которого снимается недоступность.

Пример 15. Смоделировать СМО, в которой поток входящих заявок начинает обслуживаться в одном устройстве и переводится на обслуживание в другое устройство, если первое устройство становится недоступным на какое-то время. Генерация заявок происходит с интервалом времени 3–7 минут. Время обработки заявки распределено равномерно в интервале 6–12 минут. Одно из устройств становится недоступным через каждые 33 ± 2 мин на время 11–15 минут. Определить загрузку каждого из каналов и количество обработанных заявок вторым каналом, если в первом канале должны быть обслужены 100 заявок.

GENERATE 5,2	; генерация потока заявок
GATE FNV 1,UST2	; проверка 1-го канала на недоступность
UST1 SEIZE 1	; занятие 1-го канала
ADVANCE 9,3	; обработка заявки первым устройством
RELEASE 1	; освобождение первого канала
TERMINATE 1	; удаление транзакта
UST2 SEIZE 2	; занятие 2-го канала
ADVANCE 9,3	; обработка заявки вторым устройством
RELEASE 2	; освобождение второго канала
TRANSFER ,EX	; безусловный переход
GENERATE 33,2	; генерация периода недоступности
FUNAVAIL 1	; моделирование недоступности устройства
ADVANCE 13,2	; на время 11–15 минут
FAVAIL 1	; снятие недоступности с устройства
EX TERMINATE	
START 100	

Пустое поле второго блока TERMINATE позволяет не влиять на регистрацию счетчика завершений, который связан с обслуживанием заявок в первом канале. В соответствии с условием, все время, когда одно устройство недоступно, транзакты отправляются блоком GATE на второе устройство до тех пор, пока первое устройство не обслужит 100 транзактов. При этом ограничений на обслуживание транзактов вторым устройством нет.

Аналогичные команды применяются к накопителям. Блок SUNAVAIL делает накопитель недоступным, SAVAIL переводит накопитель из состояния недоступности в состояние доступности. Форматы записи блоков имеет вид:

SUNAVAIL A

SAVAIL A

В поле <A> блока задается имя накопителя.

Пример 16. Пусть заявки поступают в накопитель с допустимой емкостью, равной 3 единицам, равномерно с интервалом времени 4–6 минут. Если заявки застают накопитель в состоянии недоступности, то они направляются в первый канал, в противном случае поступают на обработку во второй канал. Время обработки для первого устройства распределено равномерно в интервале 12–14 минут, а для второго – 8–10 минут. Смоделировать процесс обработки 100 заявок, если состояние недоступности накопителя происходит через каждые 29 ± 1 минут в течение 17 минут.

NAK STORAGE 3	; накопитель емкостью 3 единицы
GENERATE 5,1,,100	; поступление заявок в накопитель
GATE SV NAK,UST2	; анализ доступности накопителя
ENTER NAK	; заявка входит в накопитель
UST1 SEIZE 1	
LEAVE NAK	; заявка покидает накопитель
ADVANCE 13,1	; и обрабатывается на первом канале
RELEASE 1	
TRANSFER ,EX	
UST2 SEIZE 2	; занятие второго канала
ADVANCE 9,1	; обработка заявок вторым устройством
RELEASE 2	;освобождение второго канала
EX TERMINATE 1	
GENERATE 29,1	; генерация периода недоступности для накопителя
SUNAVAIL NAK	; каждые 29 ± 1 минут
ADVANCE 17	; в течение 17 минут накопитель недоступен
SAVAIL NAK	
TERMINATE	
START 100	

В данном примере поступающие в СМО заявки сначала буферизуются в накопителе емкостью в 3 единицы. Затем, в зависимости доступности или недоступности накопителя, они направляются в один из каналов (устройств) с помощью блока GATE. В операционном поле <X> блока GATE задан логический операнд SNV – многоканальное устройство (накопитель) не доступно. В поле <A> определено имя накопителя NAK, состояние которого проверяется. Если логический операнд имеет значение «истина», то транзакт пытается войти в следующий по номеру блок. Если же операнд имеет значение «ложь», то транзакт пытается войти в альтернативный блок, заданный полем (блок с меткой UST2).

1.8. Организация циклов

Блок ASSIGN является основным средством для изменения значений параметров транзактов. Параметры транзактов принимают значения из множества целых чисел. Каждый транзакт может иметь один или более параметров. Блок ASSIGN заменяет, увеличивает или уменьшает текущее значение параметра транзакта на заданное значение. Формат записи блока ASSIGN имеет вид:

ASSIGN A,B,C

В поле <A> указывается номер или имя параметра транзакта, подлежащего изменению. Если в поле <A> после имени параметра стоит знак «+» или «-», то значение операнда добавляется или вычитается из текущего содержимого параметра. Если знак не указан, то параметру транзакта присваивается значение операнда . В поле <C> может быть указано имя или номер функции-модификатора, действующей аналогично функции-модификатору в поле блока GENERATE. Если поле <C> не задано, модификация не производится.

Блок LOOP используется для организации циклов. Как правило, блок LOOP применяется в паре с блоком ASSIGN. Формат записи блока:

LOOP A,B

В поле <A> задается номер параметра транзакта, значение которого используется для организации количества повторений (параметр цикла). При входе транзакта в данный блок значение параметра уменьшается на единицу. Если оно становится равным нулю, то транзакт переходит в следующий блок, иначе транзакт переходит в блок, указанный в поле .

Оператор EQU предназначен для присвоения числовых значений именам, используемым в модели. Назначение числовых значений может происходить при выполнении каких-либо выражений, задании стандартных числовых атрибутов и т.д. Результат вычисления выражения преобразуется к целому виду.

Пример 17. Пусть каждая заявка СМО проходит пять циклов обработки в обслуживающем канале. Время обработки распределено равномерно в интервале 5–9 минут. Входной поток заявок также подчиняется равномерному закону с интервалом поступления заявок 3–7 минут. Смоделировать процесс обработки 100 заявок.

MET1 EQU 5	; присвоение числа 5 имени MET1
MET2 EQU 7	; присвоение числа 7 имени MET2
GENERATE 5,2,,100	; генерация потока заявок
ASSIGN 10,MET1	; параметру транзакта под номером 10 (номер ; параметра можно выбрать произвольно) через ; метку MET1 сопоставляется число 5

```

SEIZE 1
CYC5 ADVANCE MET2,2 ; задержка транзакта в приборе обслуживания
LOOP 10,CYC5 ; организация цикла
RELEASE 1
TERMINATE 1
START 100

```

Фрагмент стандартного отчета:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	ASSIGN	100	0	0
	3	SEIZE	100	0	0
CYC5	4	ADVANCE	500	0	0
	5	LOOP	500	0	0
	6	RELEASE	100	0	0
	7	TERMINATE	100	0	0

Задачу повторного пятикратного обслуживания осуществляет блок LOOP, который отправляет транзакты на блок ADVANCE. Из стандартного отчета видно, что через блоки LOOP и ADVANCE проходит в 5 раз больше транзактов, чем через остальные блоки.

1.9. Обработка результатов моделирования средствами GPSS

Ряд числовых атрибутов, характеризующих функционирование моделируемой системы, можно получить с помощью средств для сбора статистики. Для получения оценок математического ожидания (среднего значения) и дисперсии последовательности значений случайных величин, полученных в результате моделирования системы, используются блоки сбора статистики TABLE и TABULATE. Блок TABLE имеет следующий формат:

<NAME> TABLE A,B,C,D

Блок TABLE определяет аргумент, а также число и ширину частотных интервалов. Метка NAME определяет имя таблицы. В поле <A> задается аргумент таблицы – имя переменной, значение которой будет табулироваться. В поле задается верхняя граница первого интервала. В поле <C> задается ширина частотного интервала – разница между верхней и нижней границей каждого частотного класса. В поле <D> задается число частотных интервалов. Желательно так подбирать ширину и количество интервалов, чтобы в полученные интервалы попадало 100% значений аргумента таблицы. Помимо таблицы частот одновременно вычисляются оценки среднего и стандартного отклонения аргумента таблицы. Блоки описания таблиц помещаются в начале программы.

Блок, связанный с оператором TABLE – TABULATE. Для сбора элементов данных сообщение должно войти в блок TABULATE с

тем же именем таблицы, что определено в блоке TABLE. Блок TABULATE имеет следующий формат:

TABULATE A,[B]

Блок TABULATE табулирует текущее значение заданного аргумента. Способ табуляции зависит от режима работы таблицы, который определяется оператором описания таблицы TABLE.

В поле <A> задается номер или имя таблицы, в которую табулируется значение аргумента. В поле задается число единиц, которые должны быть занесены в тот частотный интервал, куда попало значение аргумента. Если поле пусто, эта величина полагается равной единице.

Когда сообщение входит в блок TABULATE, оценивается аргумент таблицы (операнд <A> в операторе TABLE). Если он меньше или равен операнду в операторе TABLE, то выбирается первый частотный класс таблицы. Если аргумент таблицы не подходит для этого класса, то класс выбирается путем деления значения аргумента на операнд <C> оператора TABLE. Нижняя граница частотного класса включается в предыдущий класс. Если таблицы недостаточно для размещения этого значения, то выбирается последний частотный интервал. Затем выбирается целое из частотного класса и счетчик увеличивается на величину, определяемую операндом оператора TABULATE. По умолчанию увеличение происходит на 1. В конце работы оператора TABULATE изменяются значения среднего и стандартного отклонения аргумента таблицы.

Стандартные числовые атрибуты, связанные с описываемым оператором, следующие: TB – среднее значение аргумента; TC – число входов в таблицу; TD – стандартное отклонение.

