

Объекты GPSS World

Имитационная модель в системе GPSS World представляет собой совокупность множества объектов разных типов, взаимодействующих в процессе моделирования. Каждый объект, в общем случае, описывается совокупностью некоторых чисел, хранимых в памяти компьютера.

В GPSS World используются следующие **типы объектов**:

- основные (транзакты; блоки);
- оборудование (устройства; памяти; логические ключи; очереди);
- числовые (функции; матрицы; ячейки; таблицы; переменные);
- групповые (списки пользователя; числовые группы; группы транзактов);
- генераторы случайных чисел;
- потоки данных.

Основными объектами в GPSS-модели являются *блоки*, между которыми перемещаются подвижные объекты, называемые *транзактами*.

Моделирование в среде GPSS World заключается в продвижении транзактов от одного блока модели к другому и выполнении некоторой совокупности действий, предписываемых каждым блоком. Транзакты - единственный тип объектов, которые могут быть удалены из модели в процессе моделирования.

Большинство объектов в GPSS World создаются автоматически в процессе моделирования, однако некоторые объекты перед их использованием должны быть определены заранее (в области описания модели).

Следующие объекты должны быть объявлены перед использованием:

- памяти (с помощью оператора STORAGE);
- арифметические переменные (с помощью оператора VARIABLE);
- переменные с плавающей точкой (с помощью оператора FVARIABLE);
- булевы переменные (с помощью оператора BVARIABLE);
- матрицы (с помощью оператора MATRIX или PLUS-объявлениями временных матриц);
- таблицы (с помощью оператора TABLE);
- специальные Q-таблицы, регистрирующие статистику очередей (с помощью оператора QTABLE);
- функции (с помощью оператора FUNCTION);
- параметры транзактов (должны объявляться в блоках ASSIGN, MARK, READ, SELECT, SPLIT, COUNT или TRANSFER SBR перед обращением к ним).

1. Транзакты

Транзакты – динамические элементы GPSS-модели, движущиеся от начала к концу модели. Работа GPSS-модели заключается в перемещении транзактов от блока к блоку в заданной последовательности.

В начале моделирования в GPSS-модели нет ни одного транзакта. В процессе моделирования транзакты входят в модель в определенные моменты времени в соответствии с условиями, заданными в модели. Аналогично транзакты покидают модель в определенные моменты времени. В общем случае в модели может находиться множество транзактов, однако в один и тот же момент времени двигается только один транзакт.

Транзакт, перемещаемый в данный момент времени, называется **активным**.

При попадании активного транзакта в некоторый блок начинают выполняться действия, предписанные соответствующим оператором, и затем транзакт пытается войти в следующий блок. Такое продвижение активного транзакта продолжается до тех пор, пока не

произойдет одно из следующих возможных событий:

- транзакт входит в блок, функцией которого является задержка транзакта на некоторое определенное моделью время;
- транзакт входит в блок, функцией которого является удаление транзакта из модели;
- при попытке транзакта войти в следующий блок последний "отказывается" принять этот транзакт до тех пор, пока в модели не изменятся некоторые условия.

При появлении одного из перечисленных событий транзакт остается на месте и начинается перемещение в модели другого транзакта из списка текущих событий, который становится активным. Таким образом, моделирование заключается в перемещении транзактов между блоками GPSS-модели и выполнении соответствующих действий (процедур).

Интервал времени, в течение которого транзакт находится в модели, называется **резидентным временем транзакта**.

Интервал времени, в течение которого транзакт проходит от одной произвольно выбранной точки модели до другой точки, называется **транзитным временем** перехода между двумя этими точками.

В процессе моделирования транзакты последовательно нумеруются, начиная с 1. После команды CLEAR нумерация транзактов снова начинается с 1.

Поведение транзактов в модели определяется переменными, которые называются **атрибутами транзактов**. К ним относятся:

- **Параметры** – множество значений, закрепляемые за каждым транзактом. Транзакт может иметь любое число параметров. В качестве идентификатора параметра может использоваться его *номер* (целое положительное число) или символическое *имя*. Обращение к любому параметру активного транзакта осуществляется с помощью СЧА *PParameter*, где *Parameter* - имя или номер параметра. Если значение операнда блока определяется именем или номером параметра, то записи вида *P\$Parameter* или *PParameter* не должны использоваться, а в поле операнда должно использоваться только имя параметра или его номер.

