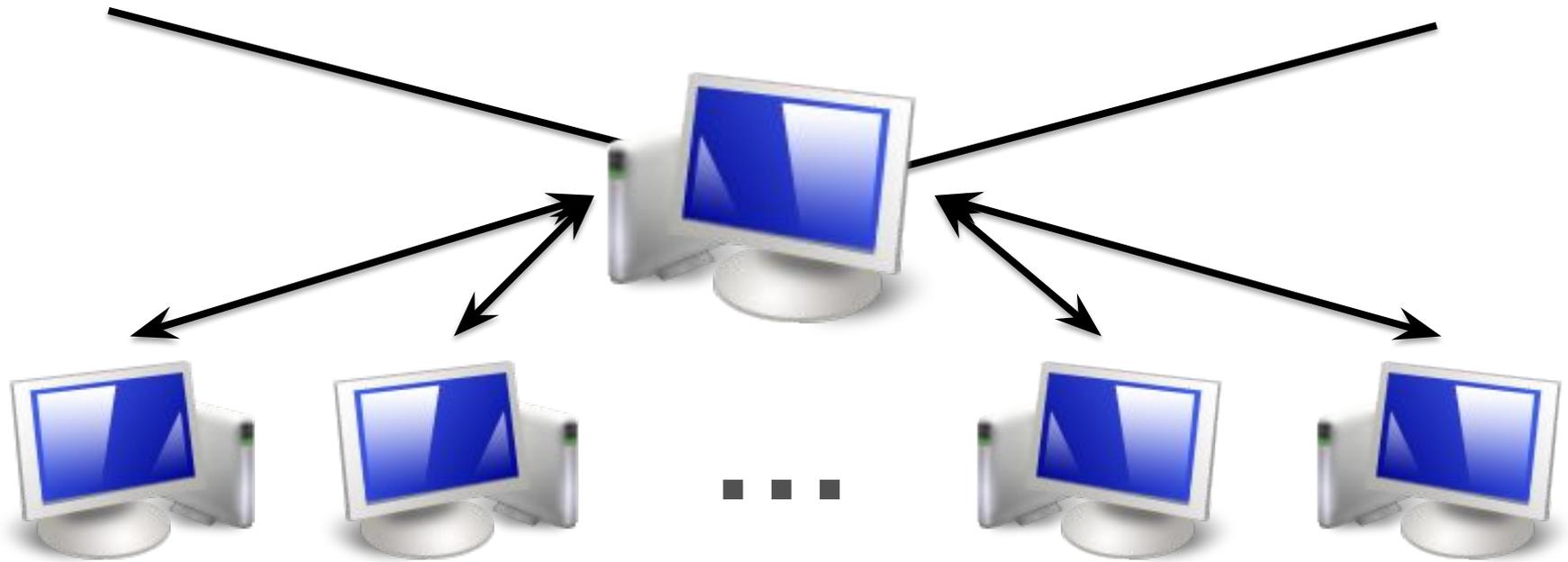


Кафедра вычислительной техники

МОДЕЛИРОВАНИЕ



Смоленск-2016

Тема 2

Разработка моделей, инструментальные средства моделирования систем

Лекция

Язык моделирования GPSS

1

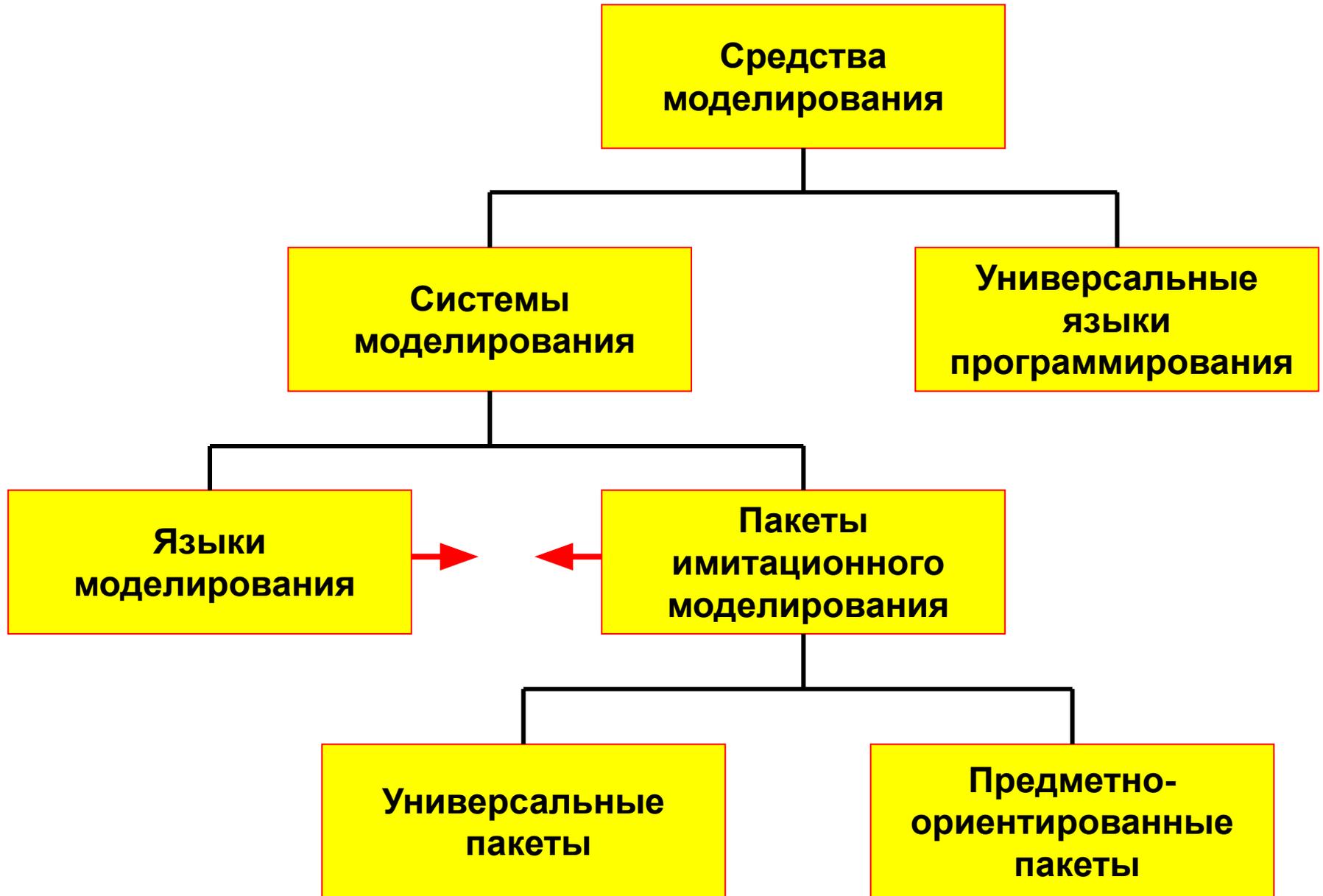
Классификация инструментальных средств моделирования