Пример 18. Интервал поступления заявок в СМО подчиняется равномерному закону распределения с диапазоном 3–7 минут. Обработка заявок происходит параллельно в двух каналах. Время обслуживания заявки первым каналом 13–17 минут, вторым – 15–19 минут. Смоделировать процесс обслуживания поступающих заявок, собрать статистику об очередях при обслуживании 100 заявок. Построить гистограммы распределения длительности пребывания заявки в очереди к первому каналу и длины очереди ко второму каналу.

TAB1 TABLE QT1,5,30,12

TAB2 TABLE Q\$OCH2,10,10,12

GENERATE 5,2

SPLIT 1,UST2 ; распределение заявок по каналам

UST1 QUEUE 1

SEIZE 1 ; обработка заявки на первом канале

DEPART 1

ADVANCE 15,2
 RELEASE 1
 TABULATE TAB1 ; статистика по времени ожидания в первой очереди
 TRANSFER ,EX
 UST2 QUEUE OCH2
 SEIZE 2 ; обработка заявки на втором канале
 DEPART OCH2
 ADVANCE 17,2
 RELEASE 2
 TABULATE TAB2 ; сбор статистики по длине второй очереди
 EX TERMINATE 1
 START 100

По условию заявки распределяются поровну между двумя каналами. В связи с тем, что время обслуживания каждым каналом различно, следует ожидать, что количество обслуженных заявок каждым каналом будет различным. Статистика об очередях собирается автоматически при наличии в системе блоков QUEUE и DEPART. Табулирование значений СЧА осуществляется с помощью оператора TABLE и блока TABULATE.

Формат записи первого оператора TABLE:

TAB1 TABLE QT1, 5, 30, 12

В поле метки находится имя оператора TAB1, в поле <A> задан стандартный числовой атрибут QT1 – среднее время пребывания транзакта (заявки) в очереди под номером 1 (в соответствии с QUEUE 1), в поле – задается верхний предел первого интервала (в примере число 5), в поле <C> задается ширина частотного интервала (в примере число 30), в поле <D> задается число частотных интервалов (12).

Формат записи второго оператора TABLE:

TAB2 TABLE Q\$OCH2, 10, 10, 12

Второй оператор TABLE табулирует величину Q\$OCH2 – длину очереди под именем OCH2. В стандартном отчете указываются следующие величины:

MEAN – среднее значение или оценка математического ожидания;
 STD.DEV – оценка среднеквадратического отклонения;
 RANGE – интервалы группирования;
 FREQUENCY – количество наблюдений, попавших в каждый интервал;
 CUM.% – накопленная частота в процентах.

Фрагмент стандартного отчета:

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM. %
TAB1	136.649	76.413	- -	0	1	1.89
			5.000			

	5.000	-	35.000		5	11.32
	35.000	-	65.000		6	22.64
	65.000	-	95.000		6	33.96
	95.000	-	125.000		6	45.28
	125.000	-	155.000		6	56.60
	155.000	-	185.000		6	67.92
	185.000	-	215.000		7	81.13
	215.000	-	245.000		6	92.45
	245.000	-	275.000		4	100.00
TAB2	56.234		31.145	0		
		-	10.000		3	6.38
	10.000	-	20.000		4	14.89
	20.000	-	30.000		5	25.53
	30.000	-	40.000		4	34.04
	40.000	-	50.000		5	44.68
	50.000	-	60.000		5	55.32
	60.000	-	70.000		4	63.83
	70.000	-	80.000		4	72.34
	80.000	-	90.000		5	82.98
	90.000	-	100.000		4	91.49
	100.000	-	110.000		4	100.00

Стандартный отчет показывает, что в интервал времени ожидания в первой очереди от 0 до 5 минут попала 1 заявка, от 5 до 35 минут – 5 заявок, от 35 до 65 – 6 заявок и т.д. В очереди длиной от 0 до 10 единиц находилось 3 заявки, от 10 до 20 – 4 заявки, от 20 до 30 – 5 заявок и т.д.

Для построения гистограммы необходимо выбрать после выполнения программы пункт меню Window/Simulation Window/Table Window. Далее в открывшемся диалоговом окне задать имя таблицы (в данном примере TAB1 или TAB2). Результаты представлены на рис.1.1, 1.2.

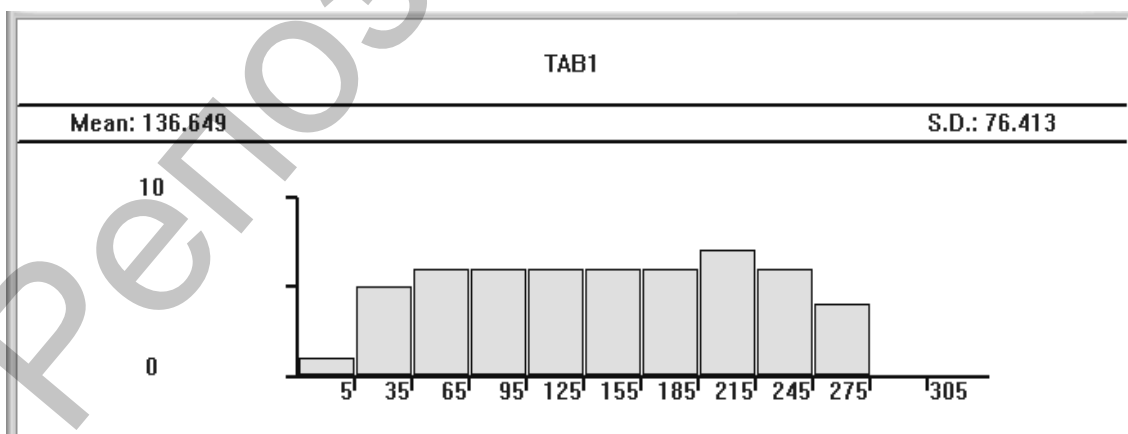


Рис. 1.1. Гистограмма распределения длительности пребывания заявки в очереди к первому каналу.

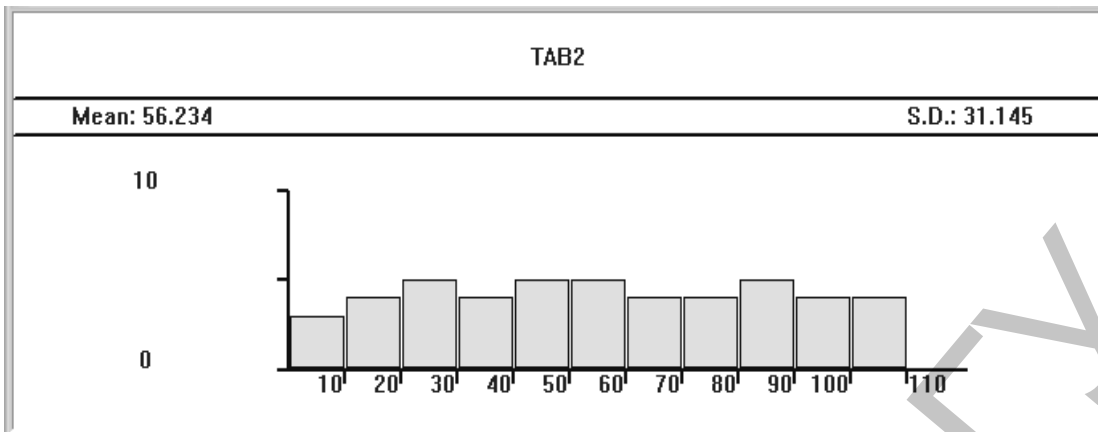


Рис. 1.2. Гистограмма распределения длины очереди ко второму каналу

1.10. Моделирование последовательности значений случайных величин с заданным законом распределения

Моделирование последовательности значений случайных величин с заданным законом распределения реализуется на основе использования базовой случайной величины, имеющей равномерное распределение в интервале (0,1). В GPSS имеются встроенные датчики случайных чисел, генерирующие целые случайные числа, равномерно распределенные от 0 до 999. Для получения случайного числа путем обращения к такому генератору достаточно записать RN с номером генератора, например RN1. Полученное число необходимо привести к отрезку (0,1) делением на 1000.

Так как встроенные генераторы основаны на конгруэнтном методе получения случайных чисел, то при каждом запуске они выдают одну и ту же последовательность чисел. Команда RMULT позволяет изменить эту последовательность путем изменения базы генератора. Формат команды:

RMULT A,B,C,D,E,F,G

Операнды <A>, , <C>, <D>, <E>, <F>, <G> задают начальные множители для соответствующих по номеру (1–7) генераторов случайных чисел. Например, RMULT 333,3 устанавливает начальные состояния множителей генераторов случайных чисел с номерами 1 и 3.

В GPSS имеется три типа переменных: арифметические переменные (целые), арифметические переменные с «плавающей точкой» (действительные) и булевские переменные. Переменные определяются перед началом моделирования с помощью следующих операторов описания:

Имя VARIABLE выражение (для целой переменной);

Имя FVARIABLE выражение (для действительной переменной);

Имя BVARIABLE выражение (для булевской переменной).

Здесь имя – имя переменной, используемое для ссылок на нее, а выражение – арифметическое или логическое (для булевской переменной) выражение, определяющее переменную. Арифметическое выражение представляет собой комбинацию операндов, в качестве которых могут выступать константы, функции, знаки арифметических операций и круглых скобок. Знаком операции умножения в GPSS является символ #.

