**Итоговый тест по дисциплине**

**"Имитационное моделирование дискретных процессов"**

(Примеры вопросов для тестирования)

1. **Имитационное моделирование - это ...**

*Имитационное моделирование* **(ИМ)** – это построение имитационной модели, основанное на изучении элементов системы, описании их функционирования и выявлении разнообразных связей и отношений между элементами. Здесь используется тот факт, что описание функционирования отдельного элемента значительно проще описания функционирования всей системы в целом.

В результате, ИМ есть процесс конструирования модели реальной системы и постановки экспериментов на этой модели с целью либо понять поведение системы, либо оценить различные стратегии, обеспечивающие функционирование данной системы.

Метод исследования, базирующийся на разработке и использовании моделей, называется моделированием. Необходимость моделирования обусловлена сложностью, а порой и невозможностью прямого изучения реального объекта (процесса). Значительно доступнее создавать и изучать прообразы реальных объектов (процессов), т.е. модели.

ИМ применимо в любой отрасли науки. Его применяют в технике, в системах обработки информации, коммерческой деятельности, экономи, на транспорте, в исследовании городов и глобальных систем, а также во многих других областях.

Имитационная модель выполняет еще одну важнейшую функцию: она выступает как средство общения.



1. **Как продвигается время в дискретно-событийных моделях?**



Три подхода ИМ

*Дискретно-событийное моделирование* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *discrete-event simulation*, DES) — это вид [имитационного моделирования](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). В дискретно-событийном моделировании функционирование системы представляется как хронологическая последовательность событий. Событие происходит в определенный момент времени и знаменует собой изменение состояния системы.

Динамическая природа дискретно-событийных имитационных моделей требует, чтобы мы следили за текущим значением имитационного времени по мере функционирования имитационной модели. Нам необходим также механизм для продвижения имитационного времени от одного значения к другому. В имитационной модели переменная, обеспечивающая текущее значение модельного времени, называется часами модельного времени. При создании модели на таких универсальных языках, как FORTRAN или С, единица времени для часов модельного времени никогда не устанавливается явно. Подразумевается, что оно будет указываться в тех же единицах, что и входные параметры. К тому же модельное время и время, необходимое для прогона имитационной модели на компьютере, как правило, невозможно соотнести.

Существует два основных подхода к продвижению модельного времени: продвижение времени от события к событию и продвижение времени с постоянным шагом. Поскольку первый подход используется всеми основными имитационными программами и большинством разработчиков, создающих свои модели на универсальных языках, а также с учетом того, что второй подход является разновидностью первого, в дискретно-событийных моделях выберем такой подход, как продвижение времени от события к событию.

При использовании продвижения времени от события к событию часы модельного времени в исходном состоянии устанавливаются в 0 и определяется время возникновения будущих событий. После этого часы модельного времени переходят на время возникновения ближайшего события, и в этот момент обновляются состояние системы с учетом произошедшего события, а также сведения о времени возникновения будущих событий. Затем часы модельного времени продвигаются ко времени возникновения следующего (нового) ближайшего события, обновляется состояние системы и определяется время будущих событий, и т. д.

Процесс продвижения модельного времени от времени возникновения одного события ко времени возникновения другого продолжается до тех пор, пока не будет выполнено какое-либо условие останова, указанное заранее. Поскольку в дискретно-событийной имитационной модели все изменения происходят только во время возникновения событий, периоды бездействия системы просто пропускаются, и часы переводятся со времени возникновения одного события на время возникновения другого. Следует отметить, что длительность интервала продвижения модельного времени от одного события к другому может быть различной.

1. **Из каких основных этапов состоит исследование с имитационной моделью?**

|  |
| --- |
| В процессе имитационного моделирования можно выделить следующие основные этапы:  1. Формирование проблемы : описание исследуемой проблемы и определение целей исследования.  2. Разработка модели: логико-математическое описание моделируемой системы в соответствии с формулировкой проблемы.  3. Подготовка данных: идентификация, спецификация и сбор данных.  4. Трансляция модели: перевод модели на язык, приемлемый для используемой ЭВМ.  5. Верификация: установление правильности машинных программ.  6. Валидация: оценка требуемой точности и соответствие имитационной модели реальной системе.  7. Стратегическое и тактическое планирование: определение условий проведения машинного эксперимента с имитационной моделью.  8. Экспериментирование: прогон имитационной модели на ЭВМ для получения требуемой информации.  9. Анализ результатов: изучение результатов имитационного эксперимента для подготовки выводов и рекомендаций по решению проблемы.  10. Реализация и документирование: реализация рекомендаций, полученных на основе имитации, составление документации по модели и ее использованию.  Рассмотрим основные этапы имитационного моделирования. Первой задачей имитационного исследования является точное определение проблемы и детальная формулировка целей исследования. Как правило, определение проблемы является непрерывным процессом , который обычно осуществляется в течении всего исследования. Оно пересматривается по мере более глубокого понимания исследуемой проблемы и возникновения новых ее аспектов.  Как только сформулировано начальное определение проблемы, начинается этап построения модели исследуемой системы. Модель включает статистическое и динамическое описание системы. В статистическом описании определяются элементы системы и их характеристики, а в динамическом- взаимодействие элементов системы, в результате которых происходит изменение ее состояния во времени.  Процесс формирования модели во многом является искусством. Разработчик модели должен понять структуру системы, выявить правила ее функционирования и суметь выделить в них самое существенное, исключив ненужные детали. Модель должна быть простой для понимания и в то же время достаточно сложной, чтобы реалистично отображать характерные черты реальной системы.  Наиболее важными являются принимаемые разработчиком решения относительно того, верны ли принятые упрощения и допущения, какие элементы и взаимодействия между ними должны быть включены в модель. Уровень детализации модели зависит от целей ее создания. Необходимо рассматривать только те элементы, которые имеют существенное значение для решения исследуемой проблемы.  Как на этапе формирования проблемы, так и на этапе моделирования необходимо тесное взаимодействие между разработчиком модели и ее пользователями. Кроме того, тесное взаимодействие на этапах формулирования проблемы и разработки модели создает у пользователя уверенность в правильности модели, поэтому помогает обеспечить успешную реализацию результатов имитационного исследования.  На этапе разработки модели определяются требования к входным данным. Некоторые из этих данных могут уже быть в распоряжении разработчика модели, в то время как для сбора других потребуется время и усилия. Обычно значение таких входных данных задаются на основе некоторых гипотез или предварительного анализа.  В некоторых случаях точные значения одного (и более) входных параметров оказывают небольшое влияние на результаты прогонов модели. Чувствительность получаемых результатов к изменению входных данных может быть оценена путем проведения серии имитационных прогонов для различных значений входных параметров. Имитационная модель, следовательно, может использоваться для уменьшения затрат времени и средств на уточнение входных данных.  После того как разработана модель и собраны начальные входные данные, следующей задачей является перевод модели в форму, доступную для компьютера.  На этапах верификации и валидации осуществляется оценка функционирования имитационной модели. На этапе верификации определяется, соответствует ли запрограммированная для ЭВМ модель замыслу разработчика. Это обычно осуществляется путем ручной проверки вычисления, а также может быть использован и ряд статистических методов.  Установление адекватности имитационной модели исследуемой системы осуществляется на этапе валидации. Валидация модели обычно выполняется на различных уровнях. Специальные методы валидации включают установление адекватности путем использования постоянных значений всех параметров имитационной модели или путем оценивания чувствительности выходов к изменению значений входных данных. В процессе валидации сравнение должно осуществляться на основе анализа как реальных, так и экспериментальных данных о функционировании системы.  Условия проведения машинных прогонов модели определяется на этапах стратегического и тактического планирования.  Задача стратегического планирования заключается в разработке эффективного плана эксперимента, в результате которого выясняется взаимосвязь между управляемыми переменными, либо находится комбинация значений управляемых переменных, минимизация или максимизация имитационной модели.  В тактическом планировании в отличии от стратегического решается вопрос о том, как в рамках плана эксперимента провести каждый имитационный прогон, чтобы получить наибольшее количество информации из выходных данных. Важное место в тактическом планировании занимают определение условий имитационных прогонов и методы снижения дисперсии среднего значения отклика модели.  Следующие этапы в процессе имитационного исследования- проведение машинного эксперимента и анализ результатов- включают прогон имитационной модели на ЭВМ и интерпретацию полученных выходных данных.  Последним этапом имитационного исследования является реализация полученных решений и документирование имитационной модели и ее использование.  Ни одни из имитационных проектов не должен считаться законченным до тех пор, пока их результаты не были использованы в процессе принятия решений. Успех реализации во многом зависит от того, насколько правильно разработчик модели выполнил все предыдущие этапы процессов имитационного исследования. Если разработчик и пользователь работали в тесном контакте и достигли взаимопонимания при разработке модели и ее исследовании, то результат проекта скорее всего будет успешно внедряться. Если же между ними не было тесной взаимосвязи, то, несмотря на элегантность и адекватность имитационного моделирования, сложно будет разработать эффективные рекомендации.  Вышеперечисленные этапы редко выполняются в строго заданной последовательности, начиная с определения проблемы и кончая документированием.  В ходе имитационного моделирования могут быть сбои в прогонах модели, ошибочные допущения, от которых в дальнейшем приходится отказываться, переориентировки целей исследования, повторные оценки и перестройки модели. Такой процесс позволяет разработать имитационную модель, которая дает верную оценку альтернатив и облегчает процесс принятия решений. |
|  |

1. **Что значит "квазипараллельный алгоритм моделирования"?**

С точки зрения внешнего наблюдателя, в хорошей многозадачной ОС происходит одновременная, параллельная работа нескольких процессов. Однако понятно, что эта одновременность кажущаяся. На самом деле, если в системе работает лишь один процессор, то в каждый момент времени он выполняет команды, относящиеся только к одному из имеющихся процессов. Иллюзия параллельности создается за счет того, что процессы сменяют друг друга через малые интервалы времени, которые человек-наблюдатель не в силах отследить. Подобная организация работы называется ***квазипараллельным***выполнением процессов.

Разумеется, если в системе имеется несколько процессоров, то может быть организовано настоящее параллельное выполнение процессов, количество которых не превышает количества процессоров. При большем числе процессов может использоваться смешанная организация, сочетающая истинную параллельность и квазипараллельность.

Важно отметить, что для большинства задач взаимодействия процессов нет разницы, какого рода параллельность используется в данной ОС. Вообще, основные проблемы управления процессами можно разбить на два уровня:

* проблемы корректной и эффективной реализации параллельного (т.е. обычно квазипараллельного) выполнения процессов – это проблемы нижнего уровня;
* проблемы корректного взаимодействия параллельных процессов – это проблемы верхнего уровня, при рассмотрении которых считается, что низкоуровневые проблемы реализации процессов так или иначе решены.

Такое разбиение облегчает проектирование и отладку систем, а также позволяет лучше понять существо рассматриваемых проблем

1. Что значит распределенное имитационное моделирование?

Если предположить, что предметом исследования системы имитационного моделирования является корпоративная информационная система, охватывающая территориально распределенные офисы и/или производственные объекты, то возникает необходимость в исследовании аппаратуры (а это множество компьютеров, объединенных в локальные и глобальные сети), информационных потоков, бизнес-процессов, цепочек поставок и т.д.