**МОДЕЛИРОВАНИЕМ
НАЗЫВАЕТСЯ ЗАМЕЩЕНИЕ
ОДНОГО ОБЪЕКТА ДРУГИМ С
ЦЕЛЮ ПОЛУЧЕНИЯ
ИНФОРМАЦИИ О ВАЖНЕЙШИХ
СВОЙСТВАХ ОБЪЕКТА-
ОРИГИНАЛА С ПОМОЩЬЮ
ОБЪЕКТА-МОДЕЛИ**

Имитационное моделирование — это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, с которой проводятся эксперименты с целью получения информации об этой системе.

Экспериментирование с моделью называют *имитацией* (имитация — это постижение сути явления, не прибегая к экспериментам на реальном объекте).

Классификация средств моделирования



Сравнение универсальных языков программирования и систем моделирования

Системы моделирования

Системы моделирования обеспечивают естественную среду

для разработки моделей;

Модели, созданные с помощью систем моделирования проще модифицировать и использовать;

Системы моделирования обеспечивают более совершенные механизмы отладки моделей

Системы моделирования обеспечивают сокращение времени и стоимости разработки модели;

Универсальные языки программирования

Скорость выполнения прогонов моделей, написанных на языках программирования выше, чем моделей, написанных на языках моделирования;

Языки программирования обеспечивают большую гибкость моделей, чем системы моделирования;

Языки программирования знает большинство разработчиков;

Сравнение языков и пакетов имитационного моделирования

Языки моделирования

Языки моделирования по своей природе универсальны;

Разработка модели на языке моделирования подразумевает написание кода;

Языки моделирования предоставляют достаточно широкие возможности моделирования, но часто трудны в использовании

Пакеты имитационного моделирования

Пакеты моделирования, как правило, ориентированы на решение определенных задач

Разработка модели в пакете моделирования производится с использованием графического интерфейса;

Пакеты моделирования проще изучать и использовать, но при решении некоторых задач, они не могут обеспечить достаточную гибкость моделирования

Математические пакеты и моделирование

Математические пакеты *Mathematica*, *Maple*, *MathLab*, очень эффективно решают задачи **аналитического** моделирования.

Для задач **ИМИТАЦИОННОГО** моделирования пакеты имеют:

генераторы случайных чисел

генераторы случайных величин с различными законами распределения;

средства программирования, необходимые для описания логики работы сложных систем;

средства статистической обработки результатов моделирования.

Некоторые пакеты имеют специальные средства для построения моделей (*MathLab* имеет пакет расширения *Simulink*)

Однако, математические пакеты весьма неудобны для имитационного моделирования.

2

Общая характеристика GPSS World

GPSS – General Purpose Simulation System

(общецелевая система моделирования)

**Разработана Дж.Гордоном (США) в
фирме IBM в начале 60-х годов**

**Самый популярный в мире язык
моделирования**

Литература по GPSS:

- 1. Шрайбер Т.Д. Моделирование на GPSS. М., Машиностроение, 1980.- 592 с.**
- 2. Руководство пользователя по GPSS World. /Перевод с английского/. Казань, изд-во «Мастер Лайн», 2002.- 384 с.**
- 3. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М., ДМК Пресс, 2004.- 320 с.**

Язык GPSS – это язык декларативного типа, построенный по принципу объектно-ориентированного языка.

Как и в каждом объектно-ориентированном языке в GPSS каждый объект имеет свойства и методы, которые изменяют эти свойства.

В GPSS свойства объектов называют **стандартными числовыми атрибутами (СЧА).**

Основными элементами этого языка являются **транзакты и **блоки**, которые отображают соответственно динамические и статические объекты моделируемой системы.**

Достоинства языка GPSS:

1. Модели на GPSS более компактны, т.к. состоят из меньшего числа операторов, чем такие же модели, написанные на процедурных языках

(- в **GPSS** встроено максимально возможное число логических программ, необходимых для моделирования систем;

- в **GPSS** входят специальные средства для описания динамического поведения систем, меняющихся во времени, причем изменение состояний происходит в дискретные моменты времени).

2. GPSS удобен при программировании, поскольку интерпретатор GPSS многие функции выполняет автоматически

(- **GPSS** без специального на то указания пользователя собирает статистические данные, описывающие поведение модели, автоматически печатает итоговую статистику по завершении моделирования;

- **GPSS** обслуживает таймер модельного времени, планирует события, которые должны произойти позднее в течение времени моделирования, вызывает их своевременное появление и управляет очередностью поступления).