Например, генерацию 1000 значений равномерно распределенной на интервале [5, 15] случайной величины можно выполнить так (рис. 1.3):

```
ZN FVARIABLE 10#(RN1/1000)+5 ; определение СВ
TAB1 TABLE V$ZN 5,1,12 ; определение таблицы для гистограммы СВ
GENERATE V$ZN ; генерация значений СВ
TABULATE TAB1 ; табулирование таблицы значений СВ
TERMINATE 1
START 1000
```

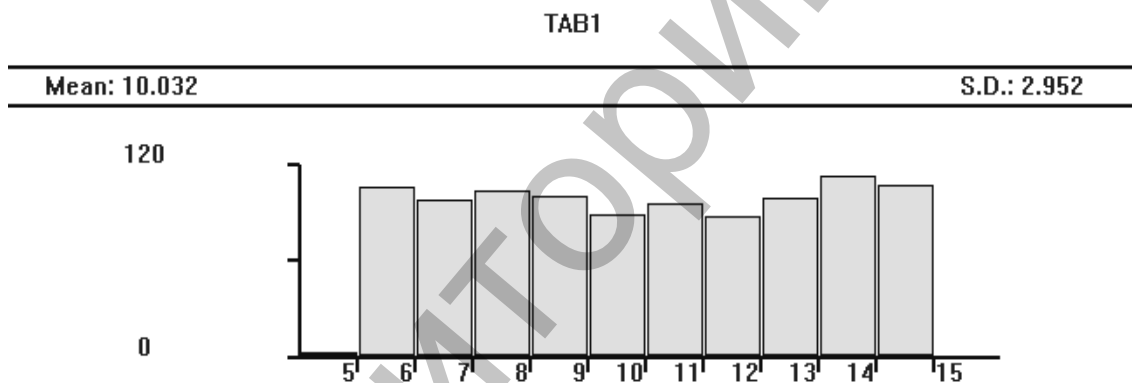


Рис. 1.3. Гистограмма равномерно распределенной СВ.

Для получения значений случайной величины Y с экспоненциальным (рис. 1.4) законом можно воспользоваться соотношением

$y = -\frac{1}{\lambda} \ln \gamma_i$, полученным на основе метода обратной функции. Для

$\lambda = 1$:

```
ZN FVARIABLE (-1)#LOG((1+RN1)/1000)
TAB1 TABLE V$ZN 0.5,0.5,12
GENERATE V$ZN
TABULATE TAB1
TERMINATE 1
START 1000
```

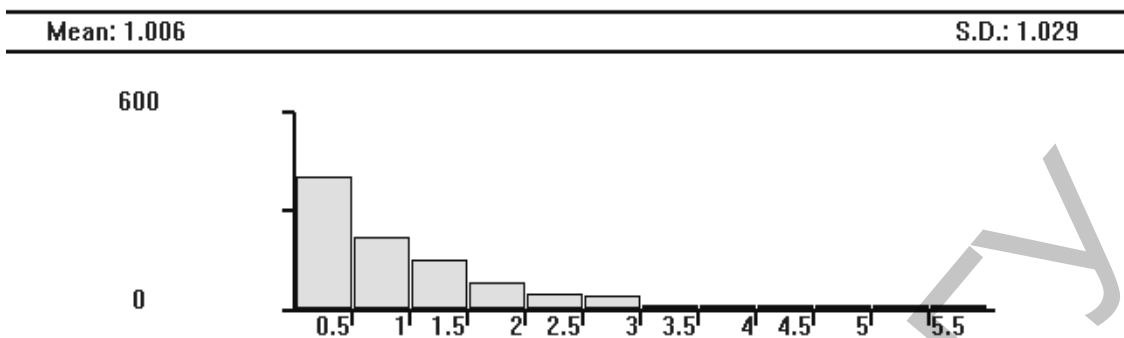


Рис. 1.4. Гистограмма СВ с экспоненциальным законом распределения.

Такой способ является достаточно трудоемким, так как требует обращения к математическим функциям, для вычисления которых необходимы десятки машинных операций. Другим возможным способом является использование вычислительных объектов GPSS типа функция.

Функции используются для вычисления величин, заданных табличными зависимостями. Существует пять типов функций (n – количество точек, задающих функцию): 1) непрерывные числовые значения C_n ; 2) дискретные числовые значения D_n ; 3) перечень числовых значений L_n ; 4) дискретные значения атрибутов E_n ; 5) перечень значений атрибутов M_n .

Первые два типа функции являются основными.

Каждая функция определяется перед началом моделирования с помощью оператора определения FUNCTION, имеющего следующий формат:

Имя FUNCTION A,B

Здесь имя – имя функции, используемое для ссылок на нее; A – стандартный числовой атрибут, являющийся аргументом функции. При использовании непрерывной функции для генерирования случайных чисел ее аргументом должен быть один из генераторов случайных чисел RN_j ; B – тип функции и число точек таблицы, определяющей функцию.

Тип непрерывной числовой функции кодируется буквой C. Так, например, в определении непрерывной числовой функции, таблица которой содержит 24 точки, поле должно иметь значение C24.

Особенностью использования встроенных генераторов случайных чисел RN_j в качестве аргументов функций является то, что их значения в этом контексте интерпретируются как дробные числа от 0 до 0,999999.

Таблица с координатами точек функции располагается в строках, следующих непосредственно за оператором FUNCTION. Эти строки не должны иметь поля нумерации. Каждая точка таблицы задается парой X_i (значение аргумента) и Y_i (значение функции), отделяемых друг от дру-

га запятой. Пары координат отделяются друг от друга символом «/» и располагаются на произвольном количестве строк. Последовательность значений аргумента X_i должна быть строго возрастающей.

При использовании функции в поле блоков GENERATE и ADVANCE вычисление интервала поступления или времени задержки производится путем умножения операнда <A> на вычисленное значение функции. Отсюда следует, что функция, используемая для генерирования случайных чисел с показательным (экспоненциальным) распределением $\lambda = 1$, должна описывать зависимость $y = -\ln(1-x)$, представленную в табличном виде.

Использование функций для получения случайных чисел с заданным законом распределения дает хотя и менее точный результат за счет погрешностей аппроксимации, но зато с меньшими вычислительными затратами (несколько машинных операций на выполнение линейной интерполяции). По сути, этот вариант реализации получения последовательности значений случайных величин соответствует методу ступенчатой аппроксимации функции плотности распределения вероятностей.

Функции всех типов имеют единственный СЧА с названием FN, значением которого является вычисленное значение функции. Вычисление функции производится при входе транзакта в блок, содержащий ссылку на СЧА FN с именем функции.

Пример 19. Выполнить моделирование случайных величин со следующими параметрами:

- 1) экспоненциальное распределение для $\lambda=1$;
- 2) экспоненциальное распределение для $\lambda=0,01$;
- 3) стандартное нормальное распределение со средним, равным 0 и стандартным квадратичным отклонением СКО=1;
- 4) нормальное распределение со средним, равным 30 и СКО=4;
- 5) дискретное распределение такое, чтобы СВ получала значения 1, 4, 5 с относительной частотой 0,40; 0,10; 0,50.

Решение заданий 1 и 2:

EXPON FUNCTION RN1,C24 ; задаем функцию

0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915

.7,1.2/.75,1.38/.8,1.6/.84,1.85/.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

TAB1 TABLE FN\$EXPON 0.5,0.5,12 ; таблица для СВ с распределением (1)

TAB2 TABLE V\$EXPV 50,50,15 ; таблица для СВ с распределением (2)

EXPV FVARIABLE 100#FN\$EXPON

GENERATE V\$EXPV

* для генерации транзактов можно использовать вместо предыдущих двух строк:

```
* GENERATE 100, FN$EXPON
TABULATE TAB1
TABULATE TAB2
TERMINATE 1
START 1000
```

На рисунке 1.5 показано распределение случайной величины для интенсивности потока событий $\lambda=0,01$. Операнд <A> получен путем преобразования интенсивности в соответствующее ей среднее время между последовательными поступлениями заявок: 1 заявка в 100 единиц времени.

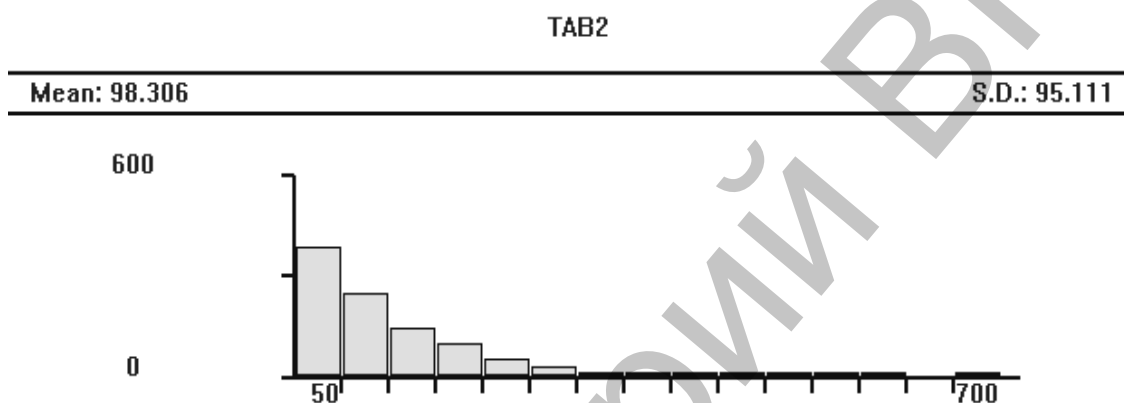


Рис.1.5. Гистограмма СВ с экспоненциальным законом распределения, $\lambda=0,01$.

