

Инструментальные средства моделирования

Цель работы – получить навыки работы с языком имитационного моделирования GPSS.

Основные сведения

1.1. Основные понятия моделирования систем

Моделирование – это метод исследования сложных систем, основанный на том, что рассматриваемая система заменяется на модель и проводится исследование модели с целью получения информации об изучаемой системе. Под моделью исследуемой системы понимается некоторая другая система, которая ведет себя с точки зрения целей исследования аналогично поведению системы. Обычно модель проще и доступнее для исследования, чем система, что позволяет упростить ее изучение. Среди различных видов моделирования, применяемых для изучения сложных систем, большая роль отводится имитационному моделированию. **Имитационной** называется модель, представленная в виде программы на специальном языке и воспроизводящая все элементарные явления, составляющие функционирование исследуемой системы во времени с сохранением их логической структуры и последовательности.

Под *системой S* понимается выделение в соответствии с некоторым правилом объединения элементов любого рода, образующих связанное целое. Все, что оказывает воздействие на систему, и на что система оказывает воздействие, называется *внешней средой*. В общем случае состав элементов системы переменный. Одни элементы находятся в системе постоянно, другие – появляются и покидают систему (временные элементы). Все атрибуты элементов и системы в целом можно разделить на два типа: переменные и постоянные. Переменными являются атрибуты, значение которых остается неизменным в рассматриваемом периоде времени.

Совокупность конкретных значений всех переменных атрибутов элементов и системы в целом в некоторый момент времени существования системы определяет *состояние системы $z(t)$* . Системы в соответствии с различными признаками могут быть классифицированы следующим образом:

- динамические – статические;
- дискретные – непрерывные – комбинированные;
- стохастические (вероятностные) – детерминированные.

Система является *динамической*, если ее состояние меняется с изменением времени, в противном случае система является *статической*. Если состояние системы, т.е. значение ее атрибутов, изменяется непрерывно, то она называется *непрерывной* системой, а если значения изменяются в дискретные моменты времени, то система называется *дискретной*. Существуют такие системы, у которых часть атрибутов, описывающих

состояние системы, меняется непрерывно, а часть дискретно. Эти системы называются непрерывно–дискретными или *комбинированными*. Система называется *стохастической*, если при одних и тех же начальных условиях результаты функционирования системы будут различаться, иначе система называется *детерминированной*.

Функционирование системы может рассматриваться и описываться как взаимодействие событий, действий или процессов, происходящих в системе.

Под *событием* понимается всякое изменение состояния системы под воздействием внешней среды и сложившихся в системе условий. Событие рассматривается как мгновенное изменение состояния системы.

Под *действием* понимается пребывание элемента системы в некотором состоянии. Переход элемента в данное состояние (начало действия) и выход из этого состояния (окончание действия) определяется условиями, сложившимися в системе.

Упорядоченная во времени логически взаимосвязанная последовательность событий, выделенная в соответствии с некоторым признаком, называется *процессом*. Таким образом, процесс – это более агрегативное понятие, чем событие и действие.

1.2. Основные принципы языка GPSS

GPSS (General Purpose Simulating System – общецелевая система моделирования) является *языком имитационного моделирования*, используемым для построения моделей и проведения моделирования на ЭВМ. Модели на GPSS компактны, часто состоят из меньшего числа операторов, чем такие же модели, написанные на процедурных языках (например, на Паскале или Си). Это объясняется тем, что в GPSS встроено максимально возможное число логических программ, необходимых для моделирующих систем. В него также входят специальные средства для описания динамического поведения систем, меняющихся во времени, причем изменение состояний происходит в дискретные моменты времени. GPSS очень удобен при программировании, поскольку интерпретатор GPSS (здесь и далее интерпретатором называется моделирующая часть системы GPSS) многие функции выполняет автоматически.

Рассмотрим систему, состоящую из одного человека, выполняющего обслуживание определенного вида. Этот человек может быть кассиром, продающим билеты на станции, контролером в универсальном магазине, парикмахером в парикмахерской с единственным креслом. "Клиенты" приходят к такому "обслуживающему прибору" в случайные моменты времени, ждут своей очереди на обслуживание (если есть необходимость), их обслуживают по принципу "первый пришел – первым обслужен". После этого они уходят. Схематично эта ситуация показана на рис.1.1, где прямоугольник

– это обслуживающий прибор, а кружок внутри него – заявка, находящаяся на обслуживании.

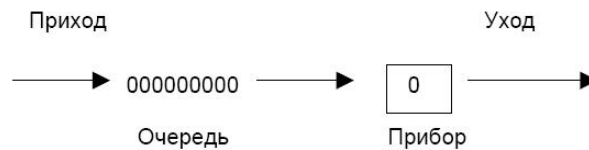


Рис.1.1. Простая система массового обслуживания

Для дальнейшего рассмотрения системы введем следующие определения:

ОЧЕРЕДЬ – это группа заявок, ожидающих обслуживания.

МОДЕЛЬНОЕ ВРЕМЯ – это промежуток времени между началом моделирования и его завершением.

СИСТЕМА МАССОВОГО ОБСЛУЖИВАНИЯ (СМО) – это система, состоящая из обслуживающего прибора, заявки, находящейся на обслуживании, и ожидающих обслуживания заявок.

Простая система массового обслуживания, изображенная на рис.1.1, характеризуется двумя независимыми случайными переменными:

ИНТЕРВАЛ ПРИБЫТИЯ заявок – это интервал времени между последовательными моментами прибытия заявок в систему.

ВРЕМЯ ОБСЛУЖИВАНИЯ – это время, требуемое прибору для выполнения обслуживания.

СИСТЕМНАЯ ВЕЛИЧИНА – это величина, зависящая от значения первых двух независимых случайных переменных. Ниже перечислены некоторые из этих случайных величин, являющихся также случайными переменными:

1. Число заявок, прибывших на обслуживание за заданный промежуток времени.
2. Число заявок, которые попали на обслуживание сразу же по прибытии.
3. Среднее время пребывания заявок в очереди.
4. Средняя длина очереди.
5. Максимальная длина очереди.
6. Нагрузка прибора, являющаяся функцией времени, которое потрачено прибором на обслуживание в течение заданного промежутка времени.

Распределения этих системных величин и являются предметом исследования. Следует заметить, что разработку логической схемы модели на ЭВМ, которая будет имитировать систему обслуживания с одним прибором и очередью, нужно вести при следующих условиях:

1. Случайные переменные ИНТЕРВАЛ ПРИБЫТИЯ и ВРЕМЯ ОБСЛУЖИВАНИЯ являются равномерно распределенными и принимают только целые значения.

2. Все прибывающие заявки должны быть обслужены независимо от длины очереди.

3. Вначале моделирования система "пуста", т.е. нет очереди, и обслуживающий прибор свободен.

4. Моделирование продолжается до тех пор, пока не будет достигнуто значение модельного времени, заданное для этой модели в качестве одного из входных данных.

Элементы процедуры решения

События

При моделировании систем массового обслуживания совершаются некоторые СОБЫТИЯ. Все события в системе должны быть каким-либо образом зафиксированы, и должно быть учтено их воздействие на текущее состояние системы. Кроме того, необходимо определить, как нужно корректировать состояние системы в связи с воздействием на нее этих событий. События разделяются на две категории:

1. Основное событие – это такое событие, время возникновения которого можно запланировать заранее. Это, например, приход заявки, начало обслуживания и завершение обслуживания.

2. Вспомогательное событие – это событие, время возникновения которого невозможно запланировать заранее. Эти события возникают тогда же, когда и основные, но являются зависимыми, возникающими как следствие основных.

Таймер модельного времени

Так как работа модели связана с последовательным возникновением событий, то вполне естественно использовать понятие ТАЙМЕР МОДЕЛЬНОГО ВРЕМЕНИ в качестве одного из элементов модели системы. Для этого вводят специальную переменную и используют ее для фиксации текущего времени работы модели. Опишем теперь некоторые специфические свойства таймера модельного времени.

Когда начинается моделирование, таймер модельного времени обычно устанавливают на нулевое значение. Разработчик сам решает вопрос о том, какое значение реального времени принять за точку отсчета. Например, началу отсчета может соответствовать 8 ч утра первого моделируемого дня. Разработчик также должен решить вопрос о выборе величины единицы времени. Единицей времени может быть 1 с, 5 с, 1 мин, 20 мин или 1 ч. Когда единица времени выбрана, все значения времени, получаемые при

моделировании или входящие в модель, должны быть выражены через эту единицу.

На практике значения модельного времени должны быть достаточно малыми по сравнению с реальными промежутками времени, протекающими в моделируемой системе. В данной системе обычно выбирают единицу времени, равную 1 мин. Если при моделировании некоторой системы при текущем значении модельного времени ее состояние изменилось, то нужно увеличить значение таймера. Чтобы определить, на какую величину должно быть увеличено значение таймера, используют один из двух методов:

1. *Концепция фиксированного приращения значений таймера.*

При таком подходе увеличивают значение таймера ровно на одну единицу времени. Затем нужно проверить состояния системы и определить те из запланированных событий, которые должны произойти при новом значении таймера. Если таковые имеются, то необходимо выполнить операции, реализующие соответствующие события, снова изменить значение таймера на одну единицу времени и т.д. Если проверка покажет, что для нового значения таймера не запланировано ни одного события, то произойдет передвижение таймера непосредственно к следующему значению.