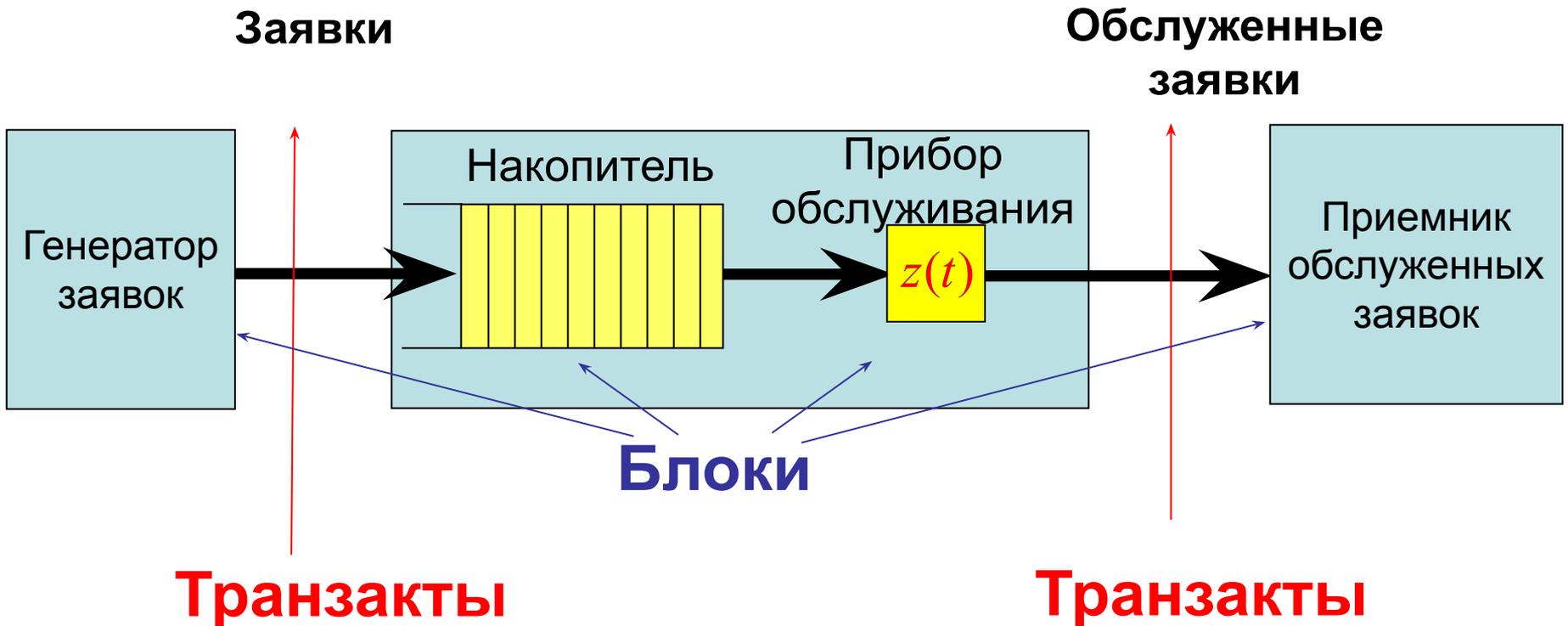
Система GPSS ориентирована на класс объектов, процесс функционирования которых можно представить в виде множества состояний и правил перехода из одного состояния в другое, определяемых в дискретной пространственно-временной области.

Примерами таких объектов являются:
вычислительные системы, сети ЭВМ,
системы передачи сообщений,
транспортные объекты,
склады, магазины, предприятия и т.п.

В качестве формальных моделей таких объектов используют:
системы массового обслуживания,
автоматы,
стохастические сети и т.п.

Простейшая система массового обслуживания

$$z(t) = \begin{cases} \text{канал свободен} & ; \\ \text{канал занят} & . \end{cases}$$



Транзакты

Содержательное значение транзактов определяет разработчик модели

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

С точки зрения программы, транзакт - это структура данных, которая содержит поля:

имя или номер транзакта;

время появления транзакта;

текущее модельное время;

номер блока, в котором находится транзакт;

номер блока, куда он продвигается;

момент времени начала продвижения;

приоритет транзакта;

параметры транзакта: P1, P2, ...

Транзакты

Транзакты нумеруются по мере их появления в модели.

Каждый транзакт занимает некоторый объем памяти ЭВМ.

Память под транзакты выделяется динамически. После того, как транзакт закончит свое движение по блокам модели, его необходимо уничтожить для освобождения памяти.

Блоки

<i>Система</i>	<i>Элементы систем, которые моделируются блоками</i>
Магазин	Кассир (продавец), очередь в кассу
Автомобильное шоссе	Перекресток, очередь перед светофором
Склад	Средства доставки грузов, список заказов

С точки зрения программы, блок - это подпрограмма, написанная на макроассемблере или на языке Си, и содержащая набор параметров (операндов) для обращения к ним.

Взаимодействие блоков и транзактов

Передача управления от блока к блоку в GPSS-программах реализуется с помощью движения транзактов в модельном времени.

В начале моделирования в GPSS-модели не существует ни одного транзакта.

В процессе моделирования транзакты входят в модель в определенные моменты времени, соответствующие логике функционирования моделируемой системы.

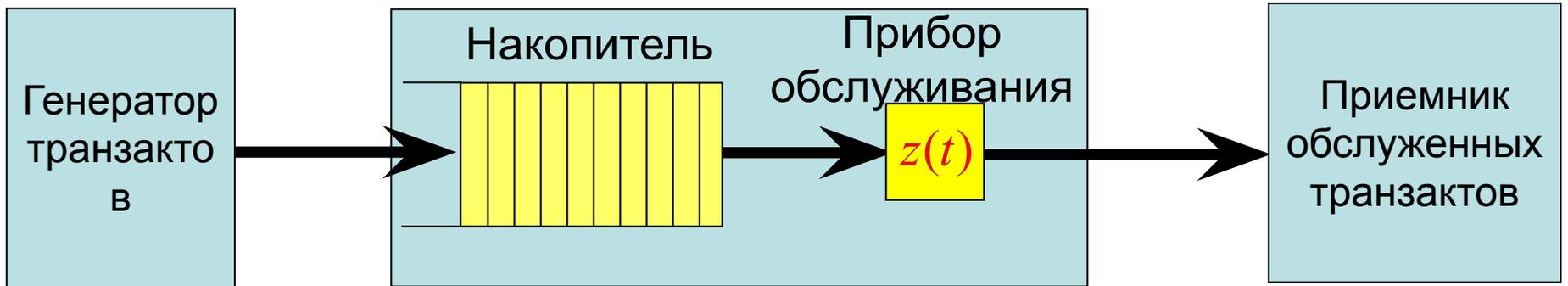
Таким же образом транзакты покидают модель в зависимости от специфики моделирования.

В общем случае в модели существует несколько транзактов, но в каждый момент времени движется только один из них.

Если транзакт начал свое движение, он передвигается от блока к блоку по пути, указанному блок-схемой (логикой работы модели).

В тот момент, когда транзакт входит в блок, вызывается соответствующая этому блоку подпрограмма.

Порядок перемещения транзактов



Транзакт перемещается в модели до тех пор, пока не выполнится одно из трех условий:

1. Транзакт входит в блок, функцией которого является задержка транзакта на определенное время

2. Транзакт входит в блок, функцией которого является удаление транзакта из модели

3. Транзакт пытается войти в следующий блок, но блок не принимает этот транзакт.

В этом случае транзакт остается в том блоке, в котором в данное время находится, но позже будет повторять попытки войти в следующий блок.

Когда условия в модели изменятся,

одна из таких попыток может быть успешной.

После этого транзакт продолжит свое перемещение по модели.