Решение заданий 3 и 4: для того чтобы с помощью таблицы задать нормальное распределение случайной величины, используется 25 точек для обеспечения достаточной точности аппроксимации. Данная таблица задает случайную величину Z с математическим ожиданием равным 0, и стандартным квадратичным отклонением равным 1. Для моделирования нормальной случайной величины X с другими значениями математического ожидания и СКО необходимо произвести вычисления по формуле:

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

где m_x – математическое ожидание СВ, σ_x – СКО.

```
NORM FUNCTION RN2,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
```

```
TAB3 TABLE FN$NORM 0,0.5,10 ; таблица для СВ с распределением (3)
```

```
TAB4 TABLE V$NORMV 20,1,25 ; таблица для СВ с распределением (4)
```

```
NORMV FVARIABLE 30+4#FN$NORM
```

```
GENERATE V$NORMV
```

```
* или GENERATE (30+4#FN$NORM) ; другой способ получения СВ
```

```
TABULATE TAB3
```

```
TABULATE TAB4
```

```
TERMINATE 1
```

```
START 1000
```

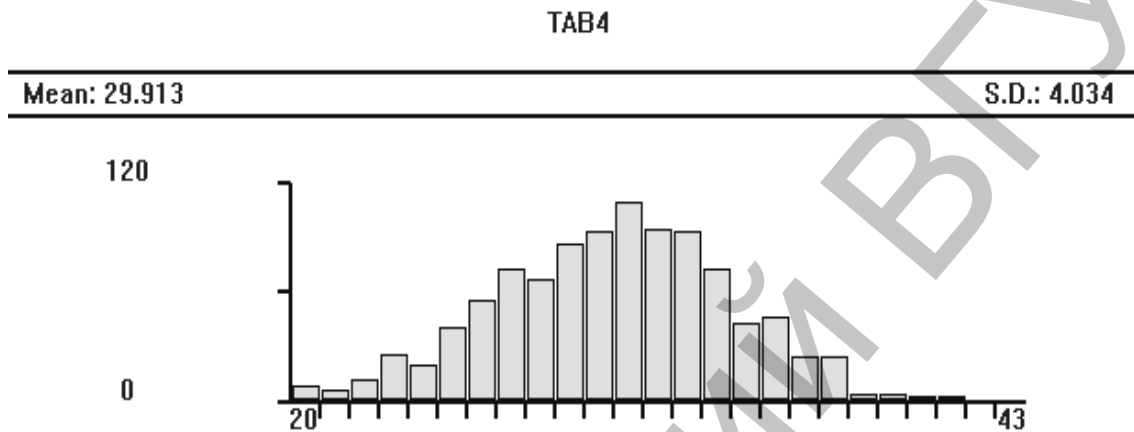


Рис. 1.6. Гистограмма СВ с нормальным распределением, $M=30$, $СКО=4$.

Решение задания 5:

```
DSV FUNCTION RN3,D3
```

```
0.4,1/.5,4/1,5
```

```
TAB5 TABLE FN$DSV,0,1,7
```

```
GENERATE FN$DSV
```

```
TABULATE TAB5
```

```
TERMINATE 1
```

```
START 1000
```

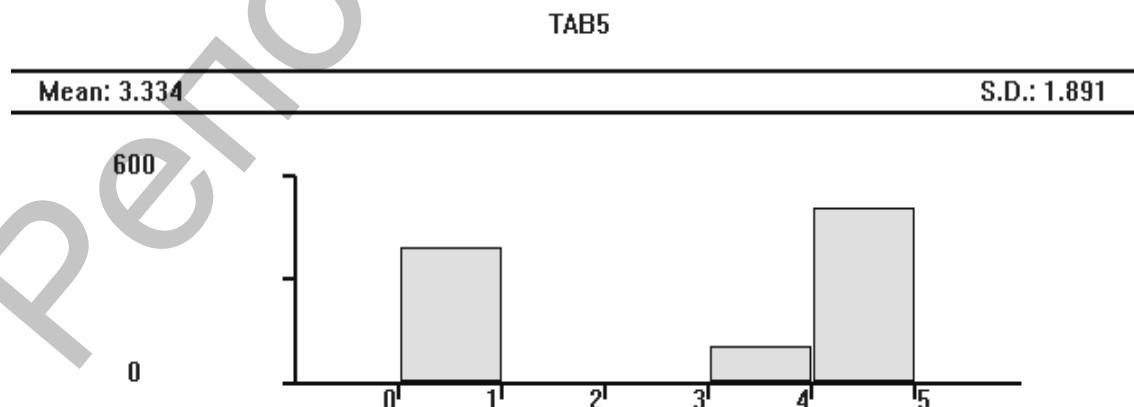


Рис. 1.7. Гистограмма СВ с дискретным распределением.

Пример 20. Интервал поступления заявок в СМО распределен равномерно в диапазоне 8–12 минут. Поток обработки заявок является

пуассоновским с интенсивностью $\lambda=0,1$. Построить частотное распределение, получить оценки математического ожидания и среднеквадратического отклонения для времени пребывания заявки в модели. Решить задачу двумя способами: моделируя время обработки заявки на основе метода обратной функции и метода ступенчатой аппроксимации.

Для решения задачи можно использовать стандартный числовой атрибут M1, который связан с каждым транзактом и хранит время пребывания транзакта в модели.

Решение методом обратной функции:

```
TABL TABLE M1,0,3,15
EXPV FVARIABLE (-10)#LOG((1+RN1)/1000)
GENERATE 10,2
ADVANCE V$EXPV
TABULATE TABL
TERMINATE 1
START 100
```

Среднее время пребывания заявки в модели 9,735 минут.

Решение с использованием таблицы с координатами точек функции (метод ступенчатой аппроксимации):

```
TABL TABLE M1,0,3,15
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/.84,1.85/.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 10,2
ADVANCE 10, FN$EXP
TABULATE TABL
TERMINATE 1
START 100
```

Среднее время пребывания заявки в модели 10,584 минут. Теоретическое значение математического ожидания равно 10. Оба полученных результата согласуются с теоретическим значением для заданной интенсивности потока λ .

1.11. Контрольные вопросы по теме

1. В чем заключается достоинство имитационного моделирования как метода исследования сложных систем?
2. В чем преимущество использования языка имитационного моделирования?
3. Что представляют собой объекты GPSS-модели?

4. Что такое транзакт? Какова его структура и функции?
5. Какие блоки языка GPSS предназначены для создания и уничтожения транзактов?
6. Какие блоки языка GPSS осуществляют модификацию параметров транзактов?
7. Каким образом можно получить поток заявок в GPSS-модели?
8. Из каких элементов состоит стандартный отчет?
9. Какие средства языка GPSS используются для имитации одноканальных устройств?
10. Какие блоки языка GPSS предназначены для управления устройствами?
11. Для чего предназначены таблицы в языке GPSS?
12. Какую информацию о таблицах можно получить из отчета?
13. Какие блоки языка GPSS предназначены для управления накопителями?
14. При каких условиях на входе прибора или МКУ образуется очередь? Чем определяется факт наличия очереди?
15. Чем отличается захват прибора от занятия прибора?
16. Как моделируются параллельно работающие приборы?
17. Какие средства GPSS используются для моделирования МКУ?
18. В чем заключается сходство и различие между прибором и МКУ?
19. Как реализуется передача управления (изменение последовательности передвижения транзактов в модели) в GPSS-модели?
20. Каким образом в языке GPSS можно управлять ключами?
21. Какие блоки языка GPSS входят в группу синхронизации транзактов?
22. Какая случайная величина называется базовой?
23. Какие существуют распределения случайных величин?
24. Как определяются в языке GPSS функции для разыгрывания значений дискретной и непрерывной случайной величины?
25. Какие средства языка GPSS используются для имитации экспоненциально или нормально распределенных случайных величин?
26. Какие статистические объекты используются в GPSS?
27. Какие блоки языка GPSS предназначены для получения статистических результатов моделирования системы?
28. Каким образом можно строить гистограммы в языке GPSS?
29. Какие средства имеются в языке GPSS для организации экспериментов с моделью?
30. Каковы особенности имитационного эксперимента на ЭВМ с точки зрения обработки результатов?

ЧАСТЬ 2. ПЛАНИРОВАНИЕ И ОБРАБОТКА ЭКСПЕРИМЕНТА С ПОМОЩЬЮ ПАКЕТА STATISTICA

Интегрированная система статистического анализа и обработки данных STATISTICA включает: электронные таблицы для исходных данных и вывода результатов статистического анализа; специализированные статистические модули, в которых собраны группы статистических процедур; графическую систему визуализации данных и результатов статистического анализа; систему подготовки отчетов; встроенный язык программирования, позволяющий расширить стандартные возможности системы.

2.1. Вычисление основных статистических характеристик

На начальном этапе обработки данных находят общие характеристики полученных показателей: среднее, дисперсию, размах вариации и т.д., определяют вид распределения. Вычисление описательных статистик производится при помощи модуля *Статистика > Основная статистика/Таблицы*. В диалоговом окне процедуры **Descriptive statistics** (описательные статистики) при помощи кнопки *Variables* следует выбрать переменные для анализа и щелкнуть по кнопке *Summary: Descriptive statistics*. В окне результатов будет представлена следующая информация:

- *Valid N* – объем выборки, по которой были рассчитаны характеристики;
- *Mean* – среднее значение по каждому показателю;
- *Minimum, maximum* – минимальное и максимальное значения показателя;
- *Std.Dev. (Standard Deviation)* – стандартное отклонение (среднеквадратическое отклонение).

Чтобы увеличить набор вычисляемых статистических характеристик, можно воспользоваться вкладкой *Advanced*. Наряду с уже рассмотренными, наиболее часто используются следующие характеристики:

- *Sum* – сумма;
- *Median* – медиана;
- *Variance* – дисперсия;
- *Standard error of mean* – стандартная ошибка среднего;
- *Conf.limits for means (95% confidence limits of mean)* – 95%-й доверительный интервал для среднего;
- *Lower&upper quartiles* – нижняя и верхняя квартили. Верхняя (нижняя) квартиль – это такое значение случайной величины, больше (меньше) которого по величине 25% случаев выборки;

- **Range** – размах (расстояние между наибольшим (maximum) и наименьшим (minimum) значениями признака);
- **Quartile range** – интерквартильный размах (расстояние между нижней и верхней квантилями);
- **Skewness** – асимметрия. Асимметрия характеризует степень смещения вариационного ряда относительно среднего значения по величине и направлению. В симметричной кривой коэффициент асимметрии равен нулю. Если правая ветвь кривой, начиная от вершины, больше левой (правосторонняя асимметрия), то коэффициент асимметрии больше нуля. Если левая ветвь кривой больше правой (левосторонняя асимметрия), то коэффициент асимметрии меньше нуля;
- **Std. err., Skewness (Standard error of Skewness)** – стандартная ошибка асимметрии;
- **Kurtosis** – эксцесс. Эксцесс характеризует степень концентрации случаев вокруг среднего значения и является своеобразной мерой крутости кривой. В кривой нормального распределения эксцесс равен нулю. Если эксцесс больше нуля, то кривая распределения характеризуется островершинностью, то есть является более крутой по сравнению с нормальной, а случаи более густо группируются вокруг среднего. При отрицательном эксцессе кривая является более плосковершинной, то есть более пологой по сравнению с нормальным распределением;
- **Std. err., Kurtosis (Standard error of Kurtosis)** – стандартная ошибка эксцесса.