Таким образом, система моделирования должна уметь оперировать с большим количеством объектов и процессов. Эти системы моделирования требуют больших вычислительных ресурсов. Целесообразно для таких имитационных экспериментов использовать многопроцессорные ЭВМ, кластеры или компьютерные сети. Системы моделирования, использующие многопроцессорные ЭВМ и компьютерные сети, называют параллельными или распределенными. В настоящее время устоявшимся термином является термин "распределенное моделирование", в дальнейшем будем использовать этот термин. Рассмотрим более подробно причины появления распределенного моделирования.

Использование распределенного моделирования объясняется:

- возможностью использования вычислительных ресурсов нескольких процессоров (компьютеров) для выполнения имитационного эксперимента (предполагается. распределение компонентов имитационной модели по процессорам (компьютерам) и совместное их участие в имитационном эксперименте) с целью повышения производительности систем имитационного моделирования;

- использованием локальной памяти других процессоров (компьютеров);

- одновременным запуском нескольких репликаций параллельно на нескольких компьютерах (снижение временных затрат на эксперимент);

- объединением уже готовых имитационных моделей и их участием в совместном имитационном эксперименте (переиспользование кода).

- возможностью участия географически удаленных друг от друга пользователей в работе над одним имитационным проектом, возможность разработки модели, запуска имитационного эксперимента и одновременного наблюдения за выполнением разработанной модели,

- повышение надежности при выполнении имитационного эксперимента, поскольку при выходе из строя процессора или компьютера, на котором выполняется один из компонентов имитационной модели, выполнение его может быть продолжено на другом процессоре (компьютере).

1. Какие системы имитационного моделирования используются в современных исследованиях?
   1. GPSS Studio
   2. AnyLogic
   3. SIMIO
   4. MatLab (+Simulink)
   5. ProModel
   6. Arena

<https://studref.com/657268/informatika/obzor_sistem_imitatsionnogo_modelirovaniya>

1. Какие программные библиотеки для имитационных моделей используются в современных разработках?

pysim...

1. Что такое имитационный компьютерный эксперимент?



<http://kursak.net/planirovanie-kompyuternyx-eksperimentov-s-modelyami-sistem/>

1. Что является основным критерием проверки адекватности модели для заказчика?

АДЕКВАТНОСТЬ МОДЕЛИ — соответствие модели моделируемому объекту или процессу. Адекватность — в какой-то мере условное понятие, так как полного соответствия модели реальному объекту быть не может, иначе это была бы не модель, а сам объект. При моделировании имеется в виду адекватность не вообще, а по тем свойствам модели, которые для исследования считаются существенными.

Модель считается адекватной, если отражает заданные свойства с приемлемой точностью. Точность определяется как степень совпадения значений выходных параметров модели и объекта.

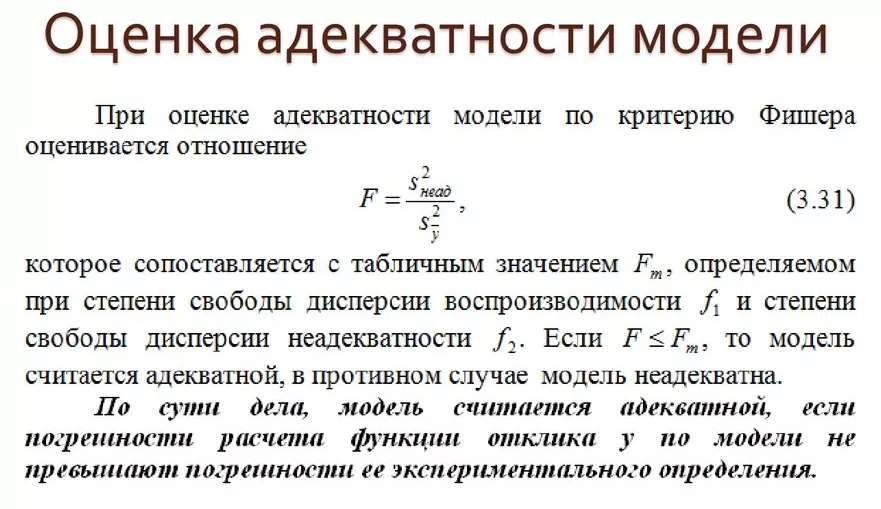
Точность модели различна в разных условиях функционирования объекта. Эти условия характеризуются внешними параметрами. В пространстве внешних параметров выделить область адекватности модели, где погрешность меньше заданной предельно допустимой погрешности. Определение области адекватности моделей - сложная процедура, требующая больших вычислительных затрат, которые быстро растут с увеличением размерности пространства внешних параметров. Эта задача по объему может значительно превосходить задачу параметрической оптимизации самой модели, поэтому для вновь проектируемых объектов может не решаться.

Точность математической модели оценивается степенью совпадения значений выходных параметров реального объекта и значений тех же параметров, рассчитанных с помощью модели.

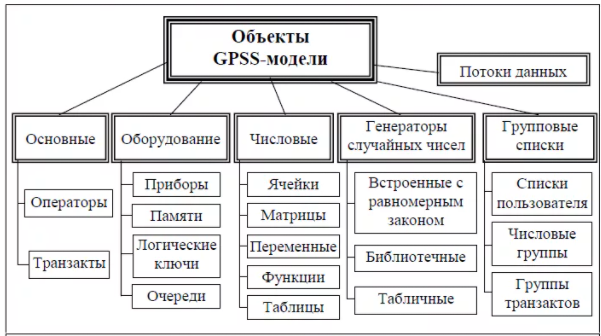
Адекватность математической модели - это ее способность отражать заданные свойства объекта с погрешностью, не выше заданной.

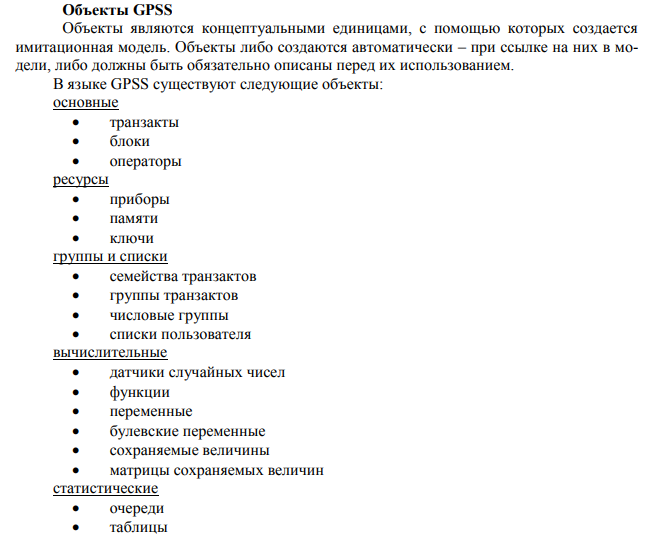
Верификация — проверка, проверяемость, способ подтверждения каких-либо теоретических положений, алгоритмов, программ и процедур путем их сопоставления с опытными (эталонными или эмпирическими) данными, алгоритмами и программами.

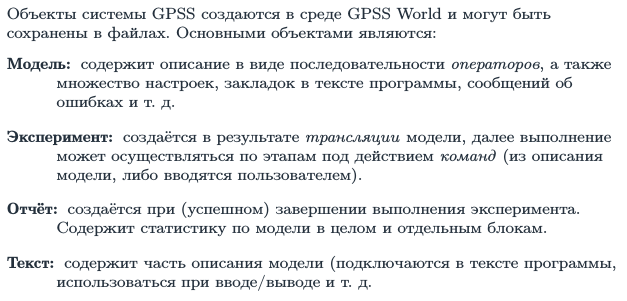
Верификация — это подтверждение соответствия конечного продукта Проверка истинности теоретических положений, установление их достоверности предопределённым эталонным требованиям логико-методологическая процедура установления истинности научной на основе их соответствия эмпирическим данным или теоретическим положениям, соответствующим эмпирическим данным. В рамках логического позитивизма принцип верифицируемости мыслится критериально исчерпывающим способом апробации научных утверждений, понятых в качестве "протокольных предположений" как фиксаций данных непосредственного опыта: утверждения, выходящие за рамки "протокольных предложений" трактуются как неверифицируемые, в случае чего в действие вступает принцип фальсифицируемости.



1. Что относится к модельным объектам GPSS World ?



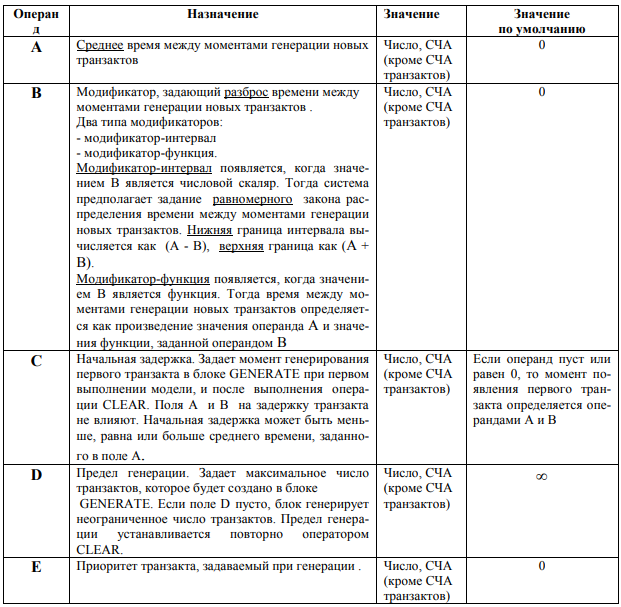


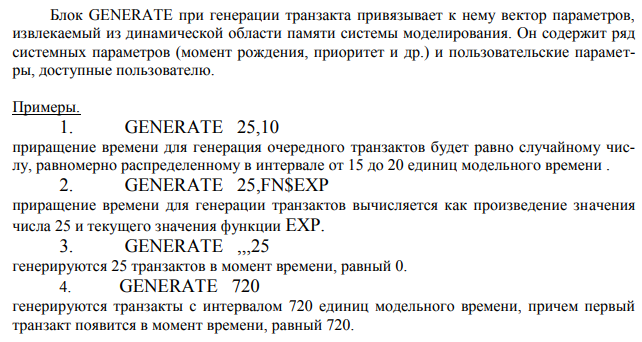


1. Какие блоки создают транзакты в GPSS ?