**Объекты типа «ресурсы»
(обслуживающие устройства)**

```
graph TD; A[Объекты типа «ресурсы»  
(обслуживающие устройства)] --> B[Устройства]; A --> C[Многоканальные устройства]; A --> D[Логические ключи];
```

Устройства

ресурс, который
в любой момент
времени
может быть
занят только
одним транзактом

**Многоканальные
устройства**

ресурсы, для
параллельной
обработки.
Могут быть
использованы
несколькими
транзактами.

**Логические
ключи**

Управляют
движением
транзактов,
разрешая или
запрещая проход



Часы модельного времени

Время измеряется в условных единицах.

Единица модельного времени определяется разработчиком таким образом, чтобы правильно отобразить события реальной системы в модели.

Часы модельного времени продвигаются от одного события к другому.

3

Построение программ в GPSS World

Типы операторов GPSS

```
graph TD; A[Типы операторов GPSS] --> B[Блоки]; A --> C[Операторы описания данных]; A --> D[Команды GPSS];
```

Блоки

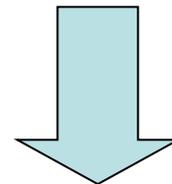
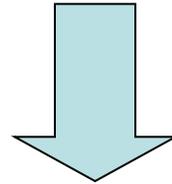
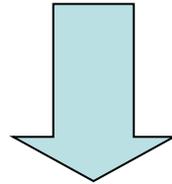
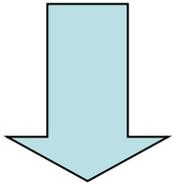
Операторы
описания
данных

Команды
GPSS

В GPSS для ссылки на блоки и объекты используются *имена* (идентификаторы).

Формат GPSS-блоков :

[< *Метка* >]< *Операция* >< *Операнды* > [<; *Комментарии* >]



Имя блока - символы, числа.
Параметр обязателен только для некоторых блоков

Глагол, который описывает основное функциональное назначение блоков.

Задают характеристики блока. Число операндов зависит от типа блока. Всего не больше семи операндов. Обозначаются буквами **A, B, C, D, E, F, G**. Могут задаваться по умолчанию. Если операнд опущен, то вместо него ставится запятая.

Отделяется от операндов символом ;

Ввод транзактов в модель, блок *GENERATE*

Блок *GENERATE* (ГЕНЕРИРОВАТЬ) – это блок, через который транзакты входят в модель. Не существует ограничений на количество разных блоков ***GENERATE*** в одной модели.

Формат блока:

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

Операнд	Значение	Значение по умолчанию
A	Средний интервал времени (число, СЧА) или вероятностное распределение интервала из встроенной библиотеки процедур (содержит >20 вероятностных распределений)	0
B	Половина поля допуска равномерно распределенного интервала (число, СЧА)	0
C	Смещение интервалов	Смещение отсутствует
D	Ограничитель транзактов	∞
E	Уровень приоритета транзакта. Возможные значения 0 – 127	0

Примеры распределений

- равномерное(Uniform);
- экспоненциальное(Exponential);
- Пуассона(Poisson);
- Гамма(Gamma);
- биномиальное(Binomial);
- треугольное(Triangular);
- нормальное(Normal);

Примеры:

Пример 1

Задание равномерного закона распределения:

GENERATE 6,4

Операнды: **A** = 6, **B** = 4. Интервал времени поступления является случайным числом со средним значением 6 и полем допуска 8, то есть он может приобретать только одно из девяти разных значений: 2,3,4,5,6,7,8,9,10.

Пример 2

Задание детерминированного значения интервалов поступления:

GENERATE 10

Операнды: **A** = 10, **B** = 0 (по умолчанию).

Транзакты входят в модель каждые 10 единиц модельного времени.

Пример 3

Генерирование одного транзакта.

GENERATE ,,1

Операнды: **A** = **B** = **C** = 0 (по умолчанию), **D** = 1.

В нулевой момент в модель входит один транзакт.

Блоки **GENERATE** являются основными средствами создания транзактов и ввода их в модель.

Примеры:

Пример 4

Для обращения к вероятностному распределению необходимо указать имя библиотечной процедуры и её параметры, заключённые в круглые скобки и отделённые друг от друга запятой:

<Имя процедуры>(G,A,B, ...)

Здесь **G** – номер генератора равномерно распределённых случайных чисел (от 1 до 999) – используется в качестве аргумента для формирования случайных величин с заданным законом распределения. Остальные параметры **A, B, ...**, количество которых для разных распределений составляет от 1 до 4, задают непосредственно параметры вероятностного распределения.

GENERATE (Exponential(1,0,50));

1 - № генератора равномерно распределённых случайных чисел;

50 – математическое ожидание случайной величины;

0 - смещение распределения относительно нуля (min. значение)

Удаление транзактов из модели, блок **TERMINATE**

Блок **TERMINATE (ЗАВЕРШИТЬ)** – это блок, через который транзакты выходят из модели. Не существует ограничений на количество разных блоков **TERMINATE** в одной модели.

Формат блока:

TERMINATE [A]

Операнд **A** является величиной уменьшения специального счетчика, который называется *счетчиком завершения*.

Этот операнд задает величину, которая вычитается из счетчика каждый раз, когда транзакт входит в блок **TERMINATE**. По умолчанию **A = 0**.

Вход транзакта в блок **TERMINATE** с нулевым значением операнда **A** не вызывает уменьшения счетчика завершения.

Счетчик завершения – это ячейка в памяти ЭВМ, которая хранит целое положительное число.

Начальное значение этого счетчика устанавливается в начале моделирования.

Оно равняется значению операнда **A** команды **START (НАЧАТЬ)**.

В процессе моделирования транзакты попадают в блок **TERMINATE** и, таким образом, уменьшают значение счетчика на величину операнда **A**.

Моделирование заканчивается, когда значение счетчика становится равным нулю или отрицательному числу.

Способы управления продолжительностью моделирования

Способ 1

1. В команде **START** операнду **A** присваивается значение заданного числа транзактов.
2. Во всех блоках **TERMINATE**, через которые транзакты заданного типа покидают модель, операнду **A** присваивается значение «1» или другое, отличное от нуля (соответственно содержательному значению транзактов).
3. Во все других блоках **TERMINATE** используется значение операнда **A** по умолчанию (**A** = 0). Значение счетчика завершения не будет зависеть от этих блоков.