2. *Концепция переменного приращения значений таймера.*

В этом случае условием, вызывающим приращение таймера, является наступление времени "близкого события". Близкое событие – это то событие, возникновение которого запланировано на момент времени, равный следующему ближайшему значению таймера модельного времени. Колебания приращения таймера от случая к случаю объясняют выражение " переменное приращение времени".

Следует заметить, что выгоднее использовать концепцию переменного приращения значений таймера, т.к. при нем можно избежать обработки в промежуточные моменты времени, когда не планируется выполнение никаких событий. В рассматриваемой задаче с одним обслуживающим прибором и очередью будет использована именно эта концепция. И один из способов определения времени ближайшего события заключается в составлении списка моментов времени, в которые запланированы выполнения различных основных событий. Когда приходит время наращивания значения таймера, в этом списке должно быть найдено ближайшее событие. (Если список моментов времени отсортирован в порядке возрастания, то никакого поиска не требуется. Таймер просто установится в нужное значение в соответствии с первой позицией списка). Затем таймеру должно быть придано его значение, и управление должно быть передано в ту часть модели, где обслуживается событие, о котором идет речь.

Завершение моделирования

Обычно после какого-то момента времени наступает необходимость прекратить моделирование. Например, нужно предотвратить приход новых заявок в систему, но обслуживание надо продолжать до освобождения системы. Одним из способов является введение в модель основного псевдособытия, называемого ЗАВЕРШЕНИЕМ МОДЕЛИРОВАНИЯ. Тогда одной из функций модели будет планирование этого события. Момент времени, наступление которого должно вызвать остановку моделирования, задается обычно в виде числа. Таким образом, в процессе моделирования нужно проверять, является ли событие "завершение моделирования" следующим событием. Если "да", то в таймере устанавливается значение времени конца моделирования, а управление передается процедуре, которая отрабатывает завершение моделирования.

Алгоритмизация модели

Мы рассмотрели различные элементы, из которых формируется модель системы обслуживания с одним прибором и очередью. Последним шагом является сбор этих элементов в единую общую модель. В общем виде логическую схему модели можно представить в виде блок-схемы на рис. 1.2.

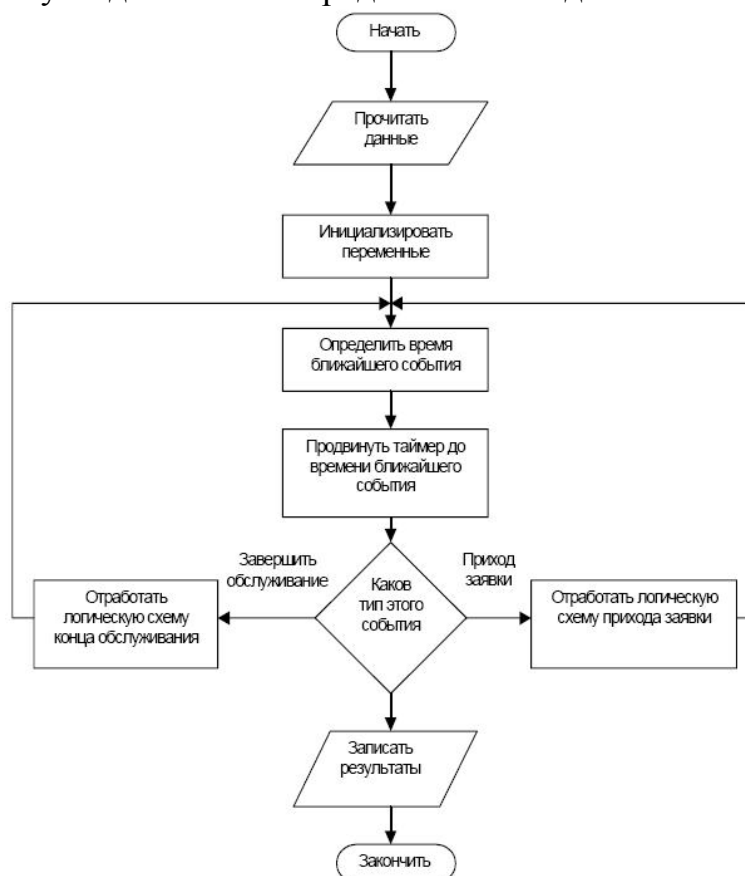


Рис. 1.2. Блок-схема GPSS-модели

Теперь наша задача состоит в создании машинной модели на ЭВМ, которая позволит изучить поведение системы в течение времени моделирования. Иначе говоря, нужно реализовать эту блок–схему на ЭВМ, используя блоки и операторы языка GPSS.

1.3. Основные концепции моделирования на GPSS

Джеффи Гордон в 1961 г разработал язык моделирования GPSS (General Purpose Simulating System – моделирующая система общего назначения).

В основе каждого языка или системы моделирования лежит определенная (базовая) концепция структуризации для описания реального объекта. Во многом это определяет класс моделируемых объектов, которые могут быть описаны системой моделирования.

Содержание базовой концепции структуризации языка GPSS

1) Это блочно–ориентированная концепция, разработанная с ориентацией на описание СМО (СМО – система массового обслуживания).

Структура моделируемого процесса изображается в виде потока, проходящего через ОУ (обслуживающие устройства), очереди, ключи и другие элементы СМО.

Модель имеет блочную структуру. Моделируемый процесс представляется как поток заявок в системе обслуживания. Блоки интерпретируются как ОУ. Заявки (транзакты) конкурируют между собой за место в ОУ, образуют очереди перед ОУ, если они заняты. Дуги на блок–схеме – потенциальные потоки заявок между ОУ. Существуют истоки и стоки этих заявок. В этом случае блок–схема модели описывает маршруты движения заявок в системе.

Следовательно, в рамках GPSS есть специальные средства, которые являются аналогами элементов систем массового обслуживания, таких как обслуживающие устройства, заявки, очереди.

Однако, GPSS является гибкой языковой средой, поэтому позволяет моделировать не только СМО, но и другие системы (например, склад, распределение ресурсов и др.)

2) GPSS – система дискретного типа.

Система GPSS ориентирована на класс объектов, процесс функционирования которых можно представить в виде множества состояний и правил перехода из одного состояния в другое, определяемых в дискретной пространственно–временной области.

GPSS позволяет описывать процессы с дискретными событиями.

Для регистрации изменений времени во времени существует таймер модельного времени (он может быть не только целочисленным). Механизм задания модельного времени: пособытийный, с переменным шагом. Изменения в реальной системе приводят к появлению событий. Событие –

изменение состояния любого элемента системы. В системе происходят события:

- поступление заявки,
- постановка заявки в очередь,
- начало обслуживания,
- конец обслуживания и др.

3) GPSS относится к классу процессно–(транзактно)–ориентированных систем моделирования.

GPSS является способом алгоритмизации дискретных динамических систем. Ориентирована на описание параллельных процессов в динамической системе. Примеры моделируемых объектов: транспортные объекты, склады, производственные системы, магазины, торговые объекты, сети ЭВМ, системы передачи сообщений. Алгоритмическая схема может быть использована для оформления сложных формальных схем. Формальные модели таких объектов: СМО и стохастические сети, автоматы, сети Петри, агрегаты и др.

Функциональная структура GPSS

рассматривается на двух уровнях.

1 уровень определяется комбинацией основных *функциональных объектов* таких, как

- *устройства*
- *памяти*
- *ключи (логические переключатели)*
- *очереди*
- *транзакты;*

2 уровень – *блок–схема модели*, составленная из типовых блоков, между которыми перемещаются транзакты.

1 уровень

Аппаратно–ориентированные объекты:

– **Транзакты** являются абстрактными подвижными элементами, которые являются аналогами различных объектов реального мира (сообщения, транспортные средства, люди, детали т.д.) Это динамические функциональные элементы GPSS, которые отражают реальные заявки на обслуживание.

Транзакты двигаются по модели, появляются в ней с той же интенсивностью, что и реальные заявки. Транзакты могут создаваться и уничтожаться. Перемещаясь между блоками модели в соответствии с логикой моделирования, транзакты вызывают (и испытывают) различные действия:

- возможны их задержки в некоторых точках модели (связанные с обслуживанием, ожиданием в очереди),

- изменение маршрутов и направления движения,
- создание копии транзактов и др.

С каждым транзактом связан упорядоченный набор параметров – *атрибутов*.

При генерации транзактов резервируются 12 параметров. Обычно первые 12 параметров являются постоянными. В их набор входит:

- № транзакта,
- № блока, в котором транзакт находится в данный момент,
- № следующего блока,
- время перехода в следующий блок,
- приоритет, характеризующий очередность обработки транзактов в определенных ситуациях,
- и др.

Далее при программировании можно присвоить транзакту набор специфичных параметров, выражающих свойства или характеристики моделируемых объектов (вес, скорость, цвет, время обработки и т.п.).

Некоторые примеры возможных аналогий между транзактами и элементами реальных систем представлены в табл. 1.1.

Таблица 1.1.

Системы	Элементы систем, символизируемые транзактами
Большой универсальный магазин	Покупатель
Автомобильное шоссе	Автомобиль
Радиомастерская	Радиоприемник
Склад	Заявка
Парикмахерская	Клиент

Таким образом, перемещение транзакта от блока к блоку в модели аналогично, например, передвижению клиента в парикмахерской от одной стадии к другой.