Далее, нажав на кнопку *Summary*, можно получить таблицу с требуемыми показателями.

2.2. Построение таблицы, графиков частот и гистограммы

Для построения гистограмм и таблиц частот используется вкладка *Normality* окна *Descriptive statistics*. Число классов (интервалов) группировки данных устанавливается при помощи счетчика переключателя *Number of intervals*. Количество классов группировки можно приблизительно определить следующим образом: при количестве наблюдений 25–40 – 5–6 классов; при количестве наблюдений 40–60 – 6–8 классов; 60–100 – 7–10; 100–200 наблюдений – 8–12, более 200 наблюдений – 10–15 классов. Если сделать активным переключатель *Integer intervals (categories)*, то интервалы группировки будут представлять собой целые числа.

Рассмотрим возможности этой процедуры на примере. Выполним обработку статистических данных, характеризующих метеоусловия аэропортов в зимний период. В качестве исходных данных ис-

пользуются сведения по 30 аэропортам в различных городах, для которых известно среднее количество дней в году с твердыми осадками (снегом).

Архангельск	60	Кишинев	20	Курск	60
Мурманск	100	Самара	60	Новгород	80
Петрозаводск	80	Волгоград	20	Красноярск	60
Таллин	45	Тбилиси	10	Иркутск	60
Рига	45	Баку	6	Чита	35
Вильнюс	40	Ереван	10	Якутск	100
Минск	45	Омск	60	Хабаровск	40
Москва	60	Алма-Ата	20	Владивосток	20
Екатеринбург	80	Ташкент	15	Магадан	60
Киев	40	Новосибирск	80	Душанбе	8

Исходные данные сохранить в файле с именем *Airports.sta*.

После открытия файла *Airports.sta* и выбора процедуры **Descriptive statistics** на экране появится одноименное диалоговое окно. Далее следует выбрать переменные для анализа (переменная 2 – *Sneg*), результирующая таблица выводится на экран нажатием кнопки **Frequency tables** (таблица частот) на вкладке **Normality** (рис. 2.1).

Frequency table: Sneg (Airports)						
K-S d=,14619, p> .20; Lilliefors p<,10						
Category	Count	Cumulative Count	Percent of Valid	Cumul % of Valid	% of all Cases	Cumulative % of All
-20,0000 <x<=0,000000	0	0	0,00000	0,0000	0,00000	0,0000
0,000000 <x<=20,00000	9	9	30,00000	30,0000	30,00000	30,0000
20,00000 <x<=40,00000	4	13	13,33333	43,3333	13,33333	43,3333
40,00000 <x<=60,00000	11	24	36,66667	80,0000	36,66667	80,0000
60,00000 <x<=80,00000	4	28	13,33333	93,3333	13,33333	93,3333
80,00000 <x<=100,0000	2	30	6,66667	100,0000	6,66667	100,0000
Missing	0	30	0,00000		0,00000	100,0000

Рис. 2.1. Таблица частот (ряд распределения) для переменной *Sneg*.

В первом столбце таблицы задаются интервалы для переменной *Sneg*, причем последняя строка содержит пропущенные значения. Второй столбец содержит число попаданий переменной в интервалы; третий столбец – кумулятивное число попаданий; четвертый и шестой столбцы – частоты в процентах соответственно для имеющихся в наличии наблюдений и для всех наблюдений; пятый и седьмой столбцы – кумулятивные частоты в процентах соответственно для имеющихся в наличии и для всех наблюдений.

Для построения графиков частот и кумулятивных частот нужно выделить четвертый и пятый столбцы и в меню **Графики > Графики данных блока** выбрать пункт **Line Plot: Entire Columns** (рис. 2.2). Для построения частотной гистограммы требуется в окне описательной статистики нажать кнопку **Histograms** (рис. 2.3).

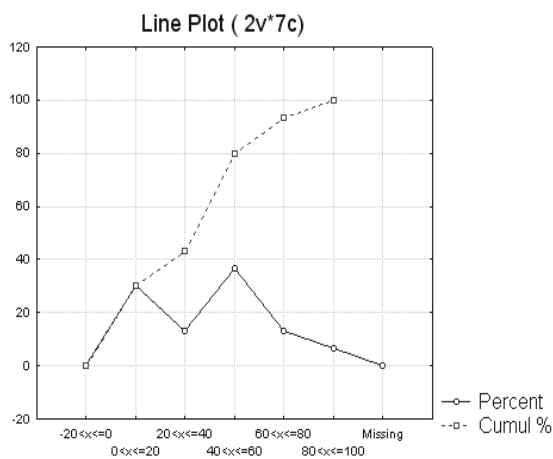


Рис. 2.2. Графики частот и кумулятивных частот для переменной *Sneg*.

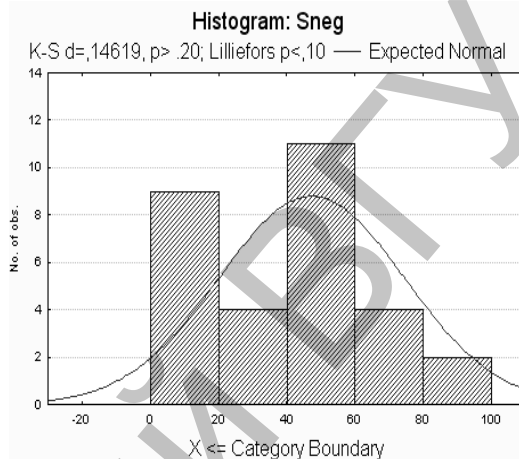


Рис. 2.3. Гистограмма для переменной *Sneg*.

Частоты в классах исследуемой совокупности изображены в виде столбцов, а сплошной линией показана нормальная функция распределения.

Для проверки близости полученного распределения к нормальному виду можно вывести значения критерия Колмогорова-Смирнова (*d*), Лилиефорса, вычислить статистику Шапиро-Уилкса. Для этого в группе опций **Distribution** необходимо установить флажок напротив соответствующих статистик. Значения статистик выводятся в области заголовка гистограммы и в заголовке таблицы частот.

О нормальности распределения также можно судить по графику на нормальной вероятностной бумаге. Его легко построить при помощи опции **Normal probability plots** вкладки **Prob.& Scatterplots**. Если точки достаточно хорошо ложатся на прямую, можно говорить о нормальном характере распределения переменной. В сомнительных случаях проверку на нормальность можно продолжить с использованием специальных статистических критериев.

Для визуализации описательных статистик можно построить диаграмму размаха, которая позволяет оценить симметричность распределения переменной (рис. 2.4). Это можно сделать при помощи кнопки **Box & Whisker plot for all variable** вкладки **Quick**, предварительно указав во вкладке **Options** нужные статистики:

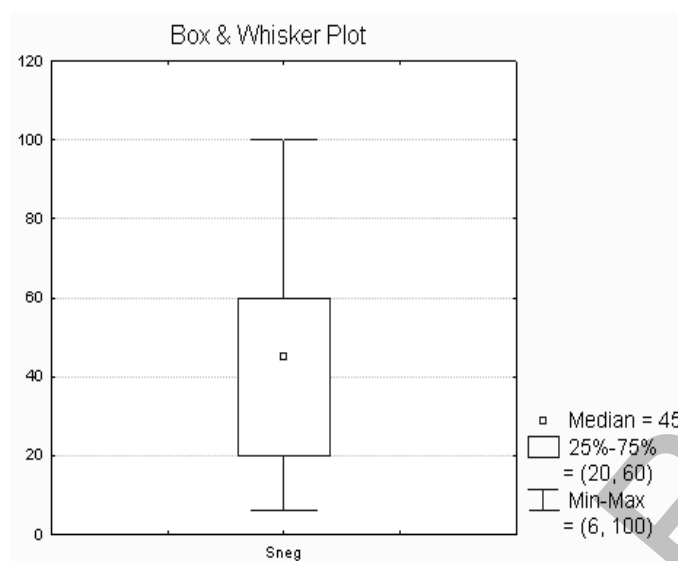


Рис. 2.4. Диаграмма размаха для переменной *Sneg*.

- *Median/Quart./Range* – Медиана / Квартили / Размах;
- *Mean/SE/SD* – Среднее / Ошибка среднего / Стандартное отклонение;
- *Mean/SD/1.96SD* – Среднее / Стандартное отклонение / Интервал 1,96* стандартного отклонения;
- *Mean/SE/1.96*SE* – Среднее / Ошибка среднего / Интервал 1,96 * ошибки среднего.

2.3. Планирование и обработка результатов эксперимента

Планирование и обработку результатов эксперимента рассмотрим на примере. Пусть имеются результаты проведения полного факторного эксперимента (ПФЭ) типа 2^3 , где число факторов $k=3$, число уровней $p=2$, число опытов $N=8$, число повторных наблюдений в каждом опыте $n=5$. Требуется по данным эксперимента построить многофакторную модель.