Первый способ позволяет закончить моделирование, когда через модель пройдет заданное количество транзактов, например 1000:

```
GENERATE 40,5  
...  
TERMINATE 1  
START 1000
```

Способы управления продолжительностью моделирования

Способ 2

• Пусть разработчик выбрал за единицу модельного времени 1 мин и хочет смоделировать поведение системы на протяжении 8 часов.

1. Ввести в модель таймер-сегмент, состоящий из двух блоков:

```
GENERATE 600  
TERMINATE 1  
START1
```

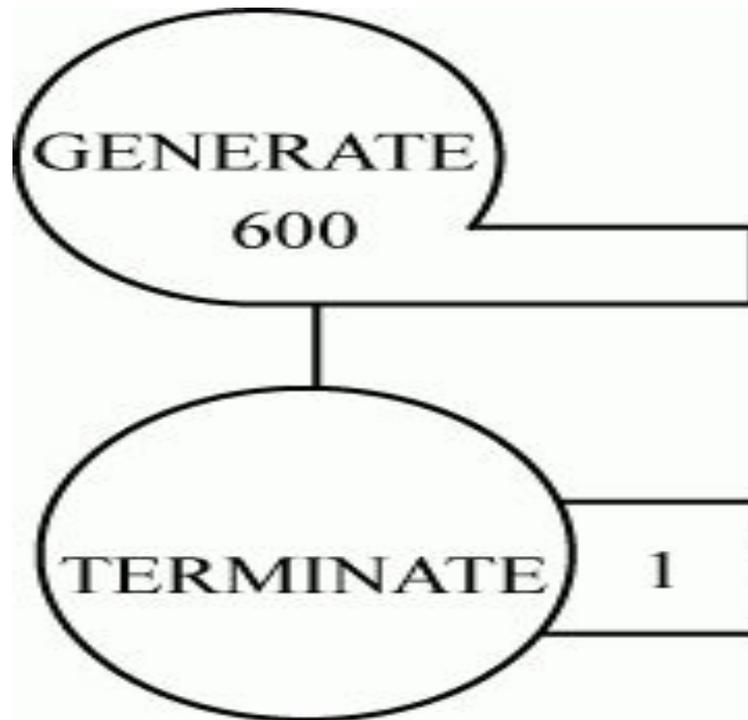
• 2. Во всех других блоках **TERMINATE** в модели использовать значение операнда **A** по умолчанию (**A** = 0). Это означает, что прекращение моделирования, определяемое счетчиком завершения, не будет зависеть от других блоков **TERMINATE**.

3. В команде **START** операнд **A** должен равняться единице.

Таким образом, в процессе моделирования завершение движения транзактов в других блоках **TERMINATE** не влияет на счетчик завершения.

В момент времени $T=480$ транзакт выйдет из блока **GENERATE** и сразу же перейдет в блок **TERMINATE**. Счетчик завершения уменьшится на единицу, и интерпретатор завершит моделирование.

Блок-диаграмма



Элементы, отображающие одноканальное устройство обслуживания

Основные свойства устройства:

1. Каждое устройство в любой момент времени может обслуживать только один транзакт

Если в процессе обслуживания появляется новый транзакт, то он должен:

- 1) либо подождать своей очереди,
- 2) либо направиться в другое место,
- 3) либо, если вновь пришедший транзакт имеет больший приоритет, устройство прерывает текущее обслуживание и начинает обслуживать новый транзакт.

2. Когда транзакт поступает в устройство, он должен пробыть там необходимое для обслуживания время

Действия транзакта для использования одноканального обслуживающего устройства:



Формат блока:

SEIZE A

Операнд	Значение	Результат по умолчанию
A	Имя (символьное или числовое) занимаемого устройства	Ошибка

Свойства блока *SEIZE* :

1. Если в текущий момент времени устройство используется, то транзакт не может войти в блок и должен ожидать своей очереди.
2. Если устройство свободно, транзакт может войти в блок.

Вход транзакта в блок вызывает выполнение подпрограммы обработки этого

блока. Состояние устройства изменяется со СВОБОДНОЕ на ЗАНЯТОЕ.

Формат блока:

RELEASE A

Операнд	Значение	Результат по умолчанию
A	Имя (символьное или числовое) освобождаемого устройства	Ошибка

Формат блока:
ADVANCE A[,B]

Операнд	Значение	Значение по умолчанию
A	Среднее время задержки на обслуживание (число, СЧА, вероятностное распределение)	0
B	Половина поля допуска равномерно распределенного времени задержки (число, СЧА)	0

Блок никогда не препятствует входу транзакта. Любое число транзактов может находиться в этом блоке одновременно. Когда транзакт попадает в такой блок, выполняется соответствующая подпрограмма и вычисляется время пребывания в нем транзакта. Вновь прибывший транзакт никак не влияет на уже находящийся в блоке транзакт.

Классический случай использования последовательности

SEIZE – ADVANCE – RELEASE:

Смоделировать работу одноканальной системы массового обслуживания на вход которой поступает однородный поток с интенсивностью $\lambda=16\pm 4$, время обслуживания транзактов в приборе имеет экспоненциальное распределение с $T_{cp}=8$. Заданное время моделирования 10 час.

GENERATE 16,4 ; Задание входного потока заявок

SEIZE PRIB ; заняли устройство с именем **PRIB**

ADVANCE (Exponential(1,0,8)); время обслуживания в устройстве

RELEASE PRIB ; освободили устройство с именем **PRIB**

TERMINATE 0 !

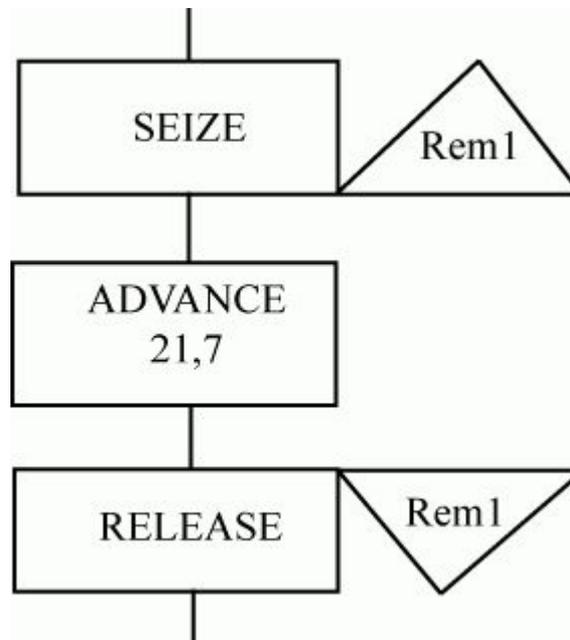
GENERATE 600

TERMINATE 1

START1

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

Блок-диаграмма



Пример: отчет по устройствам

FACILITY Номер или имя устройства	ENTRIES Количество входов	UTIL Коэффициент использования	AVE, TIME Среднее время пребывания транзакта в устройстве	AVAIL. Состояние отовности
PRIB	37	0.507	8.221	1