В самом начале моделирования в GPSS–модели нет ни одного транзакта. В процессе моделирования транзакты входят в модель в определенные моменты времени в соответствии с теми логическими потребностями, которые возникают в моделируемой системе. Подобным же образом транзакты покидают модель. В общем случае в модели существует большое число транзактов, но в один момент времени движется только один.

Если транзакт начал свое движение, он перемещается от блока к блоку по пути, предписанному блок–схемой. Такое продвижение транзакта продолжается до тех пор, пока не произойдет одно из следующих возможных событий:

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

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

3) транзакт "пытается" войти в следующий блок в соответствии с блок-схемой, но блок "отказывается" принять его.

Если возникло одно из описанных условий, то транзакт остается на месте, и начинается перемещение в модели другого транзакта. Таким образом, выполнение моделирования в системе продолжается.

Модель на GPSS состоит из одного или нескольких независимых сегментов. В процессе моделирования активным является тот из сегментов, в котором находится перемещающийся в настоящий момент транзакт. Когда он блокируется, начинает двигаться следующий транзакт, и может быть так, что этот следующий транзакт принадлежит другому сегменту модели. Таким образом, происходит переключение активности между сегментами.

– **Устройства** моделируют объекты, в которых может происходить обработка транзактов, что связано с затратами времени. Устройства являются аналогами каналов СМО (каждое устройство в данный момент времени может быть занят лишь одним транзактом). Устройство может быть прервано. В GPSS существует возможность проверки состояния устройства.

– **Памяти** – предназначены для моделирования объектов, обладающих ёмкостью. Аналогия с многоканальными СМО – память может обслуживать одновременно несколько транзактов. При этом транзакт занимает определённую часть памяти.

– **Логические переключатели** – принимают значение включено/выключено, позволяют изменять пути следования транзактов в модели.

Статистические объекты GPSS

используются тогда, когда надо собирать статистику:

- **Очереди.** В процессе движения транзакты могут задерживаться в определенных точках модели. Если необходимо собирать информацию о длине очереди транзактов и времени задержки транзактов используют соответствующие статистические объекты.

- **Таблицы.** Таблицы обрабатывают статистическую информацию, строят гистограмму распределений по любой переменной.

Вычислительные объекты GPSS

- *матрицы*
- *функции*
- *переменные различных типов и т.п.*

2 уровень

Модель на языке моделирования GPSS имеет наглядное графическое представление в виде **блок–схемы**.

Блоки – операционные объекты GPSS. Каждый блок имеет стандартное обозначение. Последовательность блоков – это есть последовательность операторов на языке GPSS. Любую модель на языке GPSS можно представить в виде совокупности блоков, между которыми перемещаются транзакты, они имеют вход–выход, в блоках реализуются все действия, связанные с обслуживанием транзакта (создание и уничтожение транзактов, изменение параметров транзакта, управление потоками транзактов, и т.д.). Блоки выполняются только в результате входа в них перемещающихся транзактов.

GPSS является системой интерпретирующего типа с собственным языком.

Существуют два особых блока: GENERATE, имеющий только выход, через него транзакты входят в модель, и блок TERMINATE, имеющий только вход – удаляет транзакты из модели. Любой процесс на языке моделирования GPSS имеет вид:

GENERATE
(Блоки модели)
TERMINATE

Описание параллельных процессов на языке GPSS представляет несколько таких цепочек блоков, взаимодействующих через общие ресурсы.

Итак, модель системы на языке GPSS представляет сеть блоков (операторов языка). Каждый блок описывает определенный этап действий в системе. Линии соединения блоков показывают направления движения подвижных элементов (транзактов) через систему или описывают некоторую последовательность событий, происходящих в моделируемой системе.

1.4. Основные команды системы моделирования GPSS

GPSS World является объектно–ориентированным языком. В совокупность его основных объектов входят объекты «Модель», используемые для создания объектов «Процесс моделирования». Объекты «Процесс моделирования» в свою очередь используются для осуществления процесса моделирования и создания объектов «Отчет».

С помощью меню главного окна можно создавать и манипулировать объектами GPSS.

В ходе создания исходной модели работа в системе происходит в интерактивном режиме с использованием окна данных *Main Window* и окна блоков *Blocks Window*.

При создании объекта «Модель» можно использовать редактор (*Edit(Правка)*) и навигатор по ошибкам (команда *Search(Поиск)*).

Command/ Create Simulation (Команда/ Создать процесс моделирования) – происходит трансляция операторов модели в исполняемый объект «Процесс моделирования».

Command/ Retranslate (Команда/ Повторная трансляция) – происходит повторная трансляция.

После успешного создания объекта «Процесс моделирования» можно передавать ему любые операторы модели для изменения структуры или состояния моделирования. Способы интерактивной передачи операторов:

1. С помощью команд меню *Command* главного окна.
2. Ввод команд в специальное диалоговое окно (*Command/ Custom...(Команда/Ввести...)*)
3. Закрепление за несколькими функциональными клавишами собственных команд (*Edit/ Settings (Правка/ Настройки)*)

Сводку всех действий, осуществляемых в системе можно посмотреть в окне «*Journal*» («Журнал»).

Для наблюдения за процессом моделирования применяются различные динамические окна:

- Окно «*Blocks*» («Блоки») – оперативный обзор динамики блоков.
- Окно «*Plot*» («График») – оперативный обзор изменения значений системных числовых атрибутов (СЧА) и выражений с помощью графиков.
- Окно «*Expression*» («Выражения») – оперативный обзор значений выражений.
- Окно «*Facilities*» («Устройства») – оперативный обзор динамики изменения состояния устройств.
- Окно «*Table*» («Таблица») – оперативный обзор динамики изменения таблицы или Q-таблицы в виде гистограммы.
- Окно «*Queues*» («Очереди») – оперативный обзор динамики изменения очередей.
- Окно «*Storages*» («Памяти») – оперативный обзор динамики изменения содержимого памяти.
- и др.

Обычно после завершения процесса моделирования создается объект «Отчет». GPSS World обеспечивает автоматическое составление подробных статистических стандартных отчетов, которые, как правило, содержат достаточные для анализа выходные статистические данные о конечных состояниях всех традиционных объектов GPSS (приложение 1).

Основные команды системы моделирования

START A, B, C, D

Установка счетчика завершений и запуск процесса моделирования

A- счетчик завершения.

B- операнд вывода данных: NP – нет вывода данных, Null – вывод стандартного отчета.

C- не используется.

D- вывод списков: 1 – включение CEC/FEC в стандартный отчет, Null.

HALT

Прерывает процесс моделирования и сбрасывает очередь команд.

CLEAR A

Сбрасывает статистику процесса моделирования и удаляет из модели все транзакты.

A – ON или OFF. Если операнд A опущен, то подразумевается ON.

Если A равен OFF, то ячейки, логические ключи и элементы матриц остаются без изменений

RESET

Сбрасывает статистику процесса моделирования. Отмечает начало периода измерений.

REPORT A, B

Устанавливает имя файла отчета и запрашивает немедленный отчет

A – должен быть Null

B – NOW или Null. Для совместимости. Разницы между ними нет.

SHOW X

Вычисляет выражение X и отображает результат. Срочная команда.

EXIT A

Выход из системы моделирования.

A – код выхода: 1 – все объекты сохраняются, -1 – объекты не сохраняются, 0 или Null – все изменённые файлы вызовут появление окна сообщения, которое спрашивает сохранять или не сохранять файл.

Основные команды ручного моделирования:

STOP A, B, C

Устанавливает или снимает условие остановки.

A – номер транзакта

В – номер блока.

С – ON или OFF

STEP A

Устанавливает ограниченное количество входов транзактов в блоки, после которого процесс моделирования переходит в состояние останова.

А – количество входов в блок.

CONTINUE

Продолжает процесс моделирования

1.5. Основные блоки языка GPSS

Все блоки записываются с первой позиции строки, сначала идет имя блока, а затем, через запятую, параметры. В записи параметров не должно быть пробелов. Если какой-то параметр в блоке отсутствует (задается по умолчанию), то соответствующая ему запятая остается (если это не последний параметр). Если в первой позиции строки стоит символ *, то эта строка с комментарием.

GENERATE A,B,C,D,E,F

Создает транзакты через определенные интервалы времени.

А – средний интервал времени между появлениями транзактов.

В – 1) если число, то это половина поля, в котором равномерно распределено значение интервала между появлениями транзактов [А–В, А+В];

– 2) если функция, то для определения интервала значение А умножается на значение функции.

С – момент времени появления первого транзакта.

Д – предельное количество транзактов.

Е – величина приоритета транзакта.

Ф – число параметров у транзакта и их тип (РВ–байтовый целый, РН–полусловный целый, РР–полнословный целый, РЛ–с плавающей запятой).

TERMINATE A

Уничтожает транзакты из модели и уменьшает значение счетчика завершения на А единиц. Работа модели завершится, если счетчик завершения станет меньше или равен нулю. Если параметр А отсутствует, то блок просто уничтожает транзакты.

Примеры использования блока GENERATE:

1) GENERATE 5,3 – блок генерирует транзакты через 5 ± 3 единицы системного времени, т.е. интервалом прибытия является случайное число со средним значением, равным 5, и полем допуска, равным 6.

2) GENERATE 10 – генерируется поступление транзактов в систему через каждые 10 единиц системного времени.