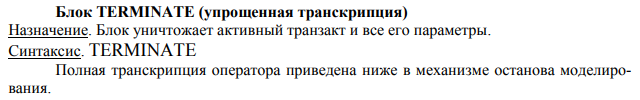
Блок GENERATE

Назначение. Блок генерирует транзакты и отправляет их вниз под себя на начало трека. Синтаксис. GENERATE [A],[B],[C],[D],[E]

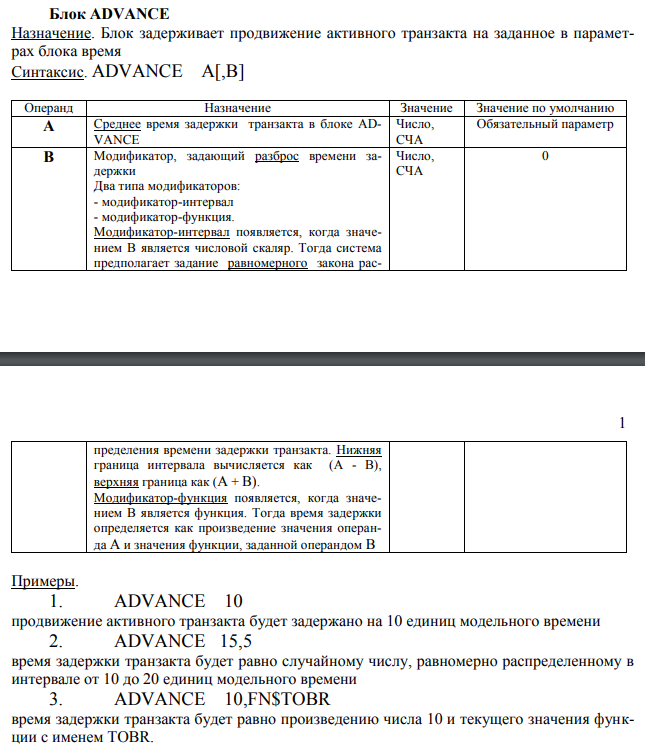




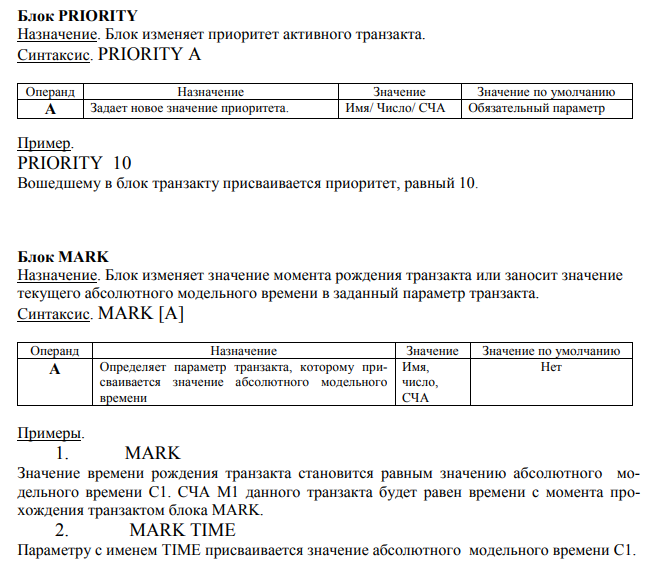
1. Какие блоки уничтожают транзакты в GPSS?

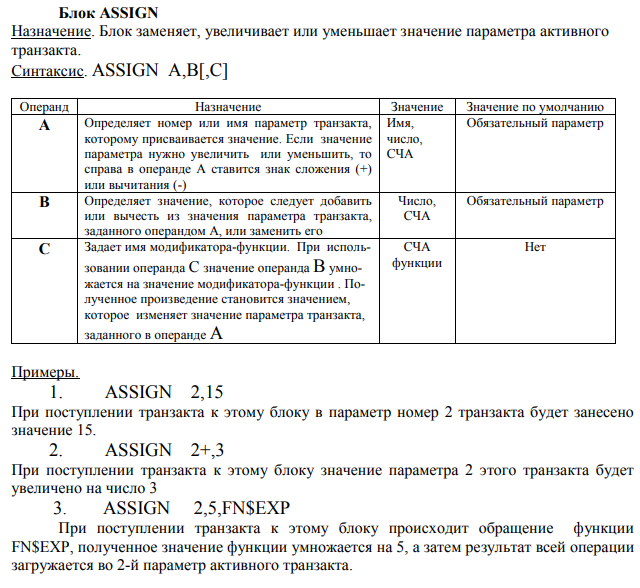


1. Какие блоки могут задерживать продвижение транзактов в модели GPSS?

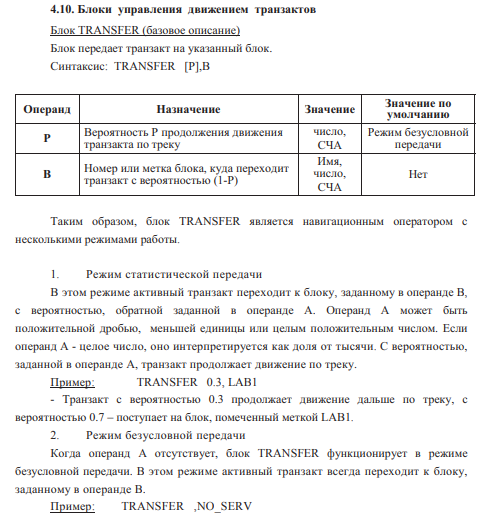


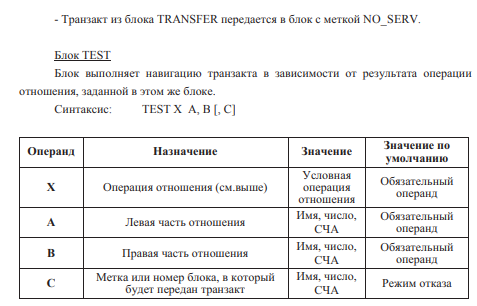
1. Какие блоки изменяют параметры локальной среды транзактов GPSS?

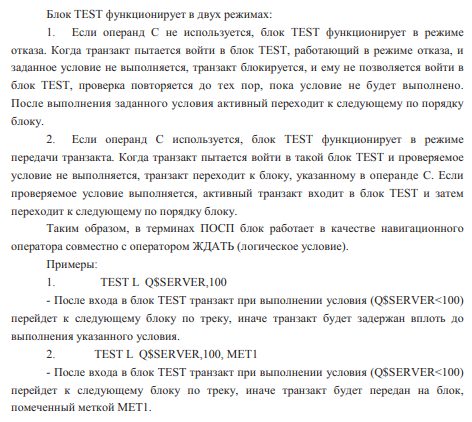


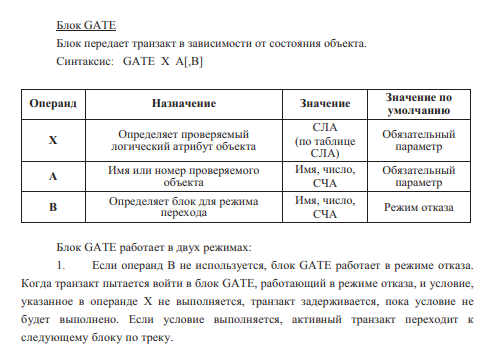


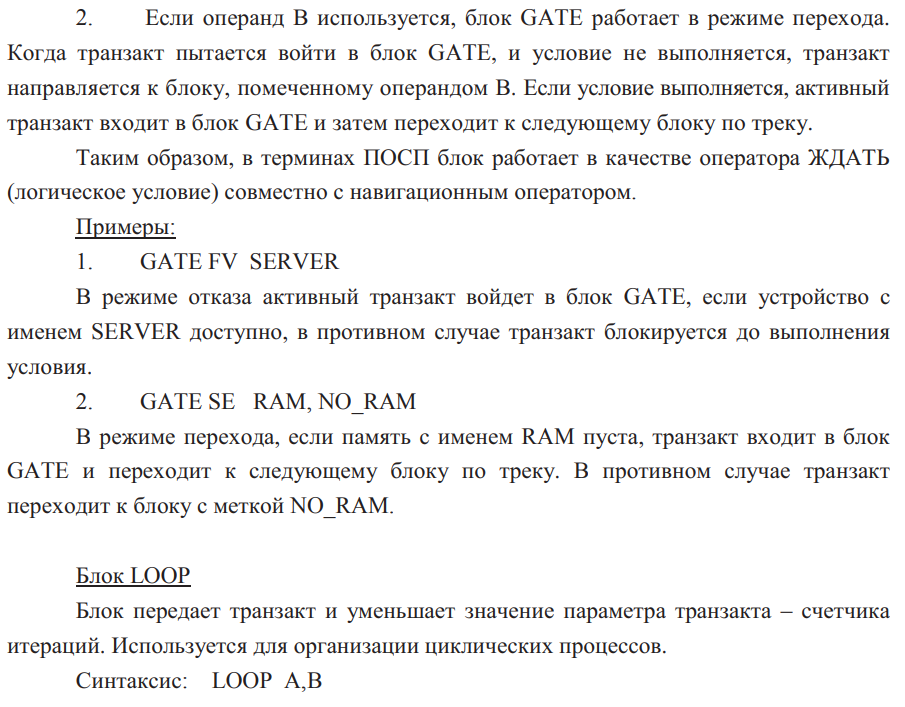
1. Какие блоки управляют маршрутизацией транзактов GPSS?