OWNER Номер последнего транзакта, занявшего устройство	PEND Количество прерванных в устройстве транзактов	INTER Количество прерывающих устройство транзактов	RETRY Количество транзактов, ожидающих специальных условий	DELAY Количество транзактов, ожидающих занятия устройства
38	0	0	0	0

Элементы, отображающие многоканальное устройство обслуживания

1. Определение емкости МКУ

Все используемые в модели МКУ должны быть заранее описаны, т.е. должно быть определено количество однотипных устройств, входящих в МКУ.

Для этого используется оператор *описания* **STORAGE** (память, емкость хранилища (количество однотипных ячеек))

<Метка> STORAGE A

Формат оператора задания емкости МКУ

<i>Поле</i>	<i>Информация в поле</i>
Метка	Имя МКУ
Операция	STORAGE
Операнд A	Емкость МКУ

Элементы, отображающие многоканальное устройство обслуживания

2. Использование МКУ

Использование МКУ аналогично использованию одноканального устройства. Элементом, который занимает и использует МКУ, является транзакт.

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

ENTER A [,B]

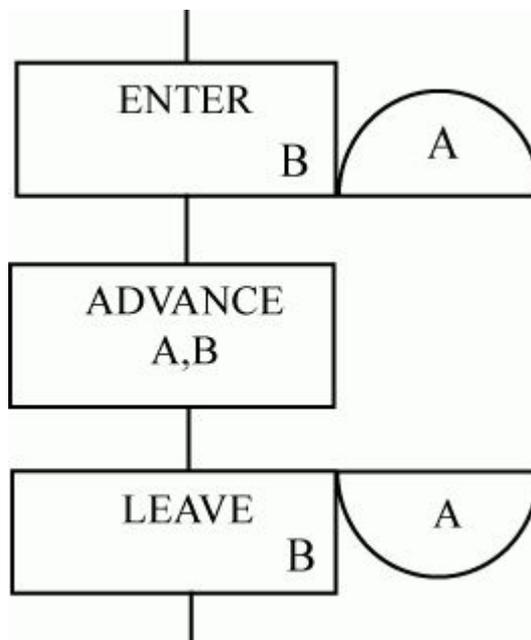
LEAVE A [,B]

<i>Операнд</i>	<i>Значение</i>	<i>Значение по умолчанию</i>
A	Имя МКУ	Ошибка
B	Количество одновременно занимаемых устройств	1

Пример:

RAM STORAGE 100 ; Объем памяти компьютера в 100 Кбайт
GENERATE 16,4 ; Задание входного потока заявок
ENTER RAM ;занять одну ячейку памяти
(ENTER RAM , P2) ; занять P2 ячеек памяти
ADVANCE 5 ; обслуживание в течении 5 ед. времени
LEAVE RAM ;освободить одну ячейку памяти
(LEAVE RAM,P2);освободить P2 ячеек памяти
.....

БЛОК-ДИАГРАММА



Пример: отчет по многоканальным устройствам

STORAGE	CAP.	REMAIN	MIN	MAX	ENTRIES
Имя устройства	Емкость МКУ	Количество единиц свободной емкости МКУ в конце периода моделирования	Минимальное количество используемой емкости	Максимальное количество используемой емкости	Количество входов в МКУ
RAM	100	100	0	62	50

AVL	AVE.C.	UTIL	RETRY	DELAY
Состояние готовности МКУ	Среднее значение занятой емкости МКУ	Коэффициент использования	Количество транзактов, ожидающих специальных условий	Количество транзактов, ожидающих занятия устройства
1	10,99	0,33	0	0

Сбор статистики об ожидании. Блоки **QUEUE, DEPART**

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

Являются *регистратором очереди* (не влияют на работу модели).

Используются только в паре.

Формат блоков:

QUEUE (стать в очередь) A [,B]

DEPART (покинуть очередь) A [,B]

Операнд	Значение	Результат по умолчанию
A	Имя очереди, в которую необходимо стать транзакту или которую надо покинуть (числовое или символическое имя, СЧА)	Ошибка
B	Число единиц, на которое увеличивается (уменьшается) длина очереди (число, СЧА)	1

ПРИМЕР:

Смоделировать работу одноканальной системы массового обслуживания на вход которой поступает однородный поток с интенсивностью $\lambda=16\pm 4$, время обслуживания транзактов в приборе имеет экспоненциальное распределение с $T_{ср}=8$. Собрать статистику о наличии очереди к прибору. Заданное время моделирования 10 час.

GENERATE 16,4 ; Задание входного потока заявок

QUEUE 1 (Q_PRIB)

SEIZE PRIB ; заняли устройство с именем **PRIB**

DEPART 1 (Q_PRIB)

ADVANCE (Exponential(1,0,8)); время обслуживания в устройстве

RELEASE PRIB ; освободили устройство с именем **PRIB**

TERMINATE 0

GENERATE 600

TERMINATE 1

START1

Порядок работы регистратора очереди

При входе транзакта в блок QUEUE (СТАТЬ В ОЧЕРЕДЬ) выполняются четыре действия:

- 1) **счетчик входов** для данной очереди увеличивается на B ;
- 2) **длина очереди (счетчик текущего содержимого)** для данной очереди увеличивается на B ;
- 3) значение текущей **длины очереди** хранится в стандартном числовом атрибуте <имя очереди>;
- 4) транзакт присоединяется к очереди с запоминая ее имени и значения текущего модельного времени.

Транзакт перестает быть элементом очереди только после того, как он переходит в блок DEPART (ПОКИНУТЬ ОЧЕРЕДЬ) соответствующей очереди.