Для увеличения скорости моделирования можно реализовать непосредственный доступ к параметрам, используя в модели блоки параметров. *Блок параметров* – это непрерывный массив параметров, который размещается и освобождается как единый сегмент памяти. При наличии ссылки на параметр из блока параметров, объект «Процесс моделирования» обратится непосредственно к нему, вместо того, чтобы проверять каждый параметр в отдельности, что уменьшает время моделирования. Для использования блоков параметров необходимо объявить их на странице «Simulation» («Моделирование») журнала настроек модели.

Параметры в GPSS применяются для косвенной адресации.

Параметры транзакта должны быть созданы (например, с помощью блоков ASSIGN, MARK, TRANSFER, SBR, SELECT, SPLIT и COUNT) и им должны быть присвоены значения, до того как они будут востребованы.

- **Приоритет.** Приоритет транзакта определяет преимущественное право на использование общего ресурса. Этим преимуществом обладают транзакты с более высоким значением приоритета. Транзакты с одинаковым приоритетом обычно выбираются по правилу FIFO, т.е. в порядке поступления.
- **Время входа транзакта в систему.** - Значение абсолютного времени в момент первого входа транзакта в модель или в блок MARK без операнда A.
- **Указатель семейства.** - Положительное целое число, присваиваемое каждому транзакту. Семейства используются для обеспечения синхронизации транзактов в блоках ASSEMBLE, GATHER и MATCH. Когда транзакт создается блоком

GENERATE, он автоматически входит в одно из семейств, указатель которого устанавливается равным номеру транзакта. Следует отметить, что в этот момент в данном семействе существует только один транзакт. Если транзакт создан блоком SPLIT, то указатель семейства устанавливается равным значению указателя порождающего транзакта. Транзакт может переходить в другое семейство после входа в блок ADOPT.

- **Индикатор задержки** – флаг, устанавливаемый в транзакте после отказа во входе в блок и сбрасываемый блоком TRANSFER SIM. Флаг используется блоком TRANSFER SIM для изменения маршрута движения транзактов.
- **Индикатор трассировки** – флаг, устанавливаемый в транзакте блоком TRACE для формирования трассировочного сообщения при каждом входе транзакта в очередной блок. Индикатор трассировки сбрасывается блоком UNTRACE.
- **Текущий блок** – номер блока, в котором находится данный транзакт.
- **Следующий блок** – номер следующего блока, в который будет пытаться перейти данный транзакт.
- **Списки.** Каждый транзакт в некоторый момент времени может находиться в одном из списков, определяющих его состояние:
 - ACTIVE – транзакт имеет наивысший приоритет и находится в *списке текущих событий*;
 - SUSPENDED – транзакт находится в *списке будущих событий* или в *списке текущих событий* и ожидает возможности стать активным;
 - PASSIVE – транзакт перешел в состояние ожидания и находится в *списке пользователя*, *списке задержки* или *списке отложенных прерываний*;
 - TERMINATED – транзакт удаляется и больше не участвует в процессе моделирования.
 - PREEMPTED – обслуживание транзакта в устройстве прервано, и он находится в одном или более *списке прерываний*.

СЧА транзактов:

- **MPParameter** – **транзитное время параметра**: разница между текущим значением абсолютного модельного времени и значением времени, записанным в параметре Parameter транзакта;
- **M1** – **время пребывания транзакта в системе**: разница между текущим значением абсолютного модельного времени и временем входа транзакта в модель;
- **PPParameter** или ***Parameter** – **значение параметра Paranieter** активного транзакта;
- **PR** – **приоритет** активного транзакта;
- **XN1** – **номер активного транзакта**.

2. Блоки

Блок – основной объект GPSS-модели.

Модель в GPSS World может быть представлена в виде специальной блок-диаграммы и эквивалентной ей программы. **Блок-диаграмма** представляет собой набор стандартных блоков, соединенных между собой в некоторой последовательности, определяемой логикой работы моделируемой системы. Каждому такому блоку в языке GPSS соответствует определенный оператор, реализующий некоторую конкретную функцию. Конфигурация блок-схемы GPSS-модели отображает направления, по которым происходит движение транзактов. Каждый транзакт находится только в одном блоке, но в каждом блоке может одновременно находиться множество транзактов.

Каждый тип блока реализует присущие ему функциональные действия, которые заключаются в преобразовании других объектов. Вначале определяется, Если активный транзакт может войти в данный блок, обновляются несколько статистик блока и транзакта, а также некоторые системные статистики. Затем выполняются действия, определенные данным блоком, и транзакт переходит к следующему блоку. Обычно это «следующий по порядку блок».

Постоянные блоки «процесса моделирования» создаются в процессе трансляции модели из операторов блоков «модели». Блоки создаются в текстовом редакторе объекта «Модель» или при помощи одного или более диалоговых окон создания блока, которые вызываются через пункт меню **Edit / Insert GPSS Block...** (**Правка / Вставить блок GPSS...**).

Операторы блоков, переданные уже существующему «процессу моделирования», создают одноразовые временные блоки в режиме, называемом «ручное моделирование». После ввода такого интерактивного блока активный транзакт делает попытку входа в него, после чего происходит уничтожение временного блока. Таким образом, операторы блоков могут использоваться для интерактивного управления процессом моделирования. После того, как выполнится действие, предусмотренное блоком, активный транзакт возобновляет движение в модели по старому маршруту, если временный блок не являлся блоком TRANSFER или другим блоком с альтернативным местом назначения.

СЧА блоков:

- *NEntnum* – **количество входов в блок**: общее количество транзактов, которые вошли в блок *Entnum*;
- *WEntnum* – **текущее количество транзактов** в блоке *Entnum*.

3. Устройства

Устройства (одноканальные устройства, приборы) – это объекты, которые могут находиться только в одном из двух состояний: «свободно» и «занято». В любой момент времени имеет только один транзакт может «занимать» устройство. Устройство свободно, если оно не занято ни одним транзактом. Устройство не может быть освобождено транзактом, который никогда его не занимал. Транзакты занимают устройства при входе в блоки SEIZE или PREEMPT. Блок PREEMPT позволяет реализовать приоритетное занятие устройства. Если транзакт пытается занять уже «занятое» устройство, то он задерживается и дожидается его освобождения.

Устройство имеет несколько списков для транзактов, которые ожидают наступления события, связанного с устройством. Каждое устройство имеет:

- *список задержки* для транзактов, ожидающих освобождения устройства;
- *список отложенных прерываний* для транзактов, претендующих на прерывание обслуживания других транзактов;
- *список прерываний*, в который входят транзакты, обслуживание которых было прервано.

О транзактах, находящихся в этих трех списках, говорят, что они находятся в состоянии «состязания» за устройство. Каждый транзакт, занимающий устройство, должен со временем освободить его с помощью блоков RELEASE или RETURN. С того момента, когда транзакт в соперничестве с другими транзактами занимает устройство, он берет на себя обязательство в конечном итоге освободить устройство.

Те транзакты, которые при попытке войти в блок SEIZE терпят неудачу, переходят в состояние ожидания и помещаются в порядке приоритета в список задержки. Когда устройство освобождается занимающим его транзактом, следующий владелец устройства выбирается из списков транзактов, ожидающих занятия устройства. Первыми выбираются

транзакты из списка отложенных прерываний, затем выбираются ранее прерванные транзакты и только потом выбираются транзакты, ожидающие освобождения устройства в обычной очереди в соответствии с приоритетами.

Блоки, связанные с устройствами:

- SEIZE – реализует захват устройства.
- RELEASE – освобождает устройство.
- PREEMPT – реализует приоритетный захват устройства.
- RETURN – освобождает устройство после приоритетного захвата.
- FAVAIL – переводит устройство в состояние «доступно».
- FUNAVAIL – переводит устройство в состояние «недоступно».

СЧА устройств:

- F_{Entnum} – **состояние устройства**: 1, если устройство $Entnum$ в данный момент времени занято, и 0, если устройство $Entnum$ – свободно.
- FC_{Entnum} – **количество занятий устройства**: количество транзактов, занимавших устройство $Entnum$ с помощью блоков SEIZE или PREEMPT.
- FI_{Entnum} – **прерванное состояние устройства**: 1, если в данный момент устройство $Entnum$ прервано блоком PREEMPT в режиме прерывания, и 0, в противном случае.
- FR_{Entnum} – **коэффициент использования (загрузка) устройства**: доля времени, в течение которого устройство $Entnum$ было занято (вычисляется как отношение времени занятости устройства $Entnum$ к общему времени моделирования, выражается в долях от тысячи и принимает значения от 0 до 1000).
- FT_{Entnum} – **среднее время использования (занятия) устройства** $Entnum$ одним транзактом.
- FV_{Entnum} – **доступность устройства**: 1, если устройство $Entnum$ находится в доступном состоянии, и 0, если устройство $Entnum$ находится в недоступном состоянии.

4. Памяти

Памяти (многоканальные устройства) представляют собой объект, состоящий из нескольких идентичных элементов, которые могут заниматься или освобождаться транзактами. Памяти могут использоваться как «маркерные накопители» для управления потоком транзактов в модели.

Когда транзакт входит в память (блок ENTER), он использует или занимает один или более элементов памяти. Транзакту запрещается входить в блок ENTER, если его потребность в памяти не может быть удовлетворена. Поэтому он должен будет дожидаться, пока другие транзакты, входя в блоки LEAVE, не освободят достаточное количество памяти.

Емкость памяти может освобождаться любым транзактом, даже если он перед этим не входил в память с помощью блока ENTER. Тем не менее, если производится попытка освободить большее количество элементов памяти, чем было задано для данной памяти оператором описания STORAGE, происходит останов по ошибке.

Когда транзакт входит в блок LEAVE и освобождает один или более элементов памяти, отыскиваются другие транзакты, потребность в памяти которых может быть удовлетворена. Для планирования ожидающих транзактов используется метод «первый подходящий с пропусками». Это означает, что каждый транзакт в очереди задержки, начиная со старших по приоритету, проверяется на наличие в памяти требуемого количества элементов. Если соответствие найдено, транзакт исключается из списка задержки памяти, входит в блок ENTER и помещается в список текущих событий после равных по приоритету

транзактов. Затем проверяется следующий транзакт в списке задержки памяти.

Блоки ENTER и LEAVE используются для обновления статистики, связанной с памятью. Несколько классов СЧА возвращают статистику памяти.

Память *пуста*, когда все элементы памяти доступны для использования, и *заполнена*, когда доступные для использования элементы памяти отсутствуют. Память может быть ни пустой, ни полностью заполненной. Также память может быть *доступной* и *недоступной*. Запросы памяти удовлетворяются только в том случае, если память в доступном состоянии.

Блоки, связанные с памятью:

- ENTER – занимает заданное число элементов памяти (приборов многоканального устройства).
- LEAVE – освобождает заданное число элементов памяти (приборов многоканального устройства).
- SAVAIL – переводит память в доступное состояние.
- SUNAVAIL – переводит память в недоступное состояние.

СЧА памяти (многоканальных устройств):

- *REntnum* – **емкость неиспользованной памяти:** количество элементов памяти (приборов многоканального устройства), доступных для использования входящими транзактами в памяти с именем (номером) *Entnum*.
- *SEntnum* – **используемая память:** количество элементов памяти (приборов многоканального устройства), занятых транзактами в памяти *Entnum*.
- *SAEntnum* – **среднее значение используемой памяти:** взвешенное по времени среднее число используемых элементов памяти (приборов многоканального устройства) *Entnum*.
- *SCEntnum* – **счетчик общего использования памяти:** количество элементов памяти (приборов многоканального устройства), которые были использованы в памяти *Entnum*.
- *SEEntnum* – **память пуста:** 1, если память *Entnum* пуста, и 0 – в противном случае.
- *SFEntnum* – **память заполнена:** 1, если память *Entnum* заполнена, , и 0 – в противном случае.
- *SREntnum* – **коэффициент использования памяти:** средний объем используемой памяти *Entnum* (доля используемых приборов многоканального устройства), выраженный в долях от тысячи и принимающий значение в диапазоне 0-1000 включительно.
- *SMEntnum* – **максимальный объем использовавшейся памяти** *Entnum*.
- *STEntnum* – **среднее время использования элемента** в памяти *Entnum* (одного прибора многоканального устройства).
- *SVEntnum* – **доступность памяти:** 1, если память *Entnum* в доступном состоянии, и 0 – в противном случае.

5. Очереди

Очереди используются для сбора статистики и регистрации текущего количества входов, среднего количества входов, общего количества входов, количества нулевых входов, максимальной длины очереди, среднего времени ожидания.

Для обновления статистики, связанной с очередью, используются блоки QUEUE и DEPART. Обычно между блоками QUEUE и DEPART вводятся блоки SEIZE, PREEMPT или ENTER. Затем статистика, связанная с устройством или памятью, автоматически сохраняется и выводится в отчет. Основные статистики очереди могут быть получены с помощью

вызовов соответствующих СЧА. Плотности распределения могут накапливаться с помощью использования команд QTABLE

Наиболее важный атрибут очереди – ее содержимое. Содержимое очереди изменяется, когда транзакты входят в блоки QUEUE и DEPART. Содержимое можно рассматривать как счетчик объектов, ожидающих в очереди. Регистрация статистик, связанных с содержимым очередей, производится автоматически. Они доступны через СЧА, описанные ниже.

Блоки, связанные с очередями:

- QUEUE – увеличивает содержимое очереди.
- DEPART – уменьшает содержимое очереди.

СЧА очередей:

- *QEntnum* – **текущее значение длины очереди:** количество транзактов в очереди *Entnum* на данный момент времени.
- *QAEntnum* – **среднее значение длины очереди:** взвешенное по времени количество транзактов в очереди *Entnum*.
- *QCEntnum* – **общее количество входов в очередь:** суммарное число всех транзактов, прошедших через очередь *Entnum*.
- *QMEntnum* – **максимальная длина очереди:** Максимальное количество транзактов, находившихся в очереди *Entnum*.
- *QTEntnum* – **среднее время ожидания в очереди:** взвешенное среднее время ожидания транзактов в очереди *Entnum*.
- *QXEntnum* – **среднее время ожидания в очереди без учета транзактов с нулевым временем ожидания:** взвешенное среднее время ожидания транзактов в очереди *Entnum*, не считая транзактов со временем ожидания, равным нулю.
- *QZEntnum* – **количество транзактов, прошедших через очередь с нулевым временем ожидания:** количество транзактов, время ожидания которых в очереди *Entnum* равно нулю.

6. Логические ключи

Логические ключи – самые простые объекты, которые могут находиться только в двух состояниях: «установлен» (1) или «сброшен» (0). Существуют блоки, которые изменяют состояние логических ключей, и СЧА, которые возвращают состояние логических ключей.

Логический ключ сбрасывается (0) после его создания, или когда объекту "Процесс моделирования» передается команда CLEAR (без опции OFF).

С логическим ключом связан блок:

- LOGIC – устанавливает, сбрасывает или инвертирует логический ключ.

СЧА логических ключей:

- *LSEntnum* – **состояние логического ключа:** 1, если логический ключ *Entnum* установлен, и 0 – в противном случае..

7. Функции

В GPSS функции возвращают значения, вычисленные по некоторому аргументу, например, случайному числу. Фактически, в качестве аргумента может использоваться любой СЧА. Функция определяется оператором описания FUNCTION, за которым следует один или более списков данных функции. Операнд А оператора FUNCTION определяет аргумент, а операнд В - тип функции и число пар данных. Пары данных - это числа, имена и/или СЧА, содержащиеся в списках данных функции, которые завершают определение

функции.

Для многих задач более удобно использовать PLUS-процедуры, чем функции. Однако функции GPSS лучше подходят для использования в случае применения их в качестве списковых функций и эмпирических вероятностных распределений. Кроме того, функция GPSS может требовать для вычисления меньше машинного времени, чем эквивалентная PLUS-процедура.

В GPSS можно использовать 5 различных типов функций:

- Тип С – Непрерывная функция. Значение функции вычисляется методом линейной интерполяции. Частный случай - использование случайного аргумента.
- Тип D - Дискретная функция. Каждому значению аргумента или вероятностной мере задается отдельное значение. Частный случай – использование случайного аргумента.
- Тип Е – Дискретная «атрибутивно-значимая» функция. Каждому значению аргумента или вероятностной мере задается вычисляемое значение СЧА. Частный случай – использование случайного аргумента.
- Тип L - Списковая числовая функция. Значение аргумента используется для определения позиции возвращаемой величины в списке.
- Тип М – Списковая «атрибутивно-значимая» функция. Значение аргумента используется для определения позиции СЧА в списке. Этот СЧА вычисляется и возвращается в качестве результата функции.

Функция, задействованная в операнде В блоков ADVANCE или GENERATE, называется «модификатор функции». Значение функции, вычисленное как число с плавающей точкой двойной точности, умножается на значение операнда А. Затем результат используется как требуемая временная задержка в данном блоке.

Значения, заданные в объявлении функции, например, в списке данных функции, хранятся в форме чисел с плавающей точкой двойной точности. Эти значения имеют точность, ограниченную 15десятичными разрядами, и величину, ограниченную 306 десятичными разрядами. Когда производится вычисление значения функции методом линейной интерполяции в качестве аргумента используется случайное число, значение аргумента вычисляется с помощью заданного генератора случайных чисел и находится в диапазоне от 0 до 0,999999. это случайное число умножается на коэффициент интерполяции и прибавляется к основанию интервала.

СЧА функций:

- *FCEntnum* – Функция. Результат вычисления функции *Entnum*

8. Таблицы

Таблица – это набор целых чисел, используемый для сбора данных для гистограммы. Каждое целое число представляет частотный класс в гистограмме.

Таблица определяется оператором описания TABLE.

С таблицами связан блок:

- TABULATE – обновляет данные гистограммы, накопленные в таблице.

СЧА таблиц:

- *TBEntnum* – Среднее значение невзвешенных аргументов таблицы *Entnum*.
- *TCEntnum* – Количество невзвешенных аргументов таблицы *Entnum*.
- *TDEntnum* – Стандартное отклонение невзвешенных аргументов таблицы *Entnum*.

9. Переменные

Переменные представляют собой выражения, которые при необходимости могут быть вычислены. Все переменные должны быть определены выражениями, которые включают константы, СЧА, библиотечные арифметические функции, арифметические и логические операторы.

Переменные определяются командами VARIABLE, FVARIABLE или BVARIABLE.

- VARIABLE – создает арифметическую переменную GPSS, которая при вызове соответствующего СЧА вычисляет выражение и возвращает конечный результат.
- FVARIABLE – создает арифметическую переменную GPSS с плавающей точкой, которая при вызове СЧА вычисляет выражение и возвращает конечный результат.
- BVARIABLE – создает булеву переменную GPSS, которая при вызове СЧА вычисляет выражение и затем возвращает 1, если результат отличен от нуля, в противном случае – 0. Булевы переменные возвращают 1, если результат «истина», и 0, если результат «ложь».

СЧА переменных:

- BVEntnum – Результат вычисления булевой переменной Entnum.
- VEntnum – Результат вычисления арифметической переменной или переменной с плавающей точкой Entnum.

10. Генераторы случайных чисел

В GPSS World потоки случайных чисел генерируются по 32-битному мультипликативному конгруэнтному алгоритму с максимальным периодом. Период равняется $2^{31}-2$ и не включает 0. Вы можете включать в модель любое количество генераторов случайных чисел без их объявления. Начальное число генератора случайных чисел совпадает с номером объекта генератора случайных чисел. Следует учесть, что операторами RMULT могут устанавливаться только генераторы случайных чисел с номерами 1-7.

Алгоритм генерации псевдослучайных чисел в GPSS World основан на мультипликативно-конгруэнтном алгоритме Лемера с максимальным периодом. Алгоритм генерирует псевдослучайные числа в открытом интервале от 0 до 2,147,483,647 и до самоповторения генерирует 2,147,483,646 уникальных псевдослучайных чисел. Дополнительно в GPSS World используется шаг перемешивания. СЧА класса RN возвращает число от 0 до 999 включительно, а при вычислении случайных функций используется случайное число, лежащее в интервале от 0 до 0.999999 включительно.

Важными атрибутами генераторов случайных чисел являются:

- **Начальные числа.** До тех пор, пока оно не будет изменено оператором RMULT, начальное число равно номеру объекта генератора случайных чисел. Например, RN25 запускается с начальным числом 25.
- **Применение в системе.** GPSS World использует генераторы случайных чисел для разрешения временных узлов в блоках TRANSFER, работающих в режиме статистической передачи, и для обеспечения блоков GENERATE и ADVANCE случайными числами. Номер генератора случайных чисел указывается на странице «Random Numbers» («Случайное число») журнала настроек модели.
- **Значения СЧА.** При помощи СЧА можно получить случайное целое число в диапазоне 0-999. Для определения новых переменных вы можете строить большие случайные числа, применяя такие выражения, как $1000\#RN2+RN2$. Вновь

определенные случайные числа возвращаются с помощью вызовов СЧА класса V. Во многих случаях целесообразно применение встроенных вероятностных распределений из библиотеки процедур.

- **Интерполяционные значения.** При использовании интерполяции в непрерывной случайной функции из потока случайных чисел извлекаются дробные значения в диапазоне 0 - 0.999999.

СЧА генераторов случайных чисел:

- **RNEntnum –случайное целое число** в диапазоне 0 - 999 из генератора случайных чисел *Entnum*.

Вероятностные распределения в GPSS World

Встроенная библиотека процедур GPSS содержит множество вероятностных распределений случайных величин, в том числе:

- равномерное;
- экспоненциальное;
- Парето;
- геометрическое,
- Пуассона;
- нормальное;
- гамма и др.

Равномерное распределение

Uniform (G, a, b)

Возвращает *вещественное* значение случайной величины, распределенной по равномерному закону.

G (от 1 до 999) – номер генератора равномерно распределенных случайных чисел;

a – минимальное значение случайной величины – левая граница распределения;

b – максимальное значение случайной величины – правая граница распределения, причем $a < b$.

Плотность распределения, математическое ожидание и дисперсия:

$$f(x) = \frac{1}{b-a} \quad (a \leq x \leq b), \quad M[X] = \frac{a+b}{2}, \quad D[X] = \frac{(b-a)^2}{12}.$$

Экспоненциальное распределение

Exponential (G, s, p)

Возвращает *вещественное* значение случайной величины, распределенной по экспоненциальному закону.

G (от 1 до 999) – номер генератора равномерно распределенных случайных чисел;

s – сдвиг (смещение) распределения относительно оси ординат;

$p \geq 0$ – величина сжатия распределения, определяемая как математическое ожидание случайной величины минус смещение.

Плотность распределения, математическое ожидание и дисперсия:

$$f(x) = \frac{1}{p} e^{-\frac{x-s}{p}}, \quad M[X] = p + s, \quad D[X] = p^2.$$

Распределение Парето

Pareto (G, k, a)

Возвращает *вещественное* значение случайной величины, распределенной по закону Парето.

G (от 1 до 999) – номер генератора равномерно распределенных случайных чисел;
k – сдвиг (смещение) распределения относительно оси ординат;
a ≥ 0 – величина сжатия распределения.

Плотность распределения, математическое ожидание и дисперсия:

$$f(x) = \frac{ak^a}{x^{a+1}} \quad (x > k, a > 0), \quad M[X] = \frac{a}{a-1}k \quad (a > 1),$$

$$D[X] = \frac{a}{(a-2)(a-1)^2}k^2 \quad (a > 2)$$

Геометрическое распределение

Geometric (G, p)

Возвращает *целочисленное* значение случайной величины, распределенной по геометрическому закону.

G (от 1 до 999) – номер генератора равномерно распределенных случайных чисел;
p (0 ≤ p ≤ 1) – вероятность успеха в испытаниях Бернулли.

Закон распределения, математическое ожидание и дисперсия:

$$f(k) = p(1-p)^k \quad (k = 0, 1, 2, \dots), \quad M[k] = \frac{1-p}{p}, \quad D[k] = \frac{1-p}{p^2}.$$

Распределение Пуассона

Poisson (G, a)

Возвращает *целочисленное* значение случайной величины, распределенной по закону Пуассона.

G (от 1 до 999) – номер генератора равномерно распределенных случайных чисел;
a – математическое ожидание (среднее значение) случайной величины.

Закон распределения, математическое ожидание и дисперсия:

$$f(k) = \frac{a^k}{k!} e^{-a} \quad (k = 0, 1, 2, \dots), \quad M[k] = a, \quad D[k] = a.$$