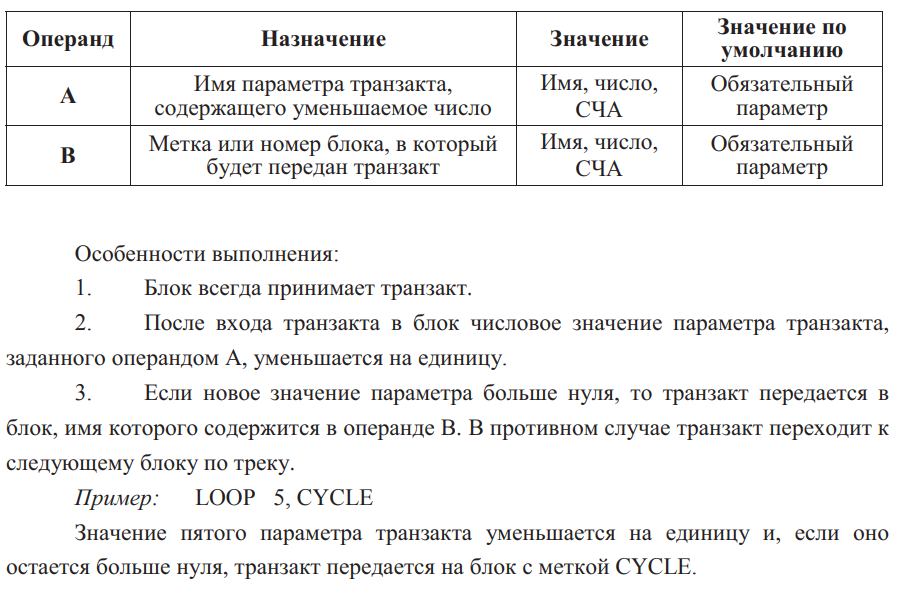




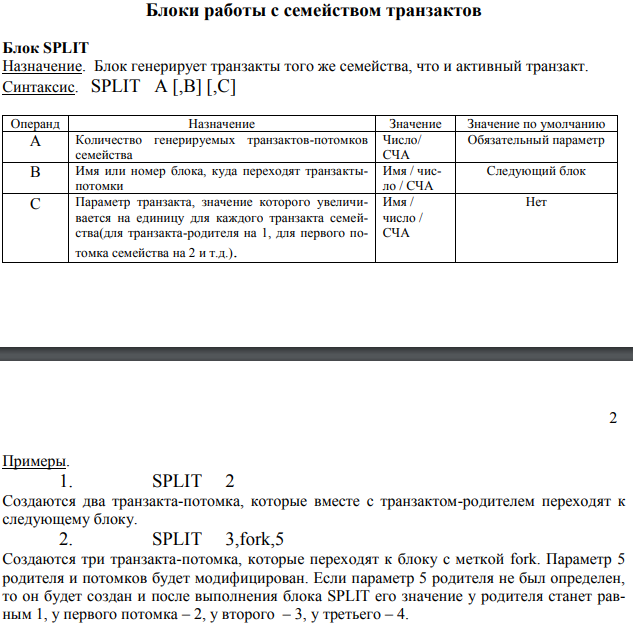


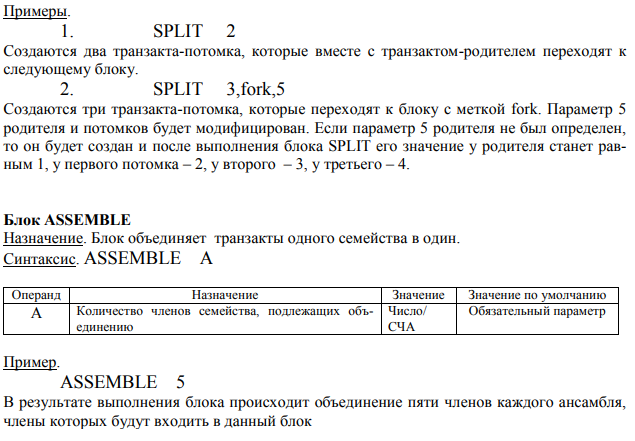


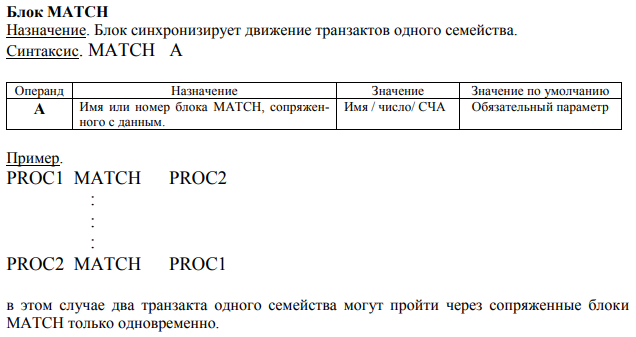




1. Какие блоки используют атрибут семейства транзакта в GPSS?







1. Как называется атрибут временной отметки транзакта GPSS?

С1

1. Какой тип описателя имеет непрерывная функция GPSS?

<имя функции> FUNCTION A,BN

Имя функции – числовое или символическое имя

А – аргумент функции. Допустимые значения – имя, число, СЧА

В – тип функции(одна буква) и N – количество пар данных в списке данных функции.

Обращение к функции выглядит как FN$I, где I – имя функции.

Функции типа С – непрерывные числовые функции.

Пример:

ART FUNCTION X1,C3

1.1,10.1/20.5,98.7/33.3,889.2

1. Какой тип описателя имеет дискретная функция GPSS?

Функции типа D – дискретные функции.

LIR FUNCTION X$A2,D5

1.1,6.9/2.1,7/6.33,9.4/7,10/9.9,12.01

1. Какой тип описателя имеет матричная функция в GPSS?
2. Какой тип аргумента может иметь непрерывная функция GPSS?

Допустимые значения аргумента: имя, число, СЧА.

Значения Х и Y должны быть целочисленными (Integer) или Вещественными (Real).

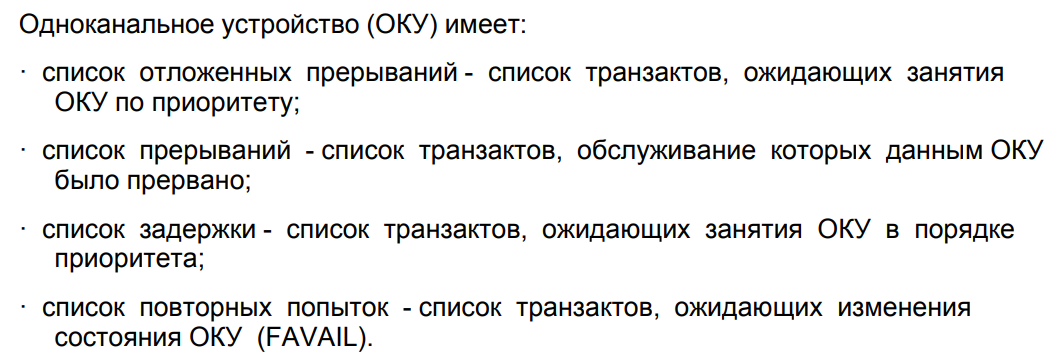
1. Какой тип аргумента может иметь дискретная функция GPSS?

Допустимые значения аргумента: имя, число, СЧА.

Значения Х должны быть целочисленными (Integer) или Вещественными (Real), а значения Y – целочисленными.

1. Какие блоки в GPSS изменяют состояние занятости одноканального устройства (прибора)?

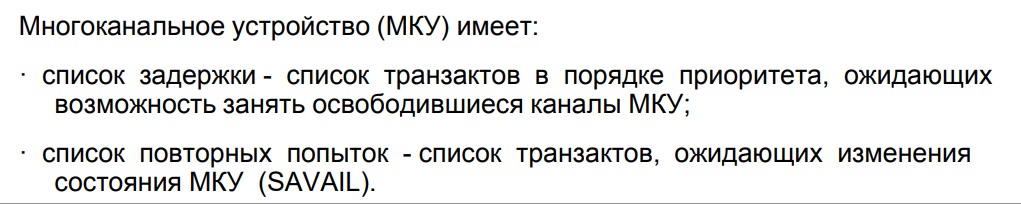
Seize, Release



1. Какие блоки GPSS изменяют состояние занятости многоканального устройства(памяти)?

Блок ENTER позволяет активному транзакту либо занять определенное число элементов памяти, либо встать в очередь к данной памяти.

Блок LEAVE позволяет активному транзакту освободить определенное число элементов памяти.



1. Перечислите блоки, используемые для имитации неисправности многоканальных устройств.

Блок SUNAVAIL переводит многоканальное устройство (а по существу накопитель) в состояние недоступности.

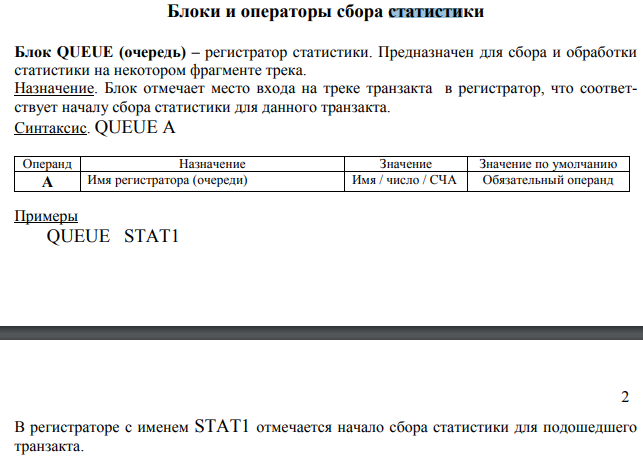
Блок SAVAIL переводит заданное многоканальное устройство из состояния недоступности в состояние доступности.

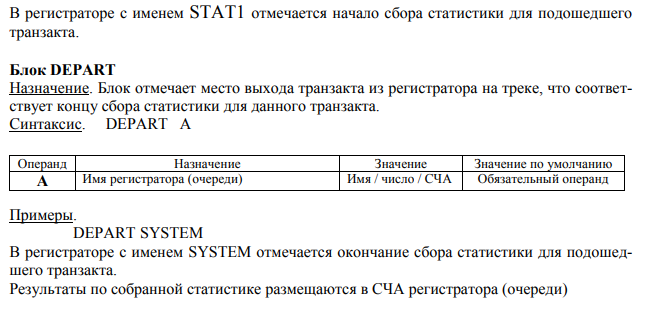
1. Перечислите блоки, используемые для имитации неисправности одноканальных устройств.

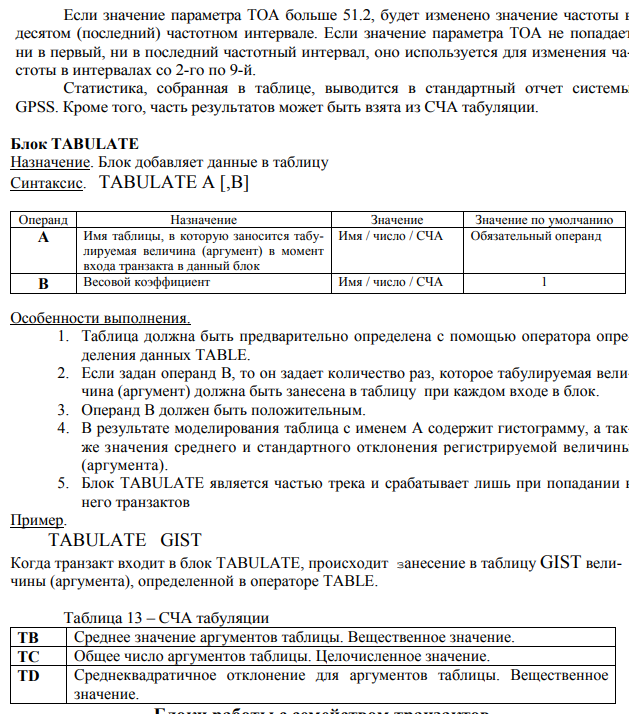
Блоком FUNAVAIL (символом F обозначает ОКУ, UNAVAIL - недоступный) моделируется недоступность ОКУ. При использовании этого блока статистика ОКУ не искажается.

Блок FAVAIL изменяет состояние ОКУ на доступное, т. е. восстанавливает обычный режим вхождения транзактов в ОКУ.

1. Какие блоки GPSS управляют статистикой очередей транзактов?

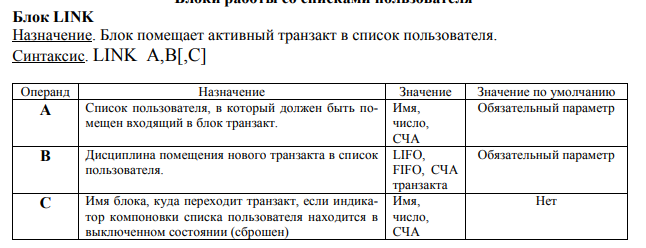


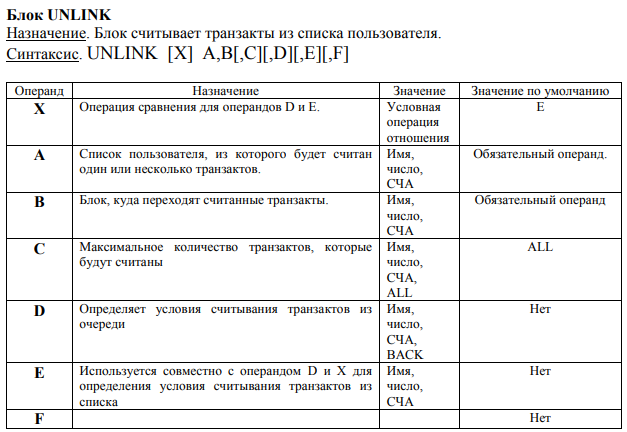




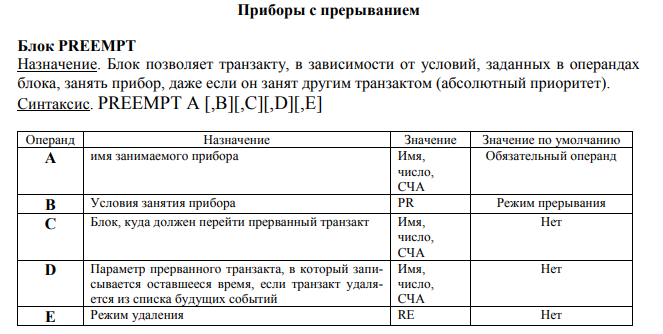
<https://infopedia.su/17x117af.html>

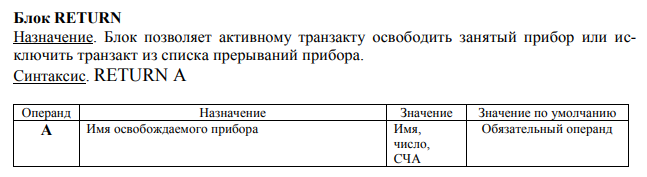
1. Какие блоки GPSS управляют пользовательскими списками транзактов?