Когда это происходит, интерпретатор выполняет следующие операции:

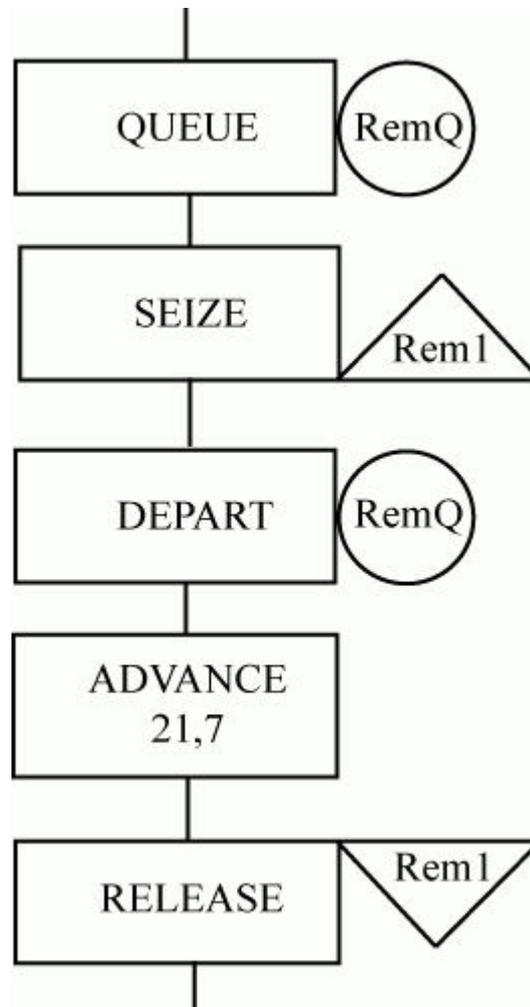
- 1) **длина очереди** соответствующей очереди уменьшается на B ;
- 2) используя привязку к значению времени, определяет: является ли время, проведенное транзактом в очереди, нулевым; если да, то такой транзакт по определению является транзактом с **нулевым пребыванием** в очереди и одновременно изменяется **счетчик нулевых вхождений**,
- 3) ликвидируется «привязка» транзакта к очереди.

Пример: Отчет по очередям

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE.TIME	AVE.(-0)	RETRY
Номер или имя очереди	Максимальная длина очереди	Текущая длина очереди	Общее кол-во входов	Количество «нулевых» входов	Средняя длина очереди	Среднее время пребывания транзактов в очереди	Среднее время пребывания в очереди без учета «нулевых» входов	Количество транзактов, ожидающих специальных условий

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
1 (QPRIB)	1	0	37	30	0.08	1.293	6.837	0

Блок-диаграмма



УСЛОВНЫЕ ОПЕРАТОРЫ

ОПЕРАТОР TEST

Сравнение параметров может быть выполнено с помощью блока **TEST (ПРОВЕРИТЬ)**

Формат блока : **TEST X A,B[,C]**

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
A	Проверяемое значение (СЧА)	Ошибка
B	Контрольное значение (число или СЧА)	Ошибка
C	Имя блока, в который переходит транзакт при условии, что заданное оператором отношения X условие, не выполняется	При отсутствии операнда C проверку выполняют в режиме отказа

Проверка числовых выражений. Блок TEST

TEST X A,B|,C|

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умолчанию</i>
X	Оператор отношения, использующийся при проверке	
	<i>Значение оператора отношений:</i>	<i>Вопрос оператора отношения:</i>
	G	A больше B?
	GE	A больше или равно B?
	E	A равно B?
	NE	A не равно B?
	LE	A меньше или равно B?
	L	A меньше B?

Блок **TEST** :может работать в двух режимах:

1. в режиме альтернативного выхода (если задан операнд **C**)
2. в режиме отказа (если операнд **C** не задан)

Пример:

Режим отказа

TEST LE Q1,Q2

Проверяющий транзакт будет задержан в предыдущем блоке до тех пор, пока длина первой очереди не станет меньше или равна длине второй очереди.

Режим условного перехода

TEST LE Q1,Q2,ZN

Проверяющий транзакт перейдет в следующий по порядку блок, если длина первой очереди меньше или равна длине второй очереди. Если это условие не выполняется, транзакт перейдет в блок с меткой **ZN**

TEST LE Q1, 5, ZN

Транзакт перейдет к следующему блоку если на данный момент времени длина очереди **Q1**, будет меньше или равна 5, в противном случае переходим к блоку с меткой **ZN**.

Блок-диаграмма

ОПЕРАТОР **TRANSFER** (ПЕРЕДАТЬ)

Он может быть использован в девяти разных *режимах*. Рассмотрим *три основных*.

Блок TRANSFER в режиме безусловной передачи.

Его формат : **TRANSFER ,B**

Позиция блока – это номер или метка блока. Так как операнд **A** не используется, то перед операндом **B** должна стоять запятая. В режиме безусловной передачи блок **TRANSFER** не может отказывать транзакту во входе. Если транзакт входит в блок, то он сразу же пытается войти в блок **B**.

Операнд	Значение	Результат по умолчанию
A	Не используется	—
B	позиция блока, в которую должен перейти транзакт	Ошибка

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

Формат блока: **TRANSFER A,[B],C**

При задании вероятности (операнд A) используется не более трех цифр, первый символ записи частоты «.» (десятичная точка), если используется действительное число, которое должно быть в пределах от 0 до 1,0 (например, 0,235). Если операнд – положительное целое число, то вероятность интерпретируется в долях тысячи.

<i>Операнд</i>	<i>Значение</i>	<i>Результат по умол чанию</i>
A	Вероятность передачи транзакта в блок C, задаваемая в долях тысячи	Ошибка
B	Позиция блока, в которую должен перейти транзакт (с вероятностью $1 - A$)	Следующий по порядку блок
C	Позиция блока, в которую должен перейти транзакт (с вероятностью A)	Ошибка

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

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

ПРИМЕР 1.

TRANSFER ,PRIB ; при каждом входе транзакт будет направляться к блоку с меткой **PRIB**.

ПРИМЕР 2.

TRANSFER .3,PRIB_1,PRIB_2 ; транзакт с вероятностью 0,3 будет направляться к блоку с меткой **PRIB_2** и с вероятностью 0,7 к блоку с меткой **PRIB_1**.

ПРИМЕР 3.

TRANSFER .3,,PRIB_2 ; транзакт с вероятностью 0,3 будет направляться к блоку с меткой **PRIB_2** и с вероятностью 0,7 к следующему по порядку блоку.

ПРИМЕР 4.