3) GENERATE 3,3,10,5 – моментом прихода первого транзакта является значение 10. После этого интервалы времени прибытия находят из равномерного распределения 3 ± 3 . Однако только первые пять транзактов должны войти в модель через этот блок.

SEIZE A

Если прибор с именем A свободен, то транзакт занимает его (переводит в состояние "занято"), если нет, то ставится в очередь к нему. Именем прибора может быть числовой номер или последовательность от 3 до 5 символов.

RELEASE A

Транзакт освобождает прибор с именем A, т.е. переводит его в состояние "свободно".

ADVANCE A,B

Задерживает обработку транзакта данным процессом и планирует время начала следующего этапа обработки.

A – среднее время задержки.

B – имеет тот же смысл, что и для GENERATE.

Пример 1. Рассмотрим систему, состоящую из кассы и очереди к ней. Клиент обслуживается в кассе за время (3 ± 1) мин. Очередь пополняется за счет вновь прибывших через (5 ± 4) мин. Программа будет выглядеть следующим образом:

GENERATE 5,4	<i>;моделирование прихода клиентов в систему через (5 ± 4) мин;</i>
SEIZE KAS	<i>;если прибор с именем KAS находится в состоянии "свободно", то транзакт занимает его (клиент обслуживается в кассе); в противном случае встает в очередь;</i>
ADVANCE 3,1	<i>;клиент обслуживается в кассе (3 ± 1) мин;</i>
RELEASE KAS	<i>;обслуживание клиента в кассе закончено (транзакт освобождает прибор KAS, переводя его в состояние "свободно");</i>
TERMINATE 1	<i>;клиент покидает систему (транзакт уничтожается и счетчик завершения уменьшается на 1).</i>

START 100

В данной программе, кроме описанных выше, использовался еще блок START.

START A,B,C,D

A – начальное значение счетчика числа завершений.

B – признак подавления печати. Если задан параметр NP, то

стандартная печать в конце моделирования производиться не будет.

С – число завершений, через которые будет выдаваться промежуточная печать.

D – признак печати цепей. Печать цепей производится, если на месте D стоит 1.

QUEUE A

Собирает статистику о входе транзакта в очередь с именем A.

DEPART A

Собирает статистику о выходе транзакта из очереди с именем A.

Пример 2. Если в задаче из предыдущего примера необходима информация об очереди, то программу нужно дополнить.

```
GENERATE 5,4
QUEUE OH
SEIZE KAS
DEPART OH
ADVANCE 3,1
RELEASE KAS
TERMINATE 1
START 100
```

В данном примере группа операторов QUEUE–DEPART позволяет дать имя очереди к прибору и собирает статистику об этой очереди.

ENTER A,B

Если в многоканальном устройстве с именем A имеются свободные каналы, то транзакт занимает B каналов (по умолчанию 1). Если свободных каналов нет, то транзакт становится в очередь и его обработка процессом приостанавливается до освобождения каналов.

LEAVE A,B

Транзакт освобождает B каналов в многоканальном устройстве A. Прежде чем использовать многоканальное устройство, его нужно описать картой STORAGE.

A STORAGE B

Здесь A – имя многоканального устройства, B – максимальная емкость.

Пример 3. Пусть заявки поступают на обработку с ограниченным числом мест в очереди, равным 3. Если очередь заполнена, то заявки покидают систему.

```
LIM STORAGE 3
GENERATE 5,4
ENTER LIM
SEIZE PRB
LEAVE LIM
ADVANCE 3,2
RELEASE PRB
TERMINATE 1
START 100
```


Пример 4. Рассмотрим работу ЗУ емкостью 256К. В ЗУ с интервалом 3 ± 2 с загружаются файлы, занимающие 4К памяти. Через 10 ± 3 с становятся ненужными и освобождают память.

```
MEM STORAGE 256
GENERATE 3,2
ENTER MEM, 4
ADVANCE 10,3
LEAVE MEM, 4
TERMINATE 1
START 250
```

ASSIGN A,B,C

Присваивает (если A+ – увеличивает, A– – уменьшает) параметру транзакта (номер которого указан в A) значение параметра B. C – тип параметра транзакта. Параметр транзакта – это величина, соответствующая атрибуту элемента модели. У каждого транзакта свои параметры. Ссылка на параметр транзакта, который обрабатывается блоком, выполняется в виде группового имени P, за которым следует последовательный номер параметра и его тип. Например, P3H – третий полусловный параметр обрабатываемого транзакта.

Примеры использования блока ASSIGN:

- 1) ASSIGN 3,2,PH – третьему полусловному параметру присвоить 2.
- 2) ASSIGN 1–,1,PB – значение первого байтового параметра уменьшить на единицу.

SAVEVALUE A,B,C

Присваивает (если A+ – увеличивает, A– – уменьшает) сохраняемой величине, указанной в параметре A (только номер или имя без X), значение параметра B. C – тип сохраняемой величины. Ссылка на сохраняемую величину выполняется в виде группового имени X, за которым следует либо номер сохраняемой величины, либо после символа \$ – символьное имя этой величины, например X\$DASP.

Для задания начальных значений сохраняемым переменным используется карта INITIAL.

INITIAL A1,B1/ ... /AN,BN

Ai – имя сохраняемой величины,

Bi – ее начальное значение.

Если сохраняемая величина не описана в карте INITIAL, то по умолчанию ее значение равняется 0.

Примеры использования оператора SAVEVALUE:

- 1) SAVEVALUE SI,12,PF – присвоить переменной с именем SI полнословного целого типа значение 12.

2) SAVEVALUE 12,-3,PB – присвоить переменной байтового типа с номером 12 значение -3.

3) SAVEVALUE A+,1,PB – увеличить значение A на единицу.

TEST X A,B,C

Проверяет отношение X (L(<), LE(<или=), E(=), NE(<>), GE(>или=), G(>)) между значениями параметров A и B. Если оно выполняется, то транзакт обрабатывается следующим блоком, если нет, то блоком, метка которого указана в блоке C.

Примеры:

1) TEST NE 12,X\$TAST,TST – управление передается блоку с меткой TST, если переменная TAST равна 12.

2) TEST LE P1H,P2H,GO – если второй параметр транзакта больше первого, то управление передается по метке GO.

Другой формой оператора TEST является оператор IF.

IF X,A

Осуществляет переход на метку A, если логическое условие X истинно.

Примеры использования оператора IF вместо TEST:

1) IF X\$TAST=12,TST

2) IF P1H>P2H,GO

Пример 5. Пусть в данном примере транзакты соответствуют грузовым автомобилям. Грузоподъемность каждого автомобиля будет содержаться в параметре транзакта. Существует 2 вида грузовиков – грузоподъемностью в 5 и 10 т. Они перевозят грузы на некоторый объект. Пусть в данный момент разгружаться может только один автомобиль. Поэтому на разгрузку образуется очередь, которая будет иметь имя Q. Сам объект назовем OBJ. Масса груза, уже перевезенная на объект, будет храниться в ячейке MAS.

```
INITIAL MAS,100                                ;пусть на объект уже
                                                ;перевезено 100 т груза

;1-й сегмент программы
; описывает поток автомобилей грузоподъемностью 5 т
GENERATE 15,5,, ,1B                            ;автомобили подъезжают к
                                                ;объекту через 15±5 мин

ASSIGN 1,5,PB
QUEUE Q
SEIZE OBJ
DEPART Q
ADVANCE 5,1                                    ;задержка на разгрузку
SAVEVALUE MAS+,P1H,XH                         ;увеличение значения массы груза на объекте
RELEASE OBJ
TERMINATE
; 2-й сегмент программы
;описывает поток автомобилей грузоподъемностью 10 т
```

```

GENERATE 25,10, , , ,1B
ASSIGN 1,10,PB
QUEUE Q
SEIZE OBJ
DEPART Q
ADVANCE 10,2
SAVEVALUE MAS+, ,P1H,XH
RELEASE OBJ
TERMINATE 1
START 250

```

TRANSFER A

Осуществляется безусловный переход на метку A. Другая форма:

GOTO A

PREEMPT A,B,C,D,E

Транзакт занимает прибор с именем A, т.е. если другой транзакт обслуживается в нем, то его обслуживание прерывается и он передается на обработку блоку, указанному в C, причем в его параметре с номером D записывается время, оставшееся до конца обслуживания. Если B=PR, то обслуживание этого транзакта также может быть прервано транзактом с более высоким приоритетом. Если B опущено, то обслуживание этого транзакта прервать нельзя. Если E=RE, то прерванный транзакт теряет право на автоматическое восстановление в приборе.

RETURN A

Транзакт, захвативший прибор с именем A, освобождает его.

Функции используются для вычисления величин, заданных табличными зависимостями. Каждая функция определяется перед началом моделирования с помощью оператора определения FUNCTION, имеющего следующий формат:

j FUNCTION A,B,

где:

j – метка

A – аргумент функции, который может быть СЧА или один из восьми датчиков RN, равномерно распределенных в интервале [0,1[; **RNj**, где **j = 1..8**

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

B – состоит из одной буквы, определяющей тип функции, и целого положительного числа точек таблицы, для которых вычисляется функция, каждая точка таблицы задается парой Xi (значение аргумента) и Yi (значение

функции), отделяемых друг от друга запятой. Пары координат отделяются друг от друга символом "/" и располагаются на произвольном количестве строк. Значения X_i и Y_i могут быть нецелыми числами. А последовательность значений аргумента X_i должна быть строго возрастающей, то есть удовлетворять условию $X_i < X_{i+1}$.