Матрица планирования:

№ опыта	Уровни факторов			Результаты параллельных экспериментов
	X_1	X_2	X_3	
1	+	+	–	7,8; 8,5; 7,7; 7,6; 8,0
2	–	+	–	1,8; 2,5; 2,0; 1,8; 1,6
3	+	–	–	4,4; 4,8; 5,3; 4,9; 5,2
4	–	–	–	4,3; 4,2; 5,0; 4,9; 4,6
5	+	+	+	9,7; 10,4; 11,4; 10,9; 10,9
6	–	+	+	4,2; 4,4; 4,5; 4,0; 3,8
7	+	–	+	3,7; 3,4; 4,0; 3,6; 4,1
8	–	–	+	4,1; 5,1; 4,8; 5,1; 4,5

Для перехода к процедуре формирования матрицы эксперимента необходимо в меню STATISTICA выбрать раздел **Статистика > Индустриальная статистика & Сигма шесть > Экспериментальный проект**.

В стартовом меню раздела **Design&Analysis of Experiments** (экспериментальный проект) выбирается процедура **2**(K-p) standard designs**.

В появившемся окне во вкладке **Design Experiment** необходимо определить количество **входных факторов / блоков / опытов** соответственно условиям эксперимента (рис. 2.5). Если эксперимент выполняется в несколько этапов, применяется разделение опытов на серии (блоки) случайным образом.

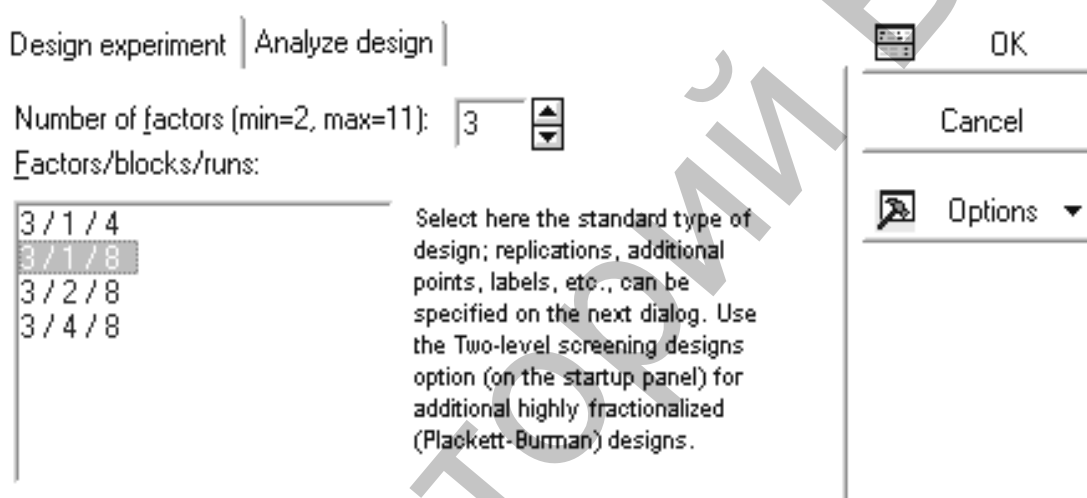


Рис. 2.5. Определение условий эксперимента.

После нажатия **OK** будет получено окно **Design of an Experiment with Two-Level Factors**, в котором необходимо выполнить следующие установки:

- во вкладке **Display design** в рамке **Denote factors** определяется, как в матрице планирования будут обозначены факторы: числами, буквами или именами; в рамке **Order of runs** – порядок размещения опытов в матрице: по порядку, случайным образом или по блокам; в рамке **Show (in Spreadsheet)** – как будут обозначены уровни факторов в матрице: числами, мин/макс значением или символами (рис. 2.6);
- во вкладке **Add to design** нужно указать количество дополнительных прогонов для каждой точки плана (4) и количество дополнительных пустых столбцов (1), в которые после проведения эксперимента будут внесены значения отклика (рис. 2.7).

Для получения матрицы эксперимента необходимо нажать кнопку **Summary**. Полученная матрица сохраняется в файле в формате *.sta. Для того чтобы сохранить только матрицу планирования, а не

всю рабочую книгу, необходимо в меню *Инструменты* > *Опции* во вкладке *Менеджер вывода* установить флажок в позиции *Поместить все результаты в индивидуальные окна*.

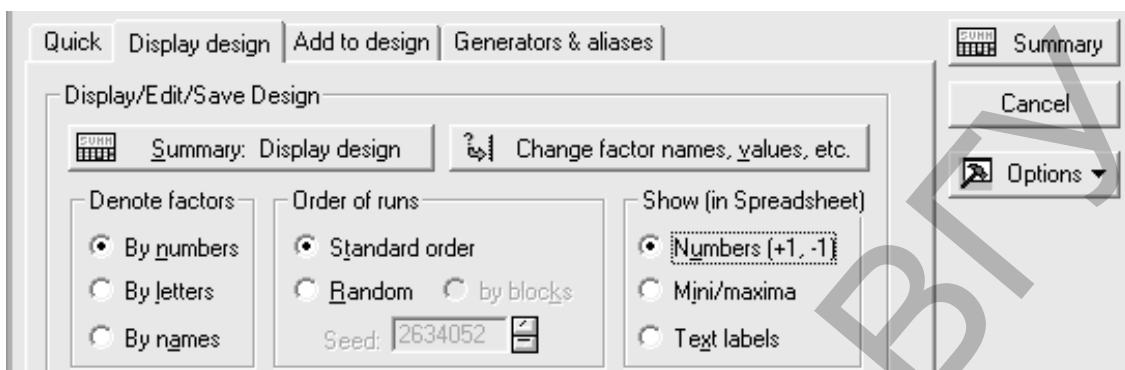


Рис. 2.6. Установки вкладки *Display design*.

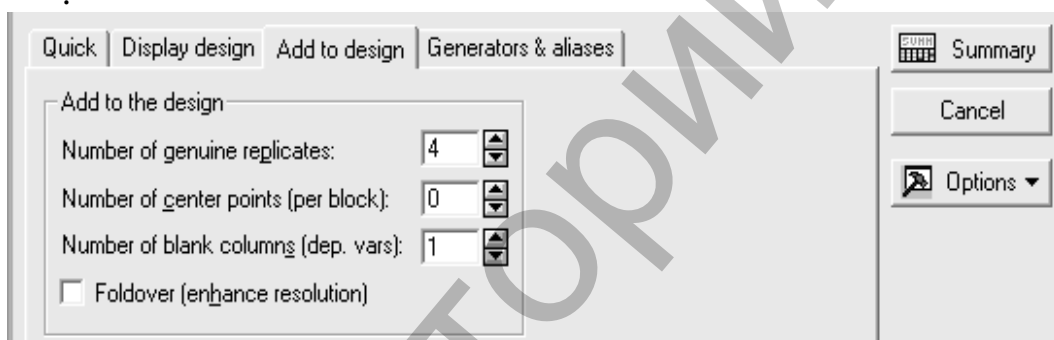


Рис. 2.7. Установки вкладки *Add to design*.

После разработки матрицы эксперимента в свободный столбец вводятся значения отклика. Далее можно воспользоваться специальными процедурами обработки данных эксперимента пакета STATISTICA. Для вызова этих процедур нужно перейти во вкладку *Analyze design*, с помощью кнопки *Variables* установить зависимые (отклик) и независимые (факторы) переменные. Далее необходимо нажать **OK** для получения окна *Analysis of an Experiment with Two-Level Factors*. Во вкладке *Model* полученного окна указывается, надо ли включать взаимодействие двух и более факторов (*3-way interactions*).

Итогами обработки данных эксперимента являются ряд параметров, наиболее интересными из которых являются значения коэффициентов линейной многофакторной модели и результаты дисперсионного анализа.

Для получения значений коэффициентов модели, их ошибок и уровней значимости используется кнопка *Regression coefficients* во вкладке *ANOVA/Effects*. Результаты обработки данных примера представлены на рис. 2.8.

Regr. Coefficients; Var.: DV_1; R-sqr=,98084; Adj: ,97664 (Design1) 2**(3-0) design; MS Residual=,157125 DV: DV_1						
Factor	Regressn Coeff.	Std.Err.	t(32)	p	-95,% Cnf.Limt	+95,% Cnf.Limt
Mean/Interc.	5,337500	0,062675	85,16188	0,000000	5,209836	5,465164
(1)F1	1,477500	0,062675	23,57408	0,000000	1,349836	1,605164
(2)F2	0,837500	0,062675	13,36264	0,000000	0,709836	0,965164
(3)F3	0,492500	0,062675	7,85803	0,000000	0,364836	0,620164
1 by 2	1,637500	0,062675	26,12695	0,000000	1,509836	1,765164
1 by 3	-0,097500	0,062675	-1,55565	0,129627	-0,225164	0,030164
2 by 3	0,752500	0,062675	12,00643	0,000000	0,624836	0,880164
1*2*3	0,222500	0,062675	3,55007	0,001216	0,094836	0,350164

Рис. 2.8. Значения коэффициентов модели.

Коэффициенты с уровнем значимости $p > 0,5$ включать в модель нецелесообразно, поэтому искомая модель будет иметь вид:

$$Y = 5,3375 + 1,4775x_1 + 0,8375x_2 + 0,4925x_3 + 1,6375x_1x_2 + 0,7525x_2x_3 + 0,2225x_1x_2x_3$$

Коэффициент детерминации R^2 показывает, в какой степени вариация отклика определяется вариацией факторов, включенных в модель. Близкое к единице значение коэффициента детерминации $R^2 = 0,98084$ говорит о хорошем качестве полученной модели.

Для получения результатов дисперсионного анализа используется кнопка *ANOVA tables* во вкладке *ANOVA/Effects* (рис. 2.9):

Полученные данные позволяют осуществить оценку степени влияния контролируемых факторов x_1 , x_2 , x_3 и их взаимодействий на отклик. Дисперсия факторов задана в столбце *SS*, остаточная дисперсия, описывающая влияние неконтролируемых факторов, задана в строке *Error*, общая дисперсия – в строке *Total SS*.