1. Какие блоки GPSS изменяют состояние прерывания одноканального устройства (прибора)?





1. Какие блоки GPSS используют атрибут семейства транзакта?(см. вопрос 16)
2. Какие блоки GPSS изменяют значение общих параметров модели (ячеек, матриц)?

Для того чтобы в модели можно было организовать запоминание каких-либо данных, в языке GPSS имеются специальные СЧА, называемые ячейками сохраняемых величин или ячейками SAVEVALUE. Ячейка имеет обозначение вида Xj, где j – номер или имя ячейки.

Запись значений в ячейку осуществляется при входе транзакта в блок SAVEVALUE. Поле A этого блока интерпретируются как номер, косвенный номер или имя ячейки, поле В - как записываемое в нее значение.

Если в поле A записан СЧА, то его значение рассматривается как номер ячейки.

Пусть, например, транзакт проходит в модели через следующую цепочку блоков:

SAVEVALUE OPERATION\_NUMBER,157

SAVEVALUE 1,Х$OPERATION\_NUMBER

SAVEVALUЕ Х1,C1.

Тогда будут выполнятся следующие действия:

* запись числа 157 в ячейку Х$OPERATIONS\_NUMBER;
* запись числа 157 в ячейку 1;
* запись текущего времени в ячейку номер 157.

В начальный момент времени, когда в ячейки еще ничего не записывалось, все они имеют нулевое значение.

Если к обозначению СЧА в поле A блока SAVEVALUE приписать знак + или -, то в ячейке вместо записи значения будет производиться добавление или, соответственно, вычитание этого значения. Например, прохождение транзактом блоков

SAVEVALUE 11+,S1

SAVEVALUE 12-,V$DIFFERENCIAL

вызовет добавление числа занятых единиц памяти номер 1 в ячейку 11 и вычитание переменной DIFFERENCIAL из ячейки номер 12.

Наряду с ячейками в моделях на языке GPSS можно использовать матрицы ячеек. В отличие от простых ячеек матрицы перед использованием должны быть описаны. Для описания матрицы применяется строка описания матрицы. В поле метки этой строки записывается имя описываемой матрицы, в поле операции - слово MATRIX, в поле операндов - параметры матрицы: в поле A записывают любое слово или оставляют поле пустым, в поле В указывают число строк матрицы, в поле С - число столбцов. Например, описать матрицу ITOGI размерами 2 4 можно с помощью строки:

ITOGI MATRIX ,2,4.

В начальный момент любая матрица содержит только нулевые значения.

После того как матрицы описаны, можно использовать в качестве СЧА индексированные переменные - ячейки этих матриц. Ячейка матрицы имеет обозначение вида **MX$name(a,b)**, где name – имя матрицы , a - СЧА, значение которого задает номер строки матрицы, b - СЧА, задающий номер столбца. Например, обозначение MX$ITOGI(1,2) соответствует ячейке в первой строке и втором столбце матрицы ITOGI.

Запись значений в ячейки матриц осуществляется с помощью блоков MSAVEVALUE. Поле A этого блока задает имя матрицы, поле В - номер строки, поле С - номер столбца, поле D - записываемое значение.

Например, при входе транзакта в блок

MSAVEVALUE ITOGI,2,4,V$ABC

в матрицу ITOGI коротких ячеек (в ячейку, расположенную на пересечении второй строки и четвертого столбца) записывается значение переменной V$ABC.

В поле A блока MSAVEVALUE после имени матрицы может быть приписан знак + или - . В этом случае значение операнда D добавляется или соответственно вы-читается в ячейке матрицы, аналогично тому, как это делается в блоке SAVEVALUE.

Задание начальных значений ячеек и матриц

Начальные значения ячеек можно сделать отличными от нуля. Для этого применяется строка описания ячеек INITIAL. В поле A этой строки указывается СЧА, обозначающий ячейку или ячейку матрицы, в поле B – задаваемое начальное значение.

Например, строка

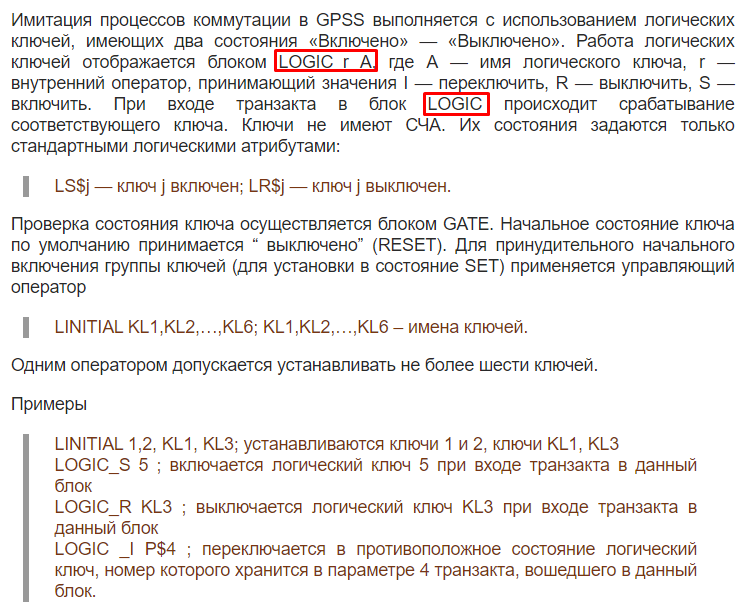
INITIAL X1,-10

установит в качестве начального значения ячейки номер 1 число –10, а с помощью строки

INITIAL MX$ITOGI(2,3),555

устанавливается начальное значение 555 в ячейку матрицы MX$ITOGI, расположенную во второй строке, третьем столбце.

1. Какой блок GPSS изменяет состояние логического переключателя?



1. Какой метод статистического анализа реализован в GPSS World?

однофакторный дисперсионный анализ

1. Какие команды управляют выполнением модели в GPSS World?

Команда START используется для инициации начала моделирования.

Команда RESET сбрасывает в ноль статистику и СЧА системы, но не удаляет транзакты из модели. Она используется для повторных экспериментов c моделью и сброса статистических данных переходного периода имитационного процесса.

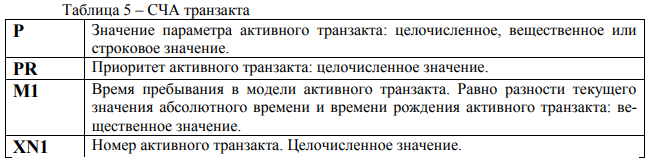
Команда CLEAR. Сбрасывает всю накопленную статистику, удаляет все транзакты из модели и устанавливает отсчет (нумерацию) транзактов, сгенерированных блоками GENERATE, начиная c единицы.

Команда EXIT (ВЫХОД) предназначена для завершения работы c системой GPSS World.

1. Какие команды GPSS World являются операторами описания параметров модели?

Оператор INITIAL позволяет задавать начальные значения сохраняемых величин, элементов матриц и логических ключей.

1. Что входит в состав стандартных параметров транзакта GPSS?



1. Что создает в модели GPSS блок вида GENERATE 1?

Назначение. Блок генерирует транзакты и отправляет их вниз под себя на начало трека.

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

1. Что в модели GPSS выполняет блок ASSIGN?

Назначение. Блок заменяет, увеличивает или уменьшает значение параметра активного транзакта.

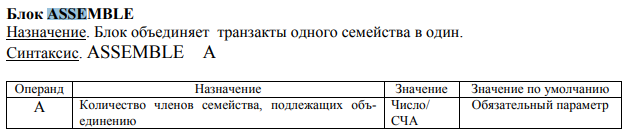
Синтаксис. ASSIGN A,B[,C]

1. Что в модели GPSS выполняет блок SPLIT?

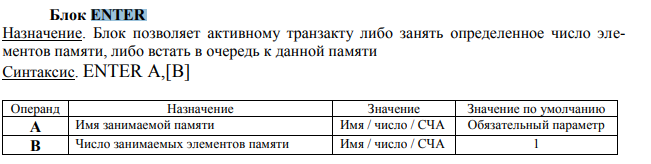
Назначение. Блок генерирует транзакты того же семейства, что и активный транзакт.

Синтаксис. SPLIT A [,B] [,C]

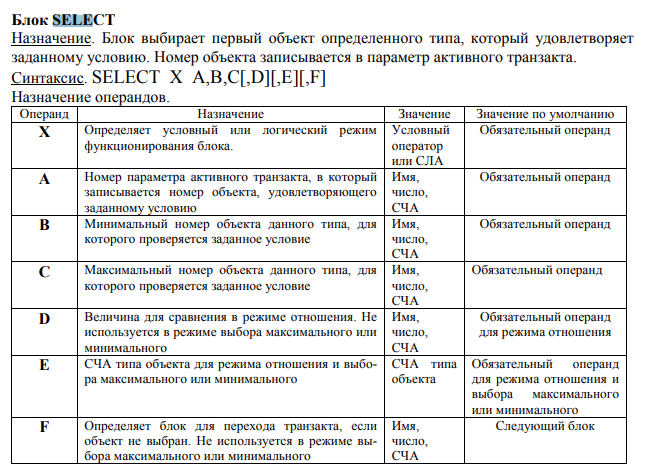
1. Что в модели GPSS выполняет блок ASSEMBLE?



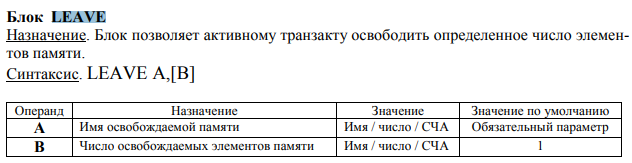
1. Что в модели GPSS выполняет блок ENTER ?



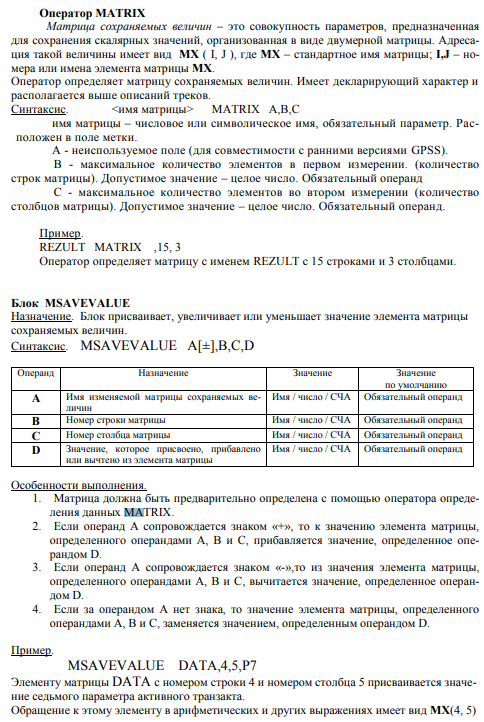
1. Что в модели GPSS выполняет блок SELECT ?



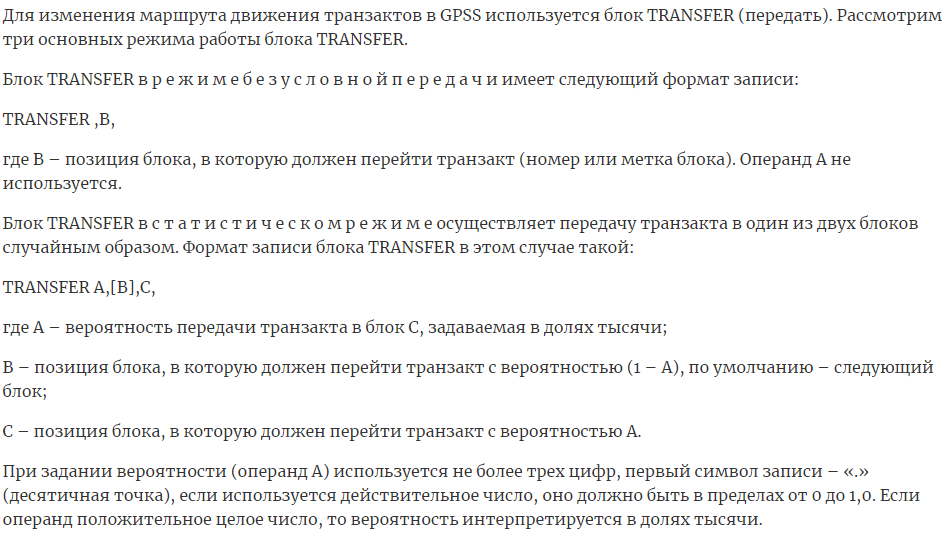
1. Что в GPSS выполняет блок LEAVE ?



1. Какой блок GPSS используется для записи значений параметров (атрибутов, переменных) в массив?



1. Укажите блоки языка GPSS, обеспечивающие передачу транзакта в блок отличный от последующего.



1. Kак правильно описать использование функции в GPSS?

Оператор FUNCTION определяет функцию GPSS, заданную таблично.

Синтаксис: <имя функции> FUNCTION A,BN

- А – аргумент функции; обязательный операнд. Допустимые значения – имя, число, СЧА.

- B – тип функции (одна буква) и N – количество пар данных в списке данных функции. Обязательный операнд.

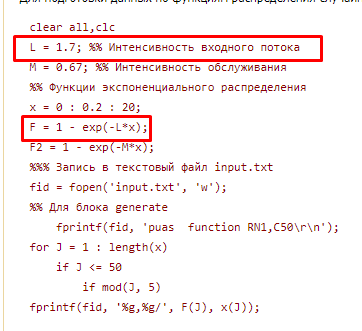
Обращение к функции выглядит, как FN I, где I – имя функции.

Существует несколько типов функций. Тип определяется операндом В оператора FUNCTION. За строкой FUNCTION А,ВN сразу же должна следовать строка, содержащая список пар данных, разделенных символом «/». Каждая пара данных определяет значения аргумента X и значения функции Y, разделенные запятой. Список данных используются для вычисления значения функции по заданным значениям аргумента.

<https://all4study.ru/gpss/funkcii-i-peremennye-v-gpss.html>

1. Какой раздел стандартного отчета GPSS содержит информацию из таблицы будущих времен?
2. В какие блоки GPSS нельзя направить транзакт?
3. В какой форме в GPSS World можно задать равномерно распределенное время задержки?
4. Для моделирования времени обслуживания, имеющего экспоненциальное распределение с параметром L=2, в GPSS используется оператор вида …

Exponential(1,0,65))



<https://intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21080?page=3>

1. Укажите последовательность блоков GPSS для настройки фрагмента, структурно реализующего учет очереди заданий к устройству-обработчику.

GENERATE  
QUEUE  
SEIZE  
DEPART  
ADVANCE  
RELEASE  
TERMINATE

1. Укажите последовательность блоков GPSS для настройки фрагмента, структурно реализующего очередь заданий с настраиваемой очередностью.

WAIT\_ST STORAGE 1000

GENERATE 20,,,,1

TRANSFER ,STARTPOINT

GENERATE 20,,,,2

TRANSFER ,STARTPOINT

GENERATE 20,,,,3

STARTPOINT ENTER WAIT\_ST

QUEUE WAIT

SEIZE A

DEPART WAIT

ADVANCE 9

RELEASE A

LEAVE WAIT\_ST

TEST E PR,3,STOPPOINT3

TEST E PR,2,STOPPOINT2

TEST E PR,1,STOPPOINT1

STOPPOINT3 TERMINATE 1

STOPPOINT2 TERMINATE 1

STOPPOINT1 TERMINATE 1

1. Укажите последовательность блоков GPSS для настройки фрагмента, структурно реализующего очередь заданий с ограниченным размером.

<https://www.rsdn.org/forum/other/2944849.hot>

Это делается с помощью TRANSFER BOTH + ENTER ... LEAVE.

**Re[2]: gpss ограничение очереди**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **От:** | [**zubrik**](https://www.rsdn.org/account/info/72955) |  |
| **Дата:** | 12.05.08 14:04 |

Здравствуйте, Mr.Cat, Вы писали:  
  
MC>Здравствуйте, zubrik, Вы писали:  
Z>>Как это записать в gpss? Особенно интересует до 7 заявок..  
  
MC>Это делается с помощью TRANSFER BOTH + ENTER ... LEAVE.  
  
Примеры таких задач есть?

... << RSDN@Home 1.2.0 alpha 4 rev. 1089>>

**Re[3]: gpss ограничение очереди**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **От:** | [**Mr.Cat**](https://www.rsdn.org/account/info/64543) |  |
| **Дата:** | 13.05.08 11:09 |

Здравствуйте, zubrik, Вы писали:  
Z>Примеры таких задач есть?  
  
Нет. Но Вы можете поглядеть [здесь](http://www.minutemansoftware.com/tutorial/tutorial_manual.htm) — внизу страницы есть некоторое количество примеров. Ну и полистайте мануал по GPSS по STORAGE, ENTER, LEAVE, TRANSFER и GATE (особенно GATE). Ну а в целом Ваша задача решается примерно так (за правильность и оптимальность не отвечаю, но примерно в этом духе).

trucks STORAGE 3

dispatcher STORAGE 7

GENERATE 5, 3

TRANSFER BOTH,wait,refuse

wait ENTER dispatcher

GATE SNF gettruck

gettruck ENTER trucks

LEAVE dispatcher

TRANSFER BOTH,truck1,try2

try2 TRANSFER BOTH,truck2,try3

try3 TRANSFER ,truck3

truck1 SEIZE T1

ADVANCE 12,5

RELEASE T1

TRANSFER ,endtruck

truck2 SEIZE T2

ADVANCE 25,8

RELEASE T2

TRANSFER ,endtruck

truck3 SEIZE T3

ADVANCE 25,8

RELEASE T1

endtruck LEAVE TRUCKS

refuse TERMINATE 1

1. Укажите последовательность блоков GPSS для настройки фрагмента, структурно реализующего выбор очереди и устройства из нескольких по некоторому критерию.

select  
queue  
seize  
advance  
release  
depart

Norm SELECT MIN Minque,First,Last,,Q ;Find minimum queue

QUEUE P$Minque ;Join the min queue

SEIZE P$Minque ;Get the checkout

DEPART P$Minque ;Depart the queue

ADVANCE V$Checkout ;Checkout time

RELEASE P$Minque ;Free the checkout

LEAVE P$Carrier ;Leave the cart

Fin TABULATE Transit ;Record transit time

TABULATE Items ;Record items bought

SAVEVALUE Customers-,1 ;One customer leaves

ADVANCE 600 ;Walk to the car

LEAVE Park ;Leave the car park

TERMINATE

Lost TERMINATE ;One customer lost

; GPSS World Sample File - SAMPQUE.GPS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Barber Shop Simulation \*

\* Time Is In Minutes \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GENERATE (Exponential(1,0,6.5)) ;Create next customer.

QUEUE Barber ;Begin queue time.

QUEUE Total\_time ;Total haircut time

TRANSFER Both,Barb1,Barb2 ;Choose not busy barber.

Barb1 SEIZE Barber1 ;Get service by Barber1.

DEPART Barber ;End queue time.

ADVANCE 10,2.5 ;Haircut takes a few minutes.

DEPART Total\_time ;Leave total time queue

RELEASE Barber1 ;Give up the barber.

TRANSFER ,Next ;Used Barber1-Done with cut.

Barb2 SEIZE Barber2 ;Get service by Barber2

DEPART Barber ;End queue time.

ADVANCE 13,4 ;Haircut takes a few minutes.

DEPART Total\_time ;Leave total time queue

RELEASE Barber2 ;Give up the barber.

Next SAVEVALUE Ave\_Queue,QT$Barber ;Save resulting average.

TERMINATE 1 ;Customer leaves.

1. Укажите последовательность блоков GPSS для настройки фрагмента, структурно реализующего трек вида "агрегат" (например, светофор).

GENERATE 20,10 ;Create next automobile.  
QUEUE Eastwest  
TEST E x$svetoff,1,gosvet;проверяем, работает ли светофор  
TEST E q$Northsouth,0  
TRANSFER ,goint  
  
gosvet TEST E X$EWlight,F$Intersection ;Block until green, and  
\* the intersection is free  
goint SEIZE Intersection  
DEPART Eastwest ;End queue time.  
ADVANCE 10 ;Cross the intersection.  
RELEASE Intersection  
TERMINATE ;Auto leaves intersection.  
\*  
GENERATE 30,10 ;Create next automobile.  
QUEUE Northsouth  
TEST NE x$svetoff,1,gogo  
TEST E X$NSlight,F$Intersection ;Block until green and  
\* the intersection is free  
gogo SEIZE Intersection  
DEPART Northsouth ;End queue time.  
ADVANCE 10 ;Cross the intersection.  
RELEASE Intersection  
TERMINATE ;Auto leaves intersection.  
  
GENERATE (2#60#60),,,1  
SAVEVALUE svetoff,1  
ADVANCE (2#60#60)  
SAVEVALUE svetoff,0  
TERMINATE  
  
GENERATE (24#60#60)  
TERMINATE 1  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* \*  
\* Traffic Light Simulation \*  
\* \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
GENERATE ,,,1  
Begin1 SAVEVALUE NSlight,Red ;North-South light turns red  
SAVEVALUE EWlight,Green ;East-West light turns green  
ADVANCE Greentime ;Light is green  
SAVEVALUE NSlight,Green ;North-South light turns green  
SAVEVALUE EWlight,Red ;East-West light turns red  
ADVANCE Redtime ;Light is red  
TRANSFER ,Begin1  
Greentime EQU 400  
\* When the light is Green (value 0) and the intersection is not busy  
\* (the State Variable(SNA) F$Intersection evaluates as 0), a car may  
\* pass into the intersection. These conditions are tested at the TEST  
\* block. When the light is red (value 100) or the intersection is busy  
\* (SNA F$Intersection returns the value 1), the condition at the TEST  
\* block will not be met and the car will not proceed.  
Green EQU 0  
Red EQU 100  
Redtime EQU 300  
\*  
\* Do START 4000 EW Congestion builds. Try Greentime EQU 1000. Fine  
\* but NS congestion builds. Greentime EQU 400 works for both.

1. Укажите последовательность блоков GPSS для настройки фрагмента, моделирующего обработку на одноканальном обработчике с повторными циклами.

; GPSS World Sample File - BOOKMRK.GPS

100 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

110 \* \*

120 \* \*

130 \* Telephone System Model \*

140 \* by Gerard F. Cummings \*

150 \* \*

160 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

170 \* Simple Telephone Simulation \*

180 \* Time Unit is one second \*

200 Sets STORAGE 2

210 Transit TABLE M1,100,100,20 ;Transit times

220 GENERATE 100,60 ;Calls arrive

230 Again GATE SNF Sets,Occupied ;Try for a line

240 ENTER Sets ;Connect call

250 ADVANCE 180,60 ;Speak for 3+/-1 min

260 LEAVE Sets ;Free a line

270 TABULATE Transit ;Tabulate transit time

280 TERMINATE 1 ;Remove a transaction

290 Occupied ADVANCE 300,60 ;Wait 5 minutes

300 TRANSFER ,Again ;Try again

310 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. В данной модели укажите номер блока, в котором происходит управление очередью на обслуживание в объекте Server…

<https://all4study.ru/gpss/ocheredi-v-gpss.html>

Транзакты в процессе перемещения по модели могут задерживаться перед некоторыми блоками, если условия входа не выполняются. Примерами таких блоков из рассмотренных выше являются блоки:  
• **SEIZE** (если ранее вошедший в этот блок транзакт не вошел в блок RELEAZE),  
• **ENTER** (если требуемая текущим транзактом емкость памяти больше емкости свободного участка данной памяти),  
• **GATE** и **TEST** (если в этих блоках не указан альтернативный выход и проверяемое условие не выполняется).

При поступление транзактов на вход задерживающих блоков образуются очереди. Для сбора статистики об очереди в местах задержки транзактов ставят блок входа в очередь QUEUE A, B, где А — имя или номер очереди, В — число элементов, добавляемых в очередь транзактом, по умолчанию В=1. Этот блок сам по себе не создает очереди, а лишь являются средством ее регистрации. При входе в блок QUEUE текущая длина очереди получает приращение на величину, задаваемую полем В. Блок выхода из очереди имеет вид DEPART A, B, где А — имя или номер очереди, B — число элементов, удаляемых из очереди транзактом.

Таким образом, работа очереди в GPSS-модели отображается двумя блоками:

QUEUE A, B  
DEPART A, B

Очереди имеют следующие СЧА:  
Q$j – текущая длина очереди j;  
QM$j – максимальная длина очереди j;  
QA$j – средняя длина очереди j;  
QC$j – число входов в очередь j;  
QZ$j – число входов в очередь с нулевым временем пребывания (транзакт прошел через блок QUEUE, не задерживаясь в очереди);  
QT$j – среднее время пребывания в очереди j, включая нулевые входы;  
QXSj – среднее время пребывания в очереди j без нулевых входов.

1. Для данной модели укажите атрибут для коэффициента использования объекта Server…

Найти **Util**

1. Для данной модели укажите атрибут, означающий вероятность обработки транзакта в устройстве Server

В примере вероятность обслуживания 25%

TRANSFER .25,,Try2 ;Want to shop aisle 1?

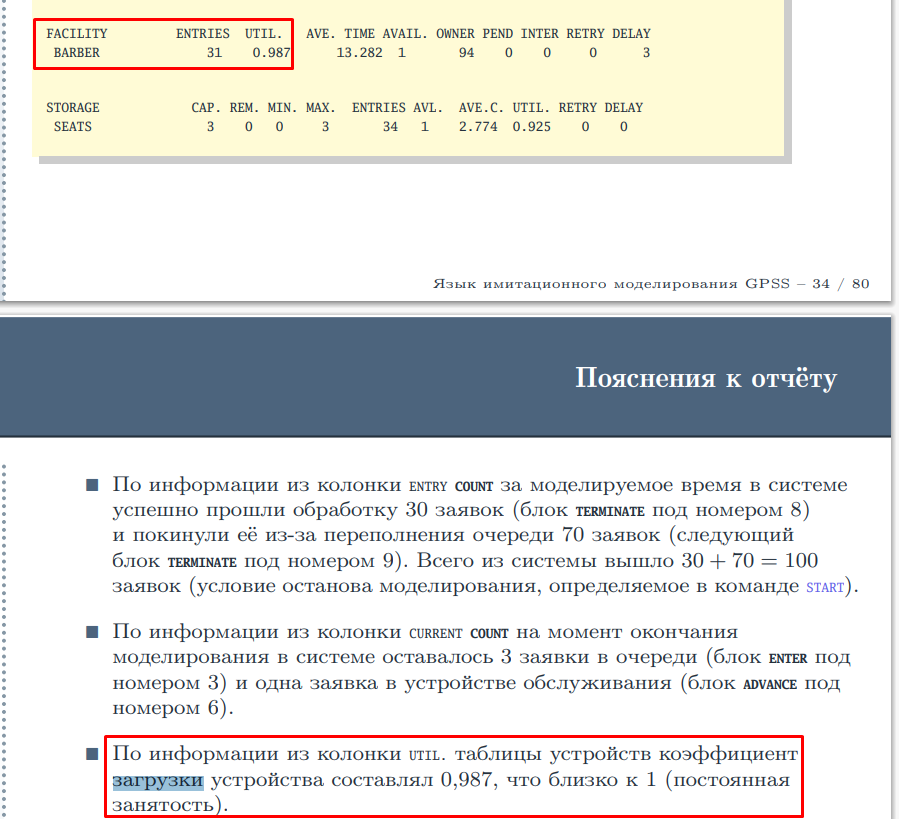
ADVANCE 120,60 ;Time in aisle 1

1. Для данной модели GPSS оцените количество прошедших через устройство Server транзактов …

Найти **Entries**

1. Для данной модели GPSS оцените коэффициент загрузки устройства Server

<http://staff.mmcs.sfedu.ru/~dubrov/files/tut_ipm_01_gpss.pdf>



1. … Для данной модели GPSS оцените время завершения моделирования …

Найти **END TIME**

1. Для данной модели GPSS оцените количество транзактов, прошедших через очередь serv …

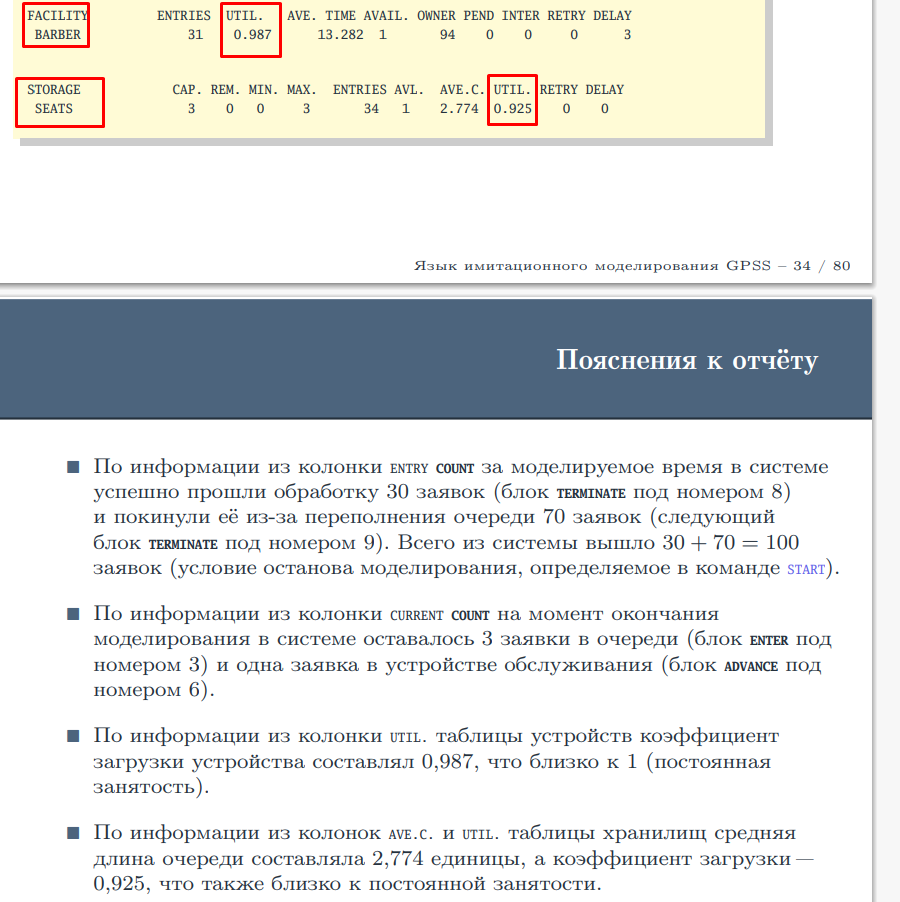
Вычесть из **ENTRY\_COUNT CURRENT\_COUT** и полученную разность вписать в ответ

1. Для данной модели GPSS оцените количество созданных транзактов в модели …

Найти **Start**

1. Для данной модели GPSS оцените значение коэффициента загрузки устройства Uzel…(смотри **61 вопрос**)
2. Для данной модели GPSS оцените коэффициент загрузки наиболее загруженного из устройств Ustr1,Ustr2,Ustr3 …

Смотреть **Util** и сравнивать результаты

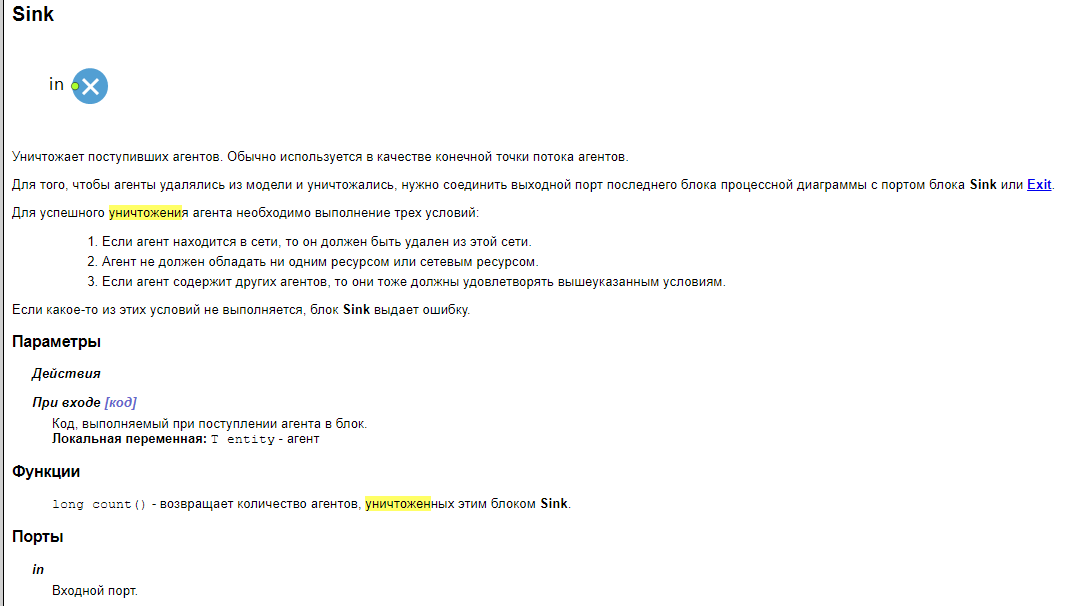


1. Для данной модели GPSS укажите порядок устройств Ustr1,Ustr2,Ustr3 по убыванию величины коэффициента загрузки…(см. вопрос 66)
2. Какой вариант гистограммы на графиках отображает состояние таблицы Distrib в конце моделирования…
3. Какие модельные блоки в среде AnyLogic используются для моделирования устройств-обработчиков?