Существует пять типов функций

- 1). Непрерывные числовые значения C_n
- 2) дискретные числовые значения D_n
- 3) перечень числовых значений L_n
- 4) дискретные значения атрибутов E_m
- 5) перечень значений атрибутов M_n

Первые два типа функции, являются основными.

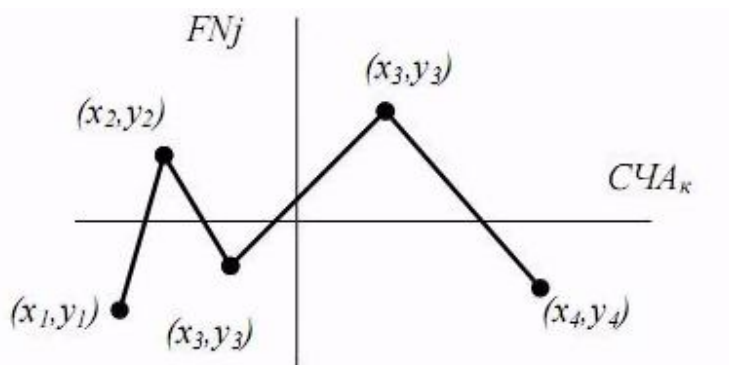


Рис. 1. Непрерывная функция

Функцию, показанную на рис. 1, можно представить следующим образом:

1 FUNCTION RN1, C5
X1,Y1/X2,Y2/X3,Y3/X4,Y4/X5,Y5

Например, если необходимо, чтобы случайная величина получала значения 1, 4, 5 с относительной частотой 0,40; 0,10; 0,50, то дискретная функция может иметь вид

15 FUNCTION RN8, D3
0.4,1/.5,4/1,5

Графическое представление данной функции приведено на рис 2.

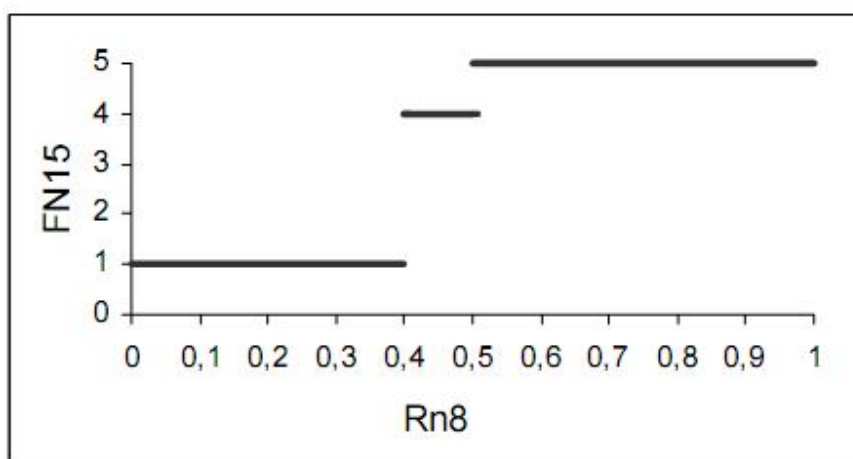


Рис 2.

При использовании аргументов, отличных от RN_j , существуют ограничения:

1. Значениями аргумента должны быть только целые числа;
2. Когда значение аргумента выходит за пределы, указанные в описании функции, то значение последней берется равным ближайшим описанным значениям.

Генерация случайных чисел в GPSS имеет следующие особенности. Все случайные равномерно распределенные числа, используемые в GPSS, получаются в результате расчета, который исходит из набора восьми основных чисел, называемых исходными. Пользователь может задать любое из этих чисел ($RN1, \dots, RN8$). В случаях, когда обращение к датчику случайных чисел подразумевается, используется RN_j . Начальные значения и, следовательно, последовательности псевдослучайных чисел, получаемые во всех системных датчиках ($RN1, \dots, RN8$), одинаковы. Для получения различных последовательностей необходимо изменять их начальное значение.

Для вызова функции любого типа используется СЧА $FN\$name$ (без пробелов), значением этого СЧА и является вычисленное значение функции.

Вычисление функции производится при входе транзакта в блок, содержащий ссылку на СЧА FN с именем функции.

При использовании функции в поле **B** 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 распределением, должна описывать зависимость $y = -\ln(x)$, представленную в табличном виде.

Оператор FUNCTION с такой таблицей, содержащей 24 точки для обеспечения достаточной точности аппроксимации, имеет следующий вид:

```
ZAD1 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
```

Вычисление непрерывной функции производится следующим образом.

Сначала определяется интервал $(X_i; X_{i+1})$, на котором находится текущее значение СЧА-аргумента (в нашем примере - сгенерированное значение RN1). Затем на этом интервале выполняется линейная интерполяция с использованием соответствующих значений Y_i и Y_{i+1} . Результат интерполяции усекается (отбрасыванием дробной части) и используется

в качестве значения функции. Если функция служит операндом В блоков **GENERATE** или **ADVANCE**, то усечение результата производится только после его умножения на значение операнда А.

Будем использовать функцию в блоках **GENERATE** и **ADVANCE**.

Поскольку в обеих моделях используется один и тот же генератор RN1, интервалы поступления и задержки, вычисляемые в блоках **GENERATE** и **ADVANCE**, должны получиться весьма близкими, а может быть и идентичными. При большом количестве транзактов, пропускаемых через модель (десятки и сотни тысяч), разница в скорости вычислений должна стать заметной от случая, если бы мы использовали переменные.

```
ZAD1 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 100,FN$EXP
ADVANCE 80,FN$EXP
TERMINATE 1
```

Порядок выполнения работы

1. Разработайте имитационные модели, заданных СМО.
2. Оформите отчет по лабораторной работе, который должен содержать:
 - задание;
 - текст программы с комментариями;
 - результаты моделирования с комментариями.

Варианты заданий

Задание 1. Смоделировать одноканальную СМО с очередью (таблица 1) для 1000 клиентов. Время между прибытиями посетителей подчинено равномерному закону. Время обслуживания также распределено равномерно. Собрать статистику об очереди.

Таблица 1

Вариант	Поступление заявки	Обработка заявки
1	15 ± 3	13 ± 3
2	23 ± 3	26 ± 3
3	16 ± 2	14 ± 2
4	21 ± 2	25 ± 2
5	5 ± 1	3 ± 2
6	4 ± 2	6 ± 1
7	8 ± 1	5 ± 2
8	10 ± 2	14 ± 1
9	11 ± 4	15 ± 2
10	12 ± 1	8 ± 3

Задание 2. Смоделировать процесс прохождения заявок через прибор (таблица 2). Поступление заявок подчиняется равномерному закону с интервалом $X \pm Y$ мин, обработка – экспоненциальному (см. таблица 4). Если у прибора нет возможности принять заявку, она становится в очередь. Собрать статистику об очереди.

Таблица 2

Вариант	Поступление заявки	Обработка заявки
1	5 ± 1	$13 \pm e^x$
2	8 ± 1	$26 \pm e^x$
3	21 ± 2	$14 \pm e^x$
4	16 ± 2	$25 \pm e^x$
5	15 ± 3	$3 \pm e^x$
6	11 ± 4	$6 \pm e^x$
7	23 ± 3	$5 \pm e^x$
8	12 ± 1	$14 \pm e^x$
9	4 ± 2	$15 \pm e^x$
10	10 ± 2	$8 \pm e^x$

Задание 3. Рассматривается система с потерями (таблица 3). Число мест в очереди ограничено Z . В случае, если все Z мест в очереди заняты, заявка теряется. Поступление заявок подчинено экспоненциальному закону (см. таблица 4), обработка \square равномерному закону.

Таблица 3

Вариант	Поступление заявки	Обработка заявки	Количество мест
1	$18 \pm e^x$	5 ± 1	2
2	$21 \pm e^x$	8 ± 1	3
3	$17 \pm e^x$	21 ± 2	4
4	$22 \pm e^x$	16 ± 2	5
5	$6 \pm e^x$	11 ± 4	3
6	$3 \pm e^x$	15 ± 3	4
7	$12 \pm e^x$	23 ± 3	5
8	$19 \pm e^x$	12 ± 1	6
9	$11 \pm e^x$	4 ± 2	5
10	$5 \pm e^x$	10 ± 2	4

Таблица 4

$\exp(x/y)$	
x	y
0	0
0.1	0.104
0.2	0.222
0.3	0.335
0.4	0.509
0.5	0.69
0.6	0.915
0.7	1.2
0.75	1.38
0.8	1.6
0.84	1.83
0.88	2.12
0.9	2.3
0.92	2.52
0.94	2.81
0.95	2.99
0.96	3.2
0.97	3.5
0.98	3.9
0.99	4.6
0.995	5.3
0.998	6.2
0.999	7
1	8