Выразив в процентах долю каждой дисперсии в общей дисперсии, можно получить значения, оценивающие степень влияния факторов на отклик модели:

Factor	SS	df	MS	F	p
(1)F1	87,3203	1	87,3203	555,7375	0,000000
(2)F2	28,0562	1	28,0562	178,5601	0,000000
(3)F3	9,7022	1	9,7022	61,7486	0,000000
1 by 2	107,2563	1	107,2563	682,6173	0,000000
1 by 3	0,3802	1	0,3802	2,4200	0,129627
2 by 3	22,6503	1	22,6503	144,1543	0,000000
1*2*3	1,9802	1	1,9802	12,6030	0,001216
Error	5,0280	32	0,1571		
Total SS	262,3737	39			

Рис. 2.9. Результаты дисперсионного анализа.

Фактор	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$	неконтр. ф.
Доля дисперсии (%)	33,3	10,7	3,7	40,9	0,1	8,6	0,8	1,9

Проведем анализ остатков. Остатками называют разности между фактическими значениями зависимой переменной и предсказанными (вычисленными по модели). Если модель адекватна, остатки представляют собой случайную величину с нормальным распределением и нулевым математическим ожиданием. Для визуальной оценки остатков используется гистограмма остатков и график остатков на нормальной вероятностной бумаге, которые можно получить, нажав кнопки *Histogram of residuals* и *Normal plot* на вкладке *Residual plots* (рис. 2.10).

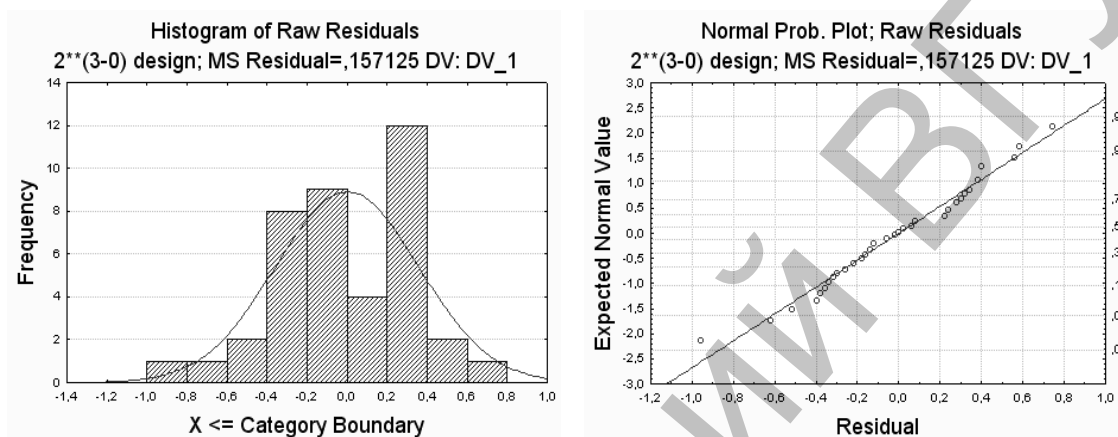


Рис. 2.10. Гистограмма и график остатков на вероятностной бумаге.

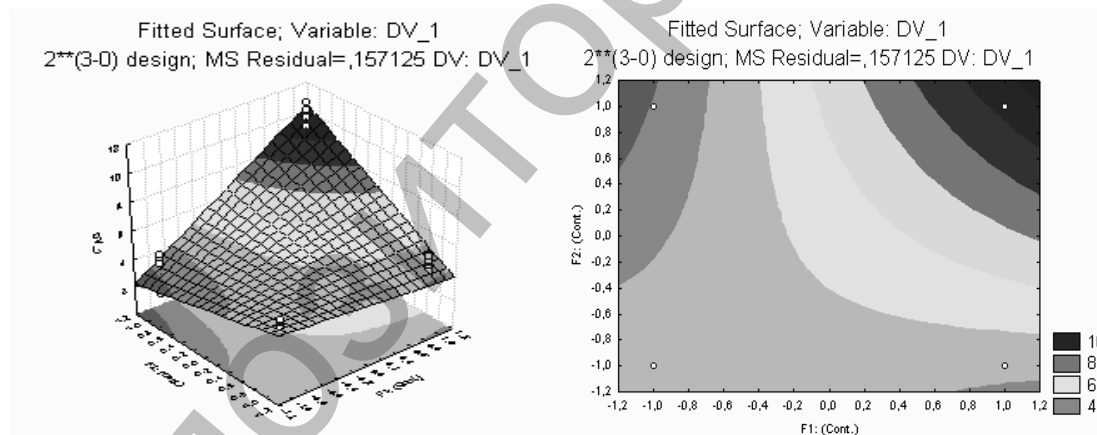


Рис. 2.11. Графическое отображение результатов эксперимента.

Вкладка *Prediction&profiling* позволяет визуализировать результаты эксперимента. Нажав *Surface plot*, можно получить изображение поверхности отклика для исследуемой зависимости. Если количество входных факторов более двух, то STATISTICA предложит определить только те два фактора, которые необходимо использовать при построении графика. Для получения поверхности отклика в виде изолиний, т.е. линий одинакового уровня выходного параметра, необходимо нажать *Contour plot* (рис. 2.11).

Если данная модель неадекватна, то, чтобы получить модель второго порядка, к плану проведенных опытов, который называется «ядром» эксперимента, добавляется несколько специальным образом

расположенных точек. Функция отклика формируется в виде полинома второй степени. Для корректной оценки всех коэффициентов полинома второй степени необходимо, чтобы в плане эксперимента каждый фактор имел не менее трех уровней варьирования. Для разработки таких планов используют процедуру центрального композиционного планирования *Central composite designs*.

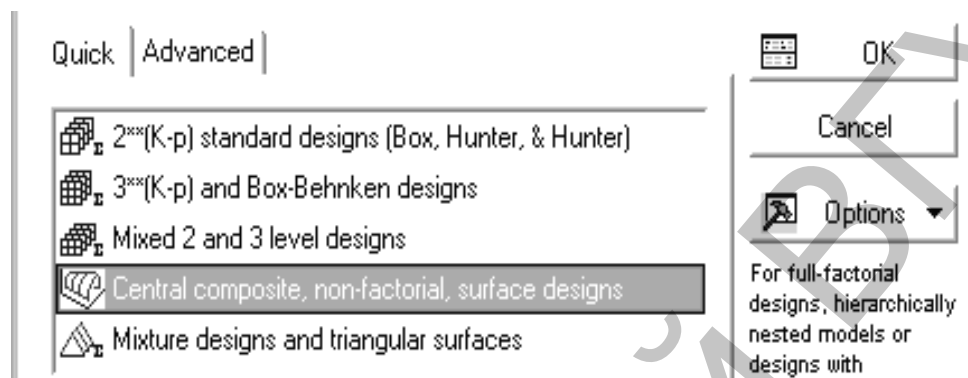


Рис. 2.12. Установки вкладки *Design & Analysis of Experiments*.

Например, для построения ортогонального центрального композиционного плана второго порядка нужно во вкладке *Add to design* ввести в план дополнительный столбец для значений отклика, добавить одну центральную точку; во вкладке *Design characteristic* указать расчет α звездных точек для ортогональности плана (*Compute / use alpha for orthogonality*).

Анализ экспериментальных данных проводится аналогично рассмотренному в примере. Необходимо определить зависимую и независимые переменные, найти коэффициенты регрессионной модели второго порядка и уровни их значимости. Исключив статистически незначимые коэффициенты, получить уравнение регрессионной модели и изображение поверхности отклика исследуемой зависимости. Провести дисперсионный анализ, оценить степень влияния факторов на отклик и адекватность модели.

2.4. Автоматическое создание отчета

При статистической обработке данных в рабочем окне системы STATISTICA одновременно может находиться множество открытых документов: файл с исходными данными, таблицы с результатами анализа, графики. Подобная ситуация типична для большинства видов статистического анализа. Если же расчеты содержат много промежуточных результатов, которые генерируют набор таблиц и графиков, то это приводит к тому, что все рабочее пространство будет заполнено, а это крайне неудобно.

В связи с этим в системе STATISTICA реализована возможность автоматического создания файла регистрации результатов анализа. Такой файл называется автоотчетом. Для создания автоотчета необходимо

вызвать пункт меню *File > Output Manager*, в окне *Options* во вкладке *Менеджер вывода* установить метку в окне *Также послать окно отчета*, а во вкладке *Отчеты* задать параметры автоматического создания отчета. Все таблицы и графики, появляющиеся на экране, будут автоматически выводиться в файл с автоотчетом, который открывается в отдельном окне на рабочем пространстве системы STATISTICA.

2.5. Контрольные вопросы по теме

1. Каковы цели стратегического и тактического планирования компьютерных экспериментов?
2. Какие виды факторов бывают в имитационном эксперименте?
3. Что называется полным факторным экспериментом? Объясните процедуру формирования плана эксперимента.
4. Каким образом формируется факторный план 2^k ? Как учитываются эффекты взаимодействия факторов в функции отклика?
5. Что такое «ядро» эксперимента? Какие точки нужно добавить к плану эксперимента, чтобы можно было построить модель второго порядка?
6. Что такое поверхность отклика? Как построить регрессионную модель?
7. Как оценить адекватность полученной модели?
8. Какие методы математической статистики используются для анализа результатов имитационного моделирования систем?
9. Какие средства пакета STATISTICA используются для обработки данных эксперимента?
10. Какими возможностями располагает система STATISTICA? Какие статистические модули имеются в пакете?
11. Назовите выборочные числовые характеристики. Как получить описательную статистику с помощью пакета STATISTICA?
12. Какие типы графиков имеются в пакете STATISTICA?
13. Как построить матрицу планирования полного факторного эксперимента в пакете STATISTICA? Что представляют собой зависимые и независимые переменные?
14. Назовите порядок операций, которые необходимо выполнить для построения уравнения регрессии.
15. С помощью каких статистических критериев оценивается значимость коэффициентов уравнения регрессии?
16. Что представляет собой коэффициент детерминации? Для чего он используется?
17. Как можно оценить степень влияния факторов на отклик модели с помощью дисперсионного анализа?
18. Что называют остатками? Как оценить адекватность модели с помощью анализа остатков?
19. Что представляет собой процедура центрального композиционного планирования?
20. Как создать автоотчет? Какую информацию содержит отчет?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Основы моделирования и первичная обработка данных. – М.: «Финансы и статистика», 1983. – 471 с.
2. Боровиков В.П., Боровиков И.П. Statistica. Статистический анализ и обработка данных в среде Windows. – М.: Информационно-издательский дом «Филинь», 1997. – 608 с.
3. Лобач В.И., Кирлица В.П., Малюгин В.И., Сталевская С.Н. Имитационное и статистическое моделирование: практикум для студентов мат. и экон. спец. – Мн.: БГУ, 2004. – 189 с.
4. Руководство по GPSS/PC. Minuteman Software, перевод с английского под ред. Якимова И.М. – Казань, 1997. – 320 с.
5. Советов Б.Я., Яковлев С.А. Моделирование систем: учебник для вузов. – М.: Высшая школа, 2007. – 343 с.
6. Томашевский В., Жданова Е. Имитационное моделирование в среде GPSS. Серия «Факультет». – М.: Бестселлер, 2003. – 416 с.
7. Тюрин Ю.П., Макаров А.А. Анализ данных на компьютере. – М.: ИНФРА-М, Финансы и статистика, 1995. – 384 с.
8. Черненко В.М. Разработка САПР. Книга 9 – Имитационное моделирование: практическое пособие / под ред. А.В. Петрова. – М.: Высшая школа, 1990. – 112 с.
9. Шеннон Р.Дж. Имитационное моделирование систем – искусство и наука. – М.: Мир, 1978. – 418 с.
10. Шрайбер Т.Дж. Моделирование на GPSS. – М.: Машиностроение, 1980. – 593 с.
11. <http://www.gpss.ru>. Сайт, посвященный системе моделирования GPSS.
12. <http://www.statsoft.ru>. Статистический портал, содержащий информацию о пакете STATISTICA.

ПРИЛОЖЕНИЯ

Приложение 1

Блоки GPSS

<i>Оператор</i>	<i>Поля</i>	<i>Описание</i>
Блоки для работы с транзактами		
GENERATE	A,B,C, D,E	Создание транзактов. А – среднее время между транзактами; В – половина диапазона или модификатор функции; С – начальная задержка; D – число транзактов, которое должно быть создано; E – приоритет транзакта
TERMINATE	A	Уничтожение транзакта. А – на сколько уменьшается счетчик, определяющий число проходов
SPLIT	A,B,C	Создание копий транзакта. А – число создаваемых копий; В – блок, куда направляются копии; С – номер параметра транзакта, в котором производится нумерация получаемых транзактов
ASSEMBLE	A	Объединение транзактов, являющихся членами одного ансамбля А – число объединяемых транзактов, далее из блока выходит 1 транзакт
GATHER	A	Аналогично SPLIT, только из блока выходят все транзакты
MATCH	A	Синхронизация двух транзактов одного ансамбля. А – метка сопряженного блока, блок применяется парами
ADVANCE	A,B	Задержка транзакта. А – среднее время задержки; В – половина диапазона или модификатор функции
ASSIGN	A,B,C	Изменение параметра транзакта. А – номер параметра или имя, В – значение, С – функция
PRIORITY	A,B	Установить приоритет. А – новое значение приоритета
Блоки, изменяющие маршруты движения транзактов		
GATE X	A,B	Выбор направления в зависимости от состояния устройств Условие X: NU (прибор свободен), U (прибор занят), NI (прибор не прерван), I (прибор прерван), FV (прибор доступен), FNV (прибор не доступен), SE (накопитель пуст), SNE (накопитель не пуст), SF (накопитель полон), SNF (накопитель не полон), SV (накопитель доступен), SNV (накопитель не доступен), LS (логический ключ включен), LR (логический ключ выключен)
TEST X	A,B,C	Сравнение двух операндов и выбор направления движения Условие X: E =, G >, GE >=, L <, LE <=, NE !=; A,B – операнды, С – альтернативный путь при невыполнении условия (если отсутствует, то транзакт блокируется).
TRANSFER	A,B,C, D	Передача транзактов (логически, статистически, условно, безусловно). А – определяет режим: BOTH (один альтернативный адрес); PICK (случайным образом); ALL (много альтернатив, В – первый адрес, С – последний адрес, D содержит индексную константу); если А пустое, то безусловная передача в В, дробь в А задает вероятность выбора

<i>Оператор</i>	<i>Поля</i>	<i>Описание</i>
Блоки для работы с приборами		
SEIZE	A	Занять прибор. A – имя или номер прибора
RELEASE	A	Освободить прибор. A – имя или номер прибора
PREEMPT	A,B,C, D,E	Захватить прибор. A – имя или номер прибора; если в B находится PR, то захват в режиме приоритета; C определяет блок, куда передается прерванный транзакт, при этом прерванный транзакт продолжает претендовать на данное устройство; D определяет параметр; если в E находится RE, то прерванный транзакт не претендует на устройство
RETURN	A	Окончание захвата. A – имя или номер прибора.
FUNAVAIL	A,B,C, D,E,F, G,H	Прибор недоступен. A – имя или номер прибора
FAVAIL	A	Прибор доступен. A – имя или номер прибора
Блоки для работы с очередями		
QUEUE	A,B	Занять очередь. A – имя очереди (номер), B – число занимаемых мест (по умолчанию 1)
DEPART	A,B	Освободить очередь. A – имя очереди (номер), B – число освобождаемых мест (по умолчанию 1)
Блоки для работы с накопителями (многоканальными устройствами)		
ENTER	A,B	Занять накопитель. A – имя накопителя (номер), B – число занимаемых ячеек (по умолчанию 1)
LEAVE	A,B	Освободить накопитель. A – имя накопителя (номер), B – число освобождаемых ячеек (по умолчанию 1)
SUNAVAIL	A	Накопитель недоступен. A – имя накопителя (номер)
SAVAIL	A	Накопитель доступен. A – имя накопителя (номер)
Блок для сбора статистических данных		
TABULATE	A,B	Сбор статистики. A – имя таблицы (номер), B – число единиц, добавляемых к интервалу наблюдения
Блоки для изменения состояния логических ключей, матриц, величин		
LOGIC X	A	Изменение состояния логического ключа. A – имя ключа. Условие X: S – включен, R – выключен, I – инвертирован
SAVEVALUE	A,B	A – имя сохраняемой величины, B – значение
MSAVEVALUE	A,B,C,D	A – имя матрицы, B – строка, C – столбец, D – значение
Операторы описания данных		
Имя VARIABLE	A	Определить целую переменную, A – выражение-инициализатор
Имя FVARIABLE	A	Определить вещественную переменную, A – выражение-инициализатор
Имя BVARIABLE	A	Определить булеву переменную, A – выражение-инициализатор
Имя MATRIX	A,B,C	Определить матрицу, A – не используется, B – количество строк, C – количество столбцов.
INITIAL	A,B	Инициализировать. A – инициализируемый элемент, B – выражение-инициализатор
Имя STORAGE	A	Определить накопитель, A – емкость накопителя

<i>Оператор</i>	<i>Поля</i>	<i>Описание</i>
Имя TABLE	A,B,C, D	Определить таблицу (гистограмму), A – табулируемое значение, B – нижний предел, C – шаг, D – число интервалов
Имя QTABLE	A,B,C, D	Определить таблицу (гистограмму) для очереди, табулируется автоматически время ожидания транзакта в очереди, A – имя очереди, далее аналогично с TABLE

Приложение 2

Стандартные числовые атрибуты

<i>Обозначение</i>	<i>Описание</i>
Общесистемные	
AC1	Значение абсолютного системного времени (время после выполнения последнего CLEAR)
C1	Значение относительного системного времени (время после выполнения последнего RESET)
TG1	Значение счетчика числа оставшихся проходов
RN	Случайное значение 0-999 генератора с заданным номером (RN1, RN2, RN3, ...)
Транзакты	
P	Значение параметра
PR	Целое значение определяет приоритет транзакта
M1	Время пребывания транзакта в модели
MP	Время с момента входа в блок
Приборы	
F	Состояние прибора (1 – занят, 0 – свободен)
FC	Счетчик числа занятий
FI	1 – если прибор захвачен, 0 – в противном случае
FR	Коэффициент использования прибора
FT	Среднее время обслуживания транзакта
FV	1 – если прибор доступен, 0 – в противном случае
Накопители	
R	Число свободных ячеек накопителя
S	Число используемых ячеек накопителя
SA	Среднее содержимое накопителя
SC	Целое число – счетчик входов
SE	1 – если накопитель пуст, 0 – в противном случае
SF	1 – если накопитель полон, 0 – в противном случае
SR	Коэффициент использования накопителя
SM	Максимальное содержимое накопителя
ST	Среднее время нахождения транзакта в накопителе
SV	1 – если накопитель доступен, 0 – в противном случае
Очереди	
Q	Текущая длина очереди
QA	Средняя длина очереди
QC	Счетчик общего числа входов
QM	Максимальная длина очереди
QT	Среднее время пребывания транзакта в очереди
QX	Среднее время пребывания транзакта в очереди без учета нулевых входов
QZ	Число элементов, не задержанных в очереди

Репозиторий ВГУ