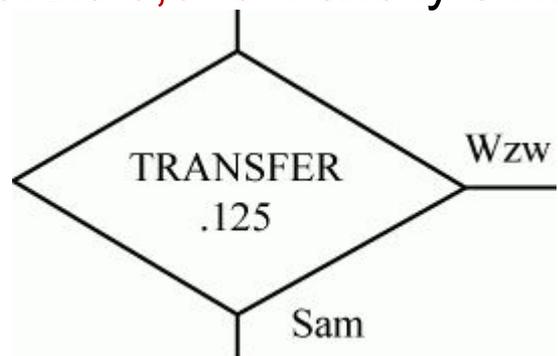
TRANSFER BOTCH,PRIB_1,PRIB_2 ; транзакт будет вначале направляться к блоку с меткой **PRIB_1**, а если он занят, то будет далее пытаться войти в блок с меткой **PRIB_2**. Если оба блока заняты то он будет ждать до тех пор пока один из блоков не освободится.

БЛОК-ДИАГРАММА

Транзакт будет направлен к блоку с меткой **OPER**



Транзакт с вероятностью **0,125** будет направлен к блоку с меткой **Wzw** и с вероятностью **0,875** к блоку с меткой **SAM**.



Параметры транзактов

Параметры транзактов – это свойства транзакта, определяемые пользователем. Множество параметров транзакта – набор стандартных числовых атрибутов, которые принадлежат транзакту. Параметры транзакта являются локальными переменными, которые доступны только данному транзакту.

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

Особенности параметров транзактов:

1. Доступ к параметрам транзактов осуществляется таким образом:

P<номер> или **P\$<имя>**,

где **P** – СЧА транзакта, определяющий его групповое имя, т.е. имя всех параметров транзакта.

2. Номера (имена) конкретных членов множества параметров задаются с помощью целых чисел 1, 2, ... или символьных имен.

Например, **P22** – это 22-параметр транзакта,

P\$COLOR – параметр с именем **COLOR**.

3. При входе транзакта в модель **начальное значение** всех его параметров устанавливается в **ноль**.

4. Значения параметров транзактов и их изменение определяет пользователь.

5. Значениями параметров транзактов могут быть любые числа.

6. Транзакт может обращаться только к своим параметрам.

7. Параметры можно использовать в качестве операндов блоков и в качестве аргументов функций.

Изменение значений параметров транзактов. Блок Assign (Назначить)

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

Формат блока:

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

Операнд	Значение	Результат по умолчанию
A	Номер или имя модифицируемого или задаваемого параметра	Ошибка
B	Величина, используемая для модификации (число или СЧА)	Ошибка
C	Имя функции	Не используется

Пример:

ASSIGN 3, 25

3-ему параметру транзакта присваивается значение 25.

ASSIGN 1, FN\$TYPE

1-ому параметру транзакта присваивается значение функции TYPE .

ASSIGN 4-,5

Значение 4-го параметра транзакта уменьшается на 5

Стандартные числовые атрибуты

В языке GPSS атрибуты (свойства) объектов – это СЧА. Каждый объект имеет свой набор СЧА. Доступ к СЧА осуществляется при использовании специальных обозначений этих атрибутов. Имя СЧА состоит из двух частей:

1) **групповое имя** – состоит из одной или двух букв, идентифицирует тип объекта (прибор, очередь) и тип информации о нем (свободен, текущее значение длины очереди);

2) имя конкретного члена группы.

<Групповое имя><Число> или <Групповое имя>\$<Имя>

Таблица 1-СЧА устройств

Обозначение	Значение
Fj или F\$имя	Показатель занятости устройства (0 – если не занято, 1 – если занято)
FCj или FC\$имя	Число занятий устройства
FRj или FR\$имя	Нагрузка устройства, выраженная в долях тысячи
FTj или FT\$имя	Целая часть значения среднего времени задержки транзакта в устройстве
FVj или FV\$имя	Флаг готовности устройства к использованию (1 – готово, 0-в противном случае)

Таблица 2 - СЧА многоканальных устройств

Обозначение	Значение
R_j или R\$имя	Емкость незаполненной части МКУ
S_j или S\$имя	Емкость заполненной части МКУ
SA_j или SA\$имя	Целая часть среднего заполнения МКУ
SC_j или SC\$имя	Счетчик числа входов в МКУ. (При каждом выполнении блока ENTER значение счетчика увеличивается на значение операнда B этого блока)
SM_j или SM\$имя	Максимально занятая емкость МКУ. Запоминает максимальное значение S_j (S\$имя)
SR_j или SR\$имя	Нагрузка МКУ, выраженная в долях тысячи
ST_j или ST\$имя	Целая часть среднего времени пребывания транзакта в МКУ
SV_j или SV\$имя	Флаг готовности МКУ _j к использованию (1 – готово, 0 – не готово)

Таблица 3 - СЧА очередей

Обозначение	Значение
Qj или Q\$имя	Текущее значение длины очереди (текущее содержимое)
QAj или QA\$имя	Целая часть среднего значения длины очереди
QCj или QC\$имя	Число входов в очередь. При каждом входе в блок QUEUE очереди значение QCj (QC\$имя) увеличивается на значение операнда B , при каждом входе в блок DEPART очереди значение QCj (QC\$имя) уменьшается на значение операнда B
QMj или QM\$имя	Максимальное значение длины очереди (максимальное значение Qj (C)\$имя))
QTj или QT\$имя	Целая часть среднего времени пребывания в очереди всех транзактов, которые входили в очередь (включая и те, которые не ждали – нулевые входы)
QXj или QX\$имя	Целая часть среднего времени пребывания в очереди для транзактов, которые ждали в очереди (ненулевые входы)
QZj или QZ\$имя	Число нулевых входов в очередь

Примеры СЧА объектов:

FR3– возвращает значение загрузки прибора с номером 3;

FT\$Auto_Master – возвращает среднее время занятия транзактом прибора с именем Auto_Master.

S22 – возвращает количество занятых приборов в многоканальном устройстве с номером 22;

SM\$Kassa_2m – возвращает максимальное количество занятых приборов в многоканальном устройстве с именем Kassa_2m.

Q\$Q_5– возвращает значение длины очереди Q_5.

QZ12 – возвращает количество входов транзактов в прибор, для которых время ожидания в очереди равно нулю.

СЧА системы и транзактов

AC1 – значение абсолютного модельного времени (с момента начала моделирования);

C1 – значение относительного модельного времени (с момента последней команды RESET)

TG1– текущее значение счетчика завершения;

Z1 – свободная оперативная память ЭВМ в байтах.

MP<Число> или **MP\$**<Имя> – транзитное время транзакта (абсолютное модельное время минус значение, содержащееся