Контрольные вопросы

1. Понятие моделирования. Характеристики моделей.
2. Понятия моделирования: система, внешняя среда. Типы атрибутов элементов и системы.
3. Классификация систем.
4. Основные понятия: событие, действие, процесс, очередь, модельное время, системы массового обслуживания, системная величина.
5. Элементы процедуры решения: события, категории событий; таймер модельного времени, методы увеличения значения таймера; завершение моделирования; алгоритмизация моделирования.
6. Языки имитационного моделирования систем: SIMULA, SIMSCRIPT, GPSS и др. Имитационное моделирование систем на GPSS.
7. Блочно-ориентированная концепция GPSS.
8. Функциональная структура GPSS. Типы объектов: транзакты, блоки, списки, устройства, памяти, логические ключи, очереди, таблицы, ячейки, функции, переменные.
9. Понятие транзакта. Списки событий (текущих и будущих). Блоки GPSS, связанные с транзактами.
10. Блок GENERATE создания транзакта. Его параметры и стандартные числовые атрибуты (СЧА). Пример использования блока GENERATE.
11. Блок ASSIGN присваивания и изменения значений параметров. Запись текущего модельного времени в заданный параметр транзакта
12. Блок MARK Изменение приоритета транзакта. Блок PRIORITY. Удаление транзактов из модели. Блок TERMINATE.
13. Моделирование обслуживания заявок (задержки транзактов на определенный отрезок модельного времени) с помощью блока ADVANCE.
14. Переменные и функции. Оператор VARIABLE. Определение функций. Пример модели.
15. Блоки GPSS, связанные с аппаратными объектами. Блоки SIZE создания и RELEASE освобождения одноканальных устройств
16. Моделирования захвата и освобождения одноканального устройства с помощью блоков PREEMPT и RETURN.
17. Определение многоканальных устройств (MKY). Оператор определения STORAGE (память).
18. Блоки ENTER (войти) и LEAVE (покинуть) занятия и освобождения каналов обслуживания MKY.
19. Создание объектов типа «очередь». Блоки QUEUE (стать в очередь) DEPART (уйти из очереди). Оператор QTABLE создания таблицы.
20. Задержка или изменение маршрутов транзактов с помощью блока GATE.
21. Приемы конструирования GPSS-моделей. Технология работы с пакетом GPSS. Приемы конструирования GPSS-моделей.
22. Загрузка интегрированной среды. Ввод новой модели. Редактирование текста модели. Запись и считывание модели с диска.
23. Прогон модели и наблюдение за моделированием. Получение и интерпретация стандартного отчета. Примеры построения GPSS-моделей.

Приложения

Приложение 1.

Результаты моделирования.

Строка заголовка стандартного отчета содержит имя файла модели, который создал отчет. Также в нее включена дата и время прогона модели.

GPSS World Simulation Report – Пример_2.8.24.2

Tuesday, September 02, 2003 11:45:55

1. Общая информация о результатах работы модели

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	43200.000	23	1	0

START TIME – начальное время. Абсолютное модельное время в момент начала моделирования. Устанавливается равным абсолютному модельному времени с помощью оператора RESET или CLEAR;

END TIME – конечное время. Абсолютное модельное время, когда счетчик завершения принимает значение 0;

BLOCKS – количество блоков, использованных в текущей модели, к моменту завершения моделирования;

FACILITIES – количество устройств, использованных в модели, к моменту завершения моделирования;

STORAGES – количество многоканальных устройств, использованных в текущей модели к моменту завершения моделирования.

2. Информация об именах

Файл статистики содержит информацию об именах, которые просматривает GPSS в ходе моделирования.

NAME	VALUE
REM1	10005.000
REMQ	10003.000
REMQ1	10006.000
REMQ2	10004.000
VRREM	10002.000
VRREM1	10000.000
VRREM2	10001.000

NAME – перечень заданных пользователем имен, содержащихся в программе модели;

VALUE – числовое значение, присваиваемое имени. Система начинает отсчет с 10 000.

3. Информация о блоках

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	728	0	0
	2	QUEUE	728	0	0
	3	QUEUE	728	153	0
	4	SEIZE	575	0	0
	5	DEPART	575	0	0
	6	DEPART	575	0	0
	7	ADVANCE	575	0	0
	8	RELEASE	575	0	0
	9	TABULATE	575	0	0
	10	TRANSFER	575	0	0
	11	GENERATE	595	0	0
	12	QUEUE	595	0	0
	13	QUEUE	595	124	0
	14	SEIZE	471	0	0
	15	DEPART	471	0	0
	16	DEPART	471	0	0
	17	ADVANCE	471	0	0
	18	RELEASE	470	0	0
	19	TABULATE	470	0	0
MET1	20	TABULATE	1045	0	0
	21	TERMINATE	1045	0	0
	22	GENERATE	100	0	0
	23	TERMINATE	100	0	0

LABEL – метка, алфавитно–цифровое имя данного блока (если оно задано);

LOC – числовой номер позиции данного блока в модели;

BLOCK TYPE – тип блока GPSS;

ENTRY COUNT – количество транзактов, вошедших в данный блок, с начала работы программы или после последнего выполнения оператора RESET или CLEAR;

CURRENT COUNT – количество транзактов, находящихся в данном блоке к моменту завершения моделирования;

RETRY – количество транзактов, ожидающих специального условия, зависящего от состояния данного блока.

4. Информация об объектах типа «устройство»

	FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
REM1	82	0.828	435.987	1	92	0	0	0	0	

Элементы статистики, представленные в данном разделе, имеют следующее содержание:

FACILITY – имя или номер устройства;

ENTRIES – количество раз, когда устройство было занято или занято с прерыванием с начала моделирования или после последнего выполнения оператора RESET или CLEAR;

UTIL. – коэффициент использования, доля времени моделирования, в течение которого устройство было занято;

AVE.TIME – среднее время занятия устройства одним транзактом в течение времени моделирования с начала моделирования или после выполнения оператора RESET или CLEAR;

AVAIL. – состояние устройства в конце моделирования (равно 1, если устройство доступно; 0 – если недоступно);

OWNER – номер транзакта, который занимает устройство (0 – устройство не занято);

PEND – количество транзактов, ожидающих выполнения с прерыванием других транзактов (т. е. вошедших в блоки PREEMPT в режиме прерывания);

INTER – количество транзактов, прерванных на данный момент (количество транзактов в списке прерываний);

RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данного устройства;

DELAY – количество транзактов, ожидающих занятия устройства (входят также транзакты, ожидающие занятия устройства в режиме прерывания с помощью блоков PREEMPT).

5. Информация об объектах типа «очередь»

	QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
REM0	2	0	82	28	0.461	242.755	368.628	0	
REM02	2	0	47	19	0.237	218.158	366.195	0	
REM01	1	0	35	9	0.223	275.785	371.249	0	

QUEUE – имя или номер очереди;

MAX – максимальное содержимое очереди в течение периода моделирования, который начинается с момента трансляции модели или применения оператора RESET или CLEAR;

CONT. – текущее содержимое очереди в конце процесса моделирования;

ENTRY – общее количество входов транзактов в очередь в течение времени моделирования.

ENTRY(O) – общее количество входов транзактов в очередь с нулевым временем ожидания;

AVE.CONT. – среднее значение содержимого очереди в течение времени моделирования;

AVE.TIME – среднее время пребывания одного транзакта в очереди с учетом всех входов в очередь;

AVE.(–0) – среднее время пребывания одного транзакта в очереди без учета «нулевых» входов в очередь;

RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния очереди.

6. Информация об объектах типа «многоканальное устройство»

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
PUN1	5	5	0	5	54	1	0.368	0.074	0	0
PUN3	4	4	0	4	74	1	0.103	0.026	0	0

STORAGE – имя или номер памяти;

CAP. – емкость памяти, заданная оператором STORAGE;

REM. – число единиц памяти, свободных в конце моделирования;

MIN. – минимальное число единиц памяти, использовавшихся за период моделирования;

MAX. – максимальное число единиц памяти, использовавшихся за период моделирования;

ENTRIES – количество входов в память за период моделирования;

AVL. – состояние памяти в конце моделирования (1 – доступно; 0 – недоступно);

AVE.C – среднее значение занятой емкости за период моделирования;

UTIL. – коэффициент использования памяти;

RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной памяти;

DELAY – количество транзактов, ожидающих в блоках ENTER, связанных с данной памятью.

7. Информация о таблицах

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM. %
VRREM1	591.899	271.392		0		
			– 420.000		11369	31.54
			420.000 – 600.000		9972	59.21
			600.000 – 780.000		6765	77.98
			780.000 – 960.000		4418	90.23
			960.000 –		3520	100.00

TABLE – имя или номер таблицы или Q-таблицы;
 MEAN – средневзвешенное значение табулируемого аргумента;
 STD.DEV. – взвешенное среднеквадратическое отклонение:

$$STD.DEV = \sqrt{ \left(\frac{SOS}{COUNT-1} \right) - \left(\frac{SUM2}{COUNT} \right) \left(\frac{COUNT-1}{COUNT} \right) },$$

где SOS – накопленная сумма квадратов;

RANGE – нижний и верхний пределы частотного класса:

- при попадании табулируемого аргумента в интервал, который имеет значение больше или меньше нижней границы частотного класса или равное верхней границе, изменяется значение частоты (FREQUENCY);
- операнд В (весовой коэффициент) блока TABULATE может быть использован для определения величины, которая добавляется в частотный класс при попадании табулируемого значения в этот частотный класс;
- частотные классы, суммарное значение которых равно 0, в файл статистики не выводятся;
- RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной таблицы;
- FREQUENCY – суммарная величина, которая формируется при попадании табулируемого аргумента в указанные границы (значения операнда В суммируются блоком TABULATE);
- CUM.% – величина частоты в процентах к общему количеству значений табулируемого аргумента.