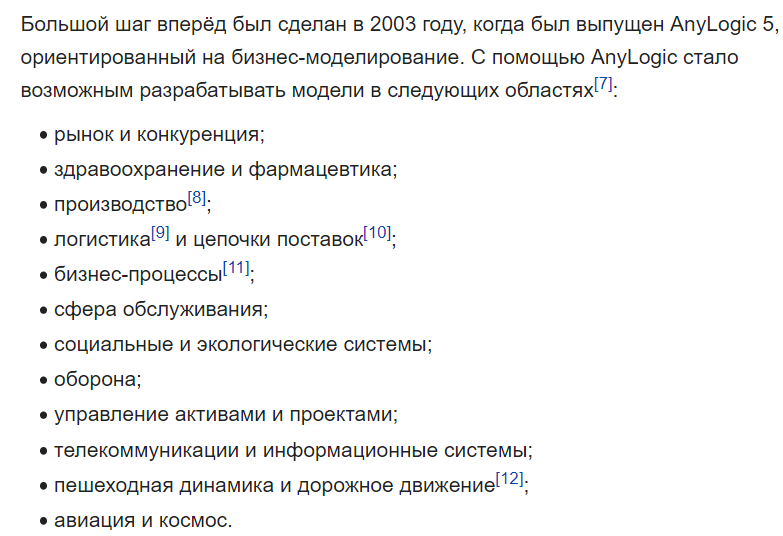
Service

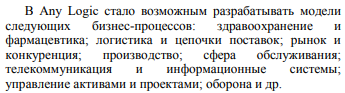
Блок **PedService** моделирует то, как пешеходы движутся к сервисам, заданным графически элементом разметки и проходят через сервис. Перетащите **блок PedService** из палитры **Пешеходной библиотеки** в графическую диаграмму, поместите сразу за блоком создания пешеходов. Размещая блок посередине соединителя, мы соединяем порты автоматически (но помните, что нужно присоединить правый порт блока *pedService*, а не нижний).

1. Какие модельные блоки используются в среде AnyLogic для уничтожения агентов?



1. Какие направления имитационного моделирования поддерживает среда AnyLogic?





1. Укажите в списке элементы моделей, используемые в среде AnyLogic –

Source, Sink, Go TO, Service

1. Какие модельные блоки в среде AnyLogic используются для создания агентов?

Source (например, Ped Source)

Объект[**PedSource**](http://127.0.0.1:64702/help/topic/com.anylogic.help/html/pedestrian/reference/pedsource.html)создает пешеходов. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток пешеходов. В нашем примере он моделирует приход пассажиров в павильон.

1. Какие модельные блоки в среде AnyLogic используются для маршрутизации движения агентов?

Go To

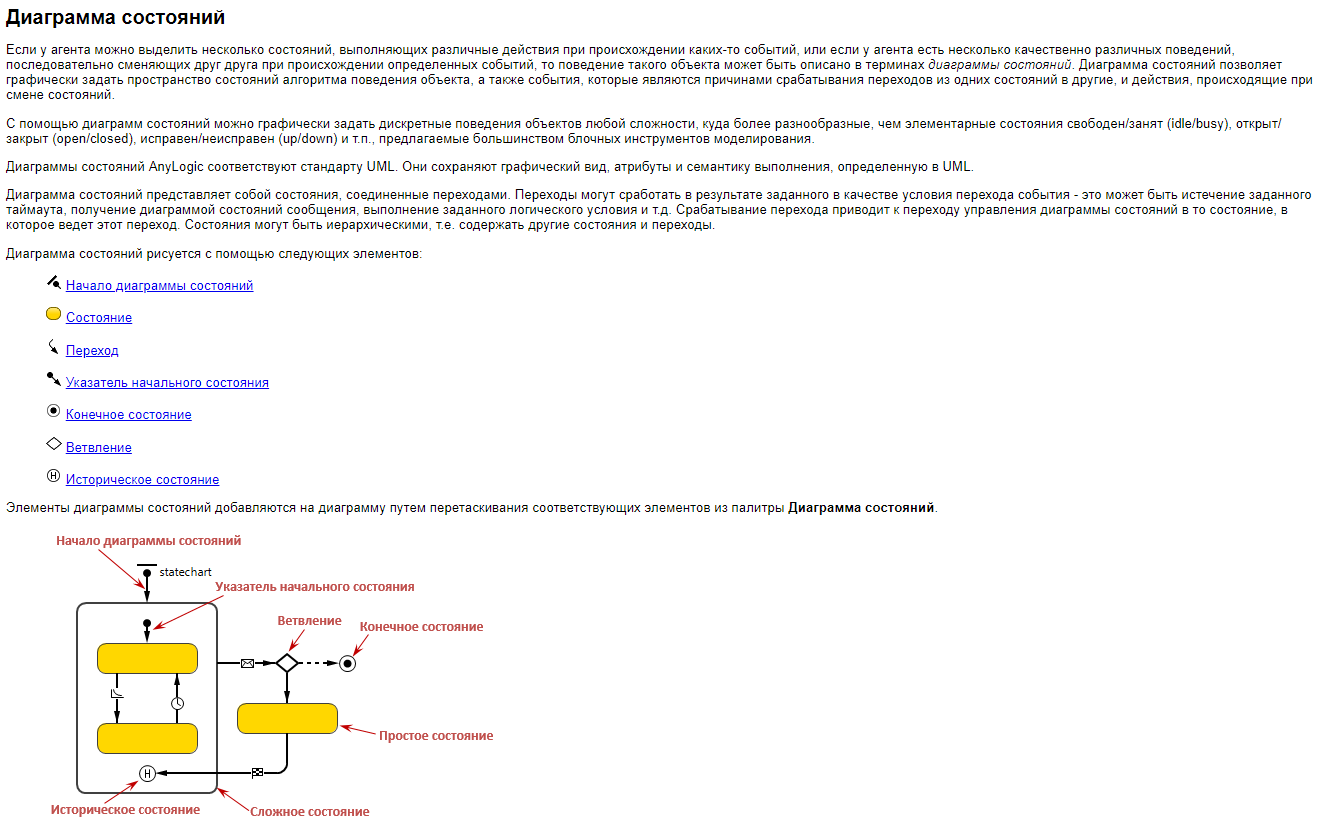
Объект [**PedGoTo**](http://127.0.0.1:64702/help/topic/com.anylogic.help/html/pedestrian/reference/pedgoto.html)моделирует перемещение пешеходов из текущего местоположения в другое (заданное параметром этого объекта). С помощью этого объекта мы будем моделировать то, как пассажиры перемещаются от входа в павильон к поездам метро.

1. Какие объекты в среде AnyLogic используются для моделирования разделяемых ресурсов системы?

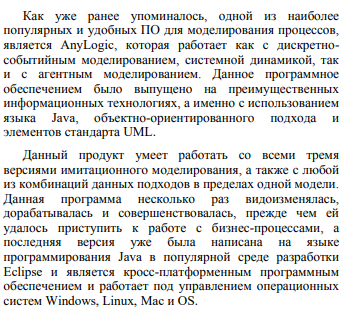
Seize, Resource Pool, Release

1. Что понимается под диаграммой состояний в среде AnyLogic?

Диаграмма состояний представляет собой состояния, соединенные переходами. Переходы могут сработать в результате заданного в качестве условия перехода события - это может быть истечение заданного таймаута, получение диаграммой состояний сообщения, выполнение заданного логического условия и т.д. Срабатывание перехода приводит к переходу управления диаграммы состояний в то состояние, в которое ведет этот переход. Состояния могут быть иерархическими, т.e. содержать другие состояния и переходы.



1. На каком языке программирования формируется программа модели в среде AnyLogic?



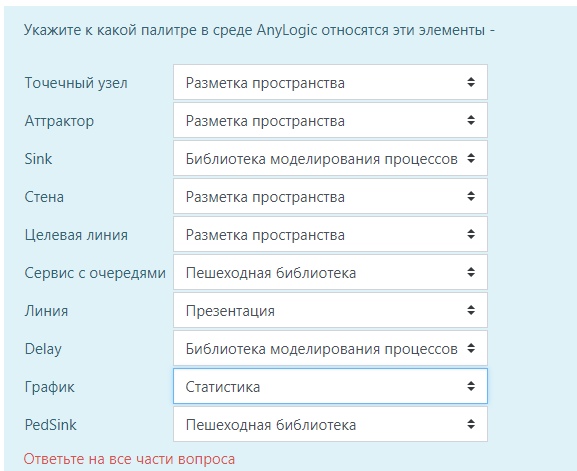
1. В каких случаях применяется агентное моделирование, а в каких дискретно-событийное?

Мы увидели, что во многих случаях АМ позволяет легче отобразить в модели многие явления реального мира, чем СД или ДС-моделирование. Это, однако, не означает, что АМ – абсолютная замена традиционных подходов. Для большого числа приложений СД и ДС позволяют эффективно строить адекватные модели и получать достоверные результаты. Более того, в таких случаях попытки применить агентное моделирование могут быть менее продуктивными: агентные модели труднее строить. Таким образом, если есть ощущение, что задача хорошо подпадает под один из традиционных подходов, его нужно использовать без колебаний. В этом под рукой большое количество коммерческих инструментов, AnyLogic – один из них. Агентное моделирование для тех, кто хочет выйти за рамки ограничений, присущих системной динамике и дискретно-событийному моделированию [см. также 14]. Оно особенно эффективно, как мы видели, при моделировании систем, содержащих большие количества активных объектов. Для таких систем AnyLogic поможет вам разработать агентную модель с минимальными усилиями, а также полностью или частично перейти от существующей СД или ДС модели к агентам.

1. Для данной модели укажите атрибут для коэффициента использования объекта Server

FR$Server

1. Укажите к какой палитре в среде AnyLogic относятся эти элементы



1. Какие модельные блоки в среде AnyLogic используются для маршрутизации движения агентов?

PedSelectOutput

SelectOutput

1. Что понимается под диаграммой состояний в среде AnyLogic

Алгоритм поведения агента

Mnbvcxz