8. Информация о списках пользователя

USER	CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
NAK		0	0	0.260	2709	1	3.458

USER CHAIN – имя или номер списка пользователя;

SIZE – количество транзактов в списке пользователя в конце времени моделирования;

RETRY – количество транзактов, ожидающих наступления специального условия, зависящего от состояния данного списка пользователя;

AVE.CONT. – среднее содержимое списка пользователя в течение времени моделирования;

ENTRIES – общее число транзактов, входивших в список пользователя в течение времени моделирования;

MAX – максимальное количество транзактов в списке пользователя за период моделирования;

AVE.TIME – среднее время пребывания транзакта в списке пользователя.

9. Информация о группах транзактов

XACT GROUP	GROUP SIZE	RETRY
USERGR	10	0

XACT GROUP – имя или номер объекта группы транзактов;

GROUP SIZE – число транзактов в группе в конце моделирования;

RETRY – число транзактов, ожидающих выполнения специального условия, зависящего от состояния данной группы транзактов.

10. Информация о числовых группах

NUMERIC GROUP	GROUP SIZE	RETRY
DEPOSITE	1	0

NUMERIC GROUP – имя или номер числовой группы;

GROUP SIZE – количество транзактов, содержащихся в числовой группе в конце моделирования;

RETRY – число транзактов, ожидающих выполнения специального условия, зависящего от состояния данной числовой группы.

11. Информация о логических переключателях

LOGICSWITCH	VALUE	RETRY
SWITCH1	1	0

LOGICSWITCH – имя или номер логического ключа;

VALUE – значение ключа в конце моделирования (1 (true) – «установлен»; 0 (false) – «сброшен»);

RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данного логического ключа.

12. Информация о сохраняемых величинах (ячейках)

SAVEVALUE	RETRY	VALUE
KOLVIDMS	0	13.000
VSETRANS	0	8.000

SAVEVALUE – имя или номер ячейки;

VALUE – значение сохраняемой величины в конце моделирования;

RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной ячейки.

13. Информация о матрицах

MATRIX	RETRY	INDICES	VA
PLAN	0		
		1 1	52
		1 2	0
		1 3	64

MATRIX – имя или номер матрицы;
 RETRY – количество транзактов, ожидающих выполнения специального условия, зависящего от состояния данной матрицы;
 INDICES – до шести целых чисел, определяющих элемент матрицы;
 VALUE – значение элемента матрицы в конце моделирования (элементы, равные 0, выводятся в отчете группами).

14. Информация о списках текущих и будущих событий

CEC	XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
32		0	3600138.403	1713	0	29		

Списки текущих (CEC) и будущих (FEC) событий выводятся в файл статистики, если в команде START значение операнда D равно 1.

Элементы статистики для CEC, представленные в данном разделе, имеют следующее содержание:

- XN – номер каждого транзакта, находящегося в списке текущих событий;
- PRI – приоритет транзакта;
- M1 – время входа транзакта в модель или время его самого раннего предка (породившего данный транзакт);
- ASSEM – номер семейства данного транзакта;
- CURRENT – номер блока, в котором находится транзакт в конце моделирования;
- NEXT – номер следующего блока, в который должен был войти транзакт;
- PARAMETER – имя или номер параметра транзакта;
- VALUE – значение параметра.

В содержании статистики для FEC имеется только одно отличие. Вместо поля M1 присутствует поле BDT, которое определяет момент абсолютного модельного времени, когда транзакт покинет список будущих событий.

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
53		0	3600142.809	1719	0	29		
32		0	3600251.545	1717	0	15		
16		1	3604171.277	1714	52	53	1	1.000

Приложение 2.

Функциональная структура GPSS

В состав GPSS входят следующие типы объектов: транзакты, блоки, списки, устройства, памяти, логические ключи, очереди, таблицы, ячейки, функции, переменные. Любую модель на языке GPSS можно представить в виде комбинации компонентов, взятых из числа названных объектов. Модель имеет три уровня представления:

- **верхний уровень**, определяемый комбинацией функциональных основных объектов: устройств, памяти, ключей, очередей;
- **средний уровень**, представляемый схемой из типовых блоков, между которыми перемещаются транзакты;
- **нижний уровень** – уровень физической реализации языка GPSS в виде программ и наборов данных, составляющих основу моделирующей системы.

1.1. Блоки

Разработчик конструирует модель из блоков, прибегая, как правило, к наглядной форме ее отображения в виде блок-схемы. Для удобства графического представления модели каждый блок GPSS имеет принятое стандартное обозначение. Построенная схема является одновременно программой на языке GPSS. Для ее ввода в ЭВМ необходимо последовательность блоков представить в виде списка операций, добавив к названиям блоков требуемые операнды.

Каждый блок GPSS имеет входы и выходы, с помощью которых осуществляется их связь в модели. Существуют два особых блока: GENERATE, имеющий только выход, и TERMINATE, имеющий только вход. Через блок GENERATE транзакты вводятся в модель. Блок TERMINATE удаляет транзакты из модели. Любую модель на языке GPSS можно представить в виде совокупности блоков:



СВЯЗАННЫЕ СЧА:

N\$j – общее число входов в блок j ;

W\$j – текущее число транзактов в блоке j.

1.2. Транзакты

Функционирование объекта отображается в модели в виде перемещения транзактов от блока GENERATE в блок TERMINATE через промежуточные блоки. Транзакты, или сообщения, являются абстрактными подвижными элементами, которые могут моделировать различные объекты реального мира: сообщения, программы, транспортные средства, людей и т.п. Перемещаясь между блоками модели, транзакты вызывают (и испытывают) различные действия. Возможны их задержки в некоторых точках модели, изменения маршрутов и направлений движения, расщепление транзактов на несколько копий и т.п. С каждым транзактом связан упорядоченный набор данных. Он включает номер транзакта; номер блока, в котором в данный момент находится транзакт; номер следующего блока; время перехода в следующий блок; приоритет, характеризующий очередность обработки транзактов в определенных случаях; а также набор параметров, с помощью которых каждому транзакту можно присвоить числовые значения, выражающие желаемые свойства или характеристики моделируемых объектов: вес, скорость, объем, цвет, время обработки и т.п.

МОДЕЛИРУЮЩИЕ БЛОКИ:

Основные блоки:

GENERATE A,B,C,D,E

Вводит транзакты в модель.

A – среднее значение интервала времени (необязательный операнд);

B – разброс или модификатор среднего значения (необязательный операнд);

C – время появления первого транзакта (необязательный операнд);

D – общее число генерируемых транзактов (необязательный операнд);

E – уровень приоритета каждого транзакта (необязательный операнд).

TERMINATE A

Удаляет активный транзакт из процесса моделирования.

A – величина уменьшения счетчика завершения (необязательный операнд).

ADVANCE A,B

Задерживает транзакт.

A – среднее время задержки;

B – разброс или модификатор среднего значения (необязательный операнд).

Блоки, изменяющие значения параметров транзактов:

ASSIGN A,B,C

Изменяет значение параметра транзакта.

A – номер изменяемого параметра;

B – новое значение параметра;

C – номер функции (необязательный операнд).

INDEX A,B

Обновляет параметр активного транзакта.

A – номер параметра;

B – числовое значение, которое должно быть добавлено к содержимому параметра.

MARK A

Записывает в параметр активного транзакта значение абсолютного модельного времени.

A – номер параметра (необязательный операнд).

PRIORITY A,B

Устанавливает приоритет активного транзакта.

A – новое значение приоритета;

B – BU (необязательный операнд).

Блоки, управляющие движением транзактов:

GATE 0 A,B

Изменяет маршрут движения транзактов в зависимости от состояния некоторого объекта.

O – Условие в виде стандартного логического атрибута (FNV,FV, I, LS, LR, M, NI, NM, NU, SE, SF, SNE, SNF, SNV, SV,U);

A – номер проверяемого объекта;

B – номер блока перемещения транзакта в случае невыполнения условия (необязательный операнд).

LOOP A,B

Изменяет параметр и управляет местом назначения активного транзакта на основании результата.

A – параметр, содержащий число;

B – номер следующего блока, если число после уменьшения не ноль (необязательный операнд).

TEST 0 A,B,C

Сравнивает значения, обычно СЧА и управляет местом назначения активного транзакта, основываясь на результате сравнения.

O – оператор отношения (E, G, GE, L, LE, NE);

A – проверяемое значение;

B – контрольное значение;

C – номер блока назначения (необязательный операнд).

TRANSFER A,B,C,D

Обеспечивает переход активного транзакта к новому блоку.

A – режим (BOTH, ALL, PICK, FN, P, SBR, SIM) (необязательный операнд);

B – номер или метка блока (необязательный операнд);

C – номер или местоположение блока (необязательный операнд);

D – приращение номера блока для режима ALL (необязательный операнд).

Блоки, работающие с транзактами в списке пользователя:

LINK A,B,C

Управляет размещением активного транзакта в списке пользователя.

A – номер списка пользователя;

B – упорядочивание списка;

C – местоположение следующего блока (необязательный операнд).

UNLINK O A,B,C,D,E,F

Управляет удалением активного транзакта из списка пользователя.

O – оператор отношения (E, G, GE, L, LE, NE);

A – номер списка пользователя;

B – номер блока назначения для удаленных транзактов;

C – максимальное количество транзактов, которое можно удалить (необязательный операнд);

D – проверяемое значение (необязательный операнд);

E – контрольное значение (необязательный операнд);

F – номер блока (альтернативное место назначения для входящего транзакта) (необязательный операнд).

Блоки, связанные с трассировкой транзактов:

TRACE

Устанавливает индикатор трассировки активного транзакта.

UNTRACE

Сбрасывает индикатор трассировки активного транзакта.

Блоки, работающие с копиями транзактов:

ASSEMBLE A

Ожидает и уничтожает транзакты.

A – счетчик транзактов.

GATHER A

Накапливает транзакты, являющиеся членами семейства.

A – счетчик транзактов, которые должны быть накоплены.

MATCH A

Пара сопряженных блоков MATCH заставляет транзакты ожидать друг друга.

A – номер блока, который проверяется на наличие транзакта.

SPLIT A,B,C

Создает транзакты того же семейства, что и активный транзакт.

A – количество создаваемых транзактов;

B – номер блока (необязательный операнд);

C – номер параметра (необязательный операнд).

СВЯЗАННЫЕ СЧА:

A1 – указатель семейства активного транзакта;

MB\$_j – СЧА равен 1, если в блоке j находится транзакт, который принадлежит тому же семейству, что и активный;

MP_j – разница между текущим значением абсолютного модельного времени и значением времени, записанным в параметре j;

M1 – время пребывания транзакта в системе;

P_j – значение параметра с номером j;

PR – приоритет активного транзакта;

XN1 – номер активного транзакта.

1.3. Списки

Списки относятся к элементам внутренней организации системы GPSS. Они представляют собой структуры данных, в которых размещается полная информация о транзактах. С помощью списков обеспечивается внутренняя логика работы моделирующей системы.

Существует несколько видов списков:

1) Список будущих событий

Список будущих событий содержит транзакты, которые смогут начать движение в модели в будущие моменты времени. Это те транзакты, для которых моменты начала движения определены в блоках GENERATE и ADVANCE.

Содержимое списка можно наблюдать – *Window\Simulation Snapshot\FEC Snapshot* («Кадр СБС»).

2) Список текущих событий

В список текущих событий входят транзакты, которые должны перемещаться в модели в текущий момент модельного времени. Если при этом транзакт входит в блок ADVANCE с ненулевым временем задержки, то он перемещается в список будущих событий.

Содержимое списка можно наблюдать – *Window\Simulation Snapshot\CEC Snapshot* («Кадр СТС»).

3) Список пользователя

Содержимое списка можно наблюдать – *Window\Simulation Snapshot\UserChains Snapshot* («Кадр списков пользователя»)

4) Список прерываний

В список прерываний помещаются транзакты, обслуживание которых прервано блоком PREEMPT. После снятия прерывания в блоке RETURN транзакты вновь возвращаются в список будущих событий.

5) Список синхронизации

В список синхронизации помещаются транзакты, ожидающие объединения с другими транзактами в блоках GATHER и ASSEMBLE или находящиеся в блоках MATCH. После выполнения условий синхронизации транзакты возвращаются в список текущих событий. Кроме рассмотренных списков, обработка которых происходит без

участия программиста, в GPSS/PC существуют списки пользователя, управление которыми осуществляется с помощью блоков LINK и UNLINK.

СВЯЗАННЫЕ СЧА:

CA\$j – среднее число транзактов в списке j;

CCSj – общее число входов транзактов в список j;

CH\$j – текущее число транзактов в списке j;

CMSj – максимальное число транзактов в списке j;

CT\$j – среднее время пребывания транзакта в списке j.

1.4. Устройства

Устройства моделируют объекты, в которых может происходить обработка транзактов. Как правило, она связана с затратами времени. Особенность устройств состоит в том, что каждое из них в данный момент времени может быть занято лишь одним транзактом. Существует аналогия между устройствами GPSS и каналами систем массового обслуживания. В GPSS имеется возможность моделировать прерывания устройств. Существуют средства логической проверки состояния устройств.

МОДЕЛИРУЮЩИЕ БЛОКИ:

SEIZE A

Занимает устройство.

A – номер устройства.

RELEASE A

Освобождает устройство.

A – номер устройства (числовое или символьное имя освобождаемого устройства).

PREEMPT A

Переводит устройство в прерванное состояние.

A – номер прерываемого устройства.

RETURN A

Удаляет транзакт из прерванного устройства.

A – номер устройства (числовое или символьное имя освобождаемого устройства).

FAVAIL A

Обеспечивает перевод устройства с номером A в доступное состояние

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

Обеспечивает перевод устройства в недоступное для входа транзактов состояние.

Проверка состояния устройства может быть выполнена с помощью блока **GATE**.

СВЯЗАННЫЕ СЧА:

FSj – состояние устройства с номером j: 0 если устройство свободно, и 1 – если устройство занято;

FC\$j – количество транзактов, занимавших устройство;

FI\$j – если устройство в настоящий момент прервано блоком PREEMPT, то 1.

FR\$j – коэффициент использования устройства j. Доля времени, когда устройство было занято. Принимает значение от 0 до 1000;

FT\$j – среднее время использования устройства j одним транзактом;

FV\$j – если устройство находится в доступном состоянии, то 1.

1.5. Памяти

Памяти служат для моделирования объектов, обладающих определенной емкостью. Емкость памяти задают с помощью оператора STORAGE. Транзакт может занимать и освобождать определённую часть памяти.

МОДЕЛИРУЮЩИЕ БЛОКИ:

ENTER A,B

Помещает транзакт в память.

A – имя памяти символическое или числовое;

B – число занимаемых единиц памяти (необязательный операнд).

LEAVE A,B

Выводит транзакт из памяти.

A – номер памяти;

B – число освобождаемых единиц, памяти (необязательный операнд).

SAVAIL A

Обеспечивает доступное состояние памяти, имя или номер которого указаны в A.

SUNAVAIL A

Обеспечивает недоступное состояние памяти.

СВЯЗАННЫЕ СЧА:

R\$j – емкость неиспользованной памяти j;

S\$j – емкость используемой памяти j;

SA\$j – среднее заполнение памяти j;

SC\$j – число входов в память j;

SE\$j – память j пуста;

SF\$j – память j заполнена;

SM\$j – максимальное заполнение памяти j;

ST\$j – среднее время использования элемента в памяти j;

SV\$j – память j в доступном состоянии;

SR\$j – коэффициент использования памяти j. Доля общего использования, представленная средним объемом используемой памяти в памяти j.

1.6. Очереди

Транзакты в процессе движения могут задерживаться перед блоками, вход в которые в данных условиях невозможен. При поступлении транзактов на вход задерживающих блоков образуются очереди. Для сбора статистики об очередях в местах задержки ставят блоки QUEUE. Эти блоки сами по себе не создают очередь, а лишь являются средством ее регистрации. При входе транзакта в блок QUEUE текущая длина очереди получает приращение. Уход из очереди отображается блоком DEPART.

МОДЕЛИРУЮЩИЕ БЛОКИ:

QUEUE A,B

Помещает транзакт в конец очереди.

A – номер очереди (числовое или символьное имя очереди);

B – число добавляемых к очереди элементов (необязательный операнд).

DEPART A,B

Удаляет транзакт из очереди.

A – номер (имя) очереди;

B – число удаляемых из очереди элементов (необязательный операнд).

СВЯЗАННЫЕ СЧА:

Q\$j – текущая длина очереди j;

QA_j – средняя длина очереди j ;
 QC_j – число входов в очередь j ;
 QM_j – максимальная длина очереди j ;
 QT_j – среднее время пребывания в очереди j , включая нулевые входы;
 QX_j – среднее время пребывания в очереди j , без нулевых входов.
 QZ_j – число входов в очередь с нулевым временем пребывания (транзакт прошел через блок QUEUE, не задерживаясь в очереди);

1.7. Таблицы

Для сбора статистических данных о различных отчетах модели и их представления в стандартной табличной форме используют таблицы. Занесение информации в таблицу осуществляется блоком TABULATE в момент входа очередного транзакта в этот блок. Описание структуры таблицы и типа заносимых данных (СЧА) осуществляется картой TABLE.

МОДЕЛИРУЮЩИЕ БЛОКИ:

TABULATE A,B

Заносит значение в таблицу.

A – имя или номер таблицы;

B – вес, указывающий сколько раз значение должно быть занесено в таблицу (необязательный операнд).

NAME TABLE A,B,C,D

Определяет таблицу.

NAME – метка объекта;

A – аргумент таблицы (необязательный операнд): элемент данных, плотность распределения которого будет заноситься в таблицу;

B – верхняя граница нижнего интервала гистограммы;

C – ширина интервалов;

D – число интервалов (частотных классов);

NAME QTABLE A,B,C,D

Определяет таблицу статистики очереди (Q–таблицу), гистограмму плотности распределения времени ожидания в очереди

NAME – метка объекта;

A – имя очереди;

B – верхняя граница нижнего интервала гистограммы;

C – ширина интервалов;

D – число интервалов (частотных классов);

СВЯЗАННЫЕ СЧА:

TB $_j$ – среднее значение фиксируемой в таблице j переменной;

TC $_j$ – число входов в таблицу j ;

TD $_j$ – стандартное среднеквадратическое отклонение табулируемой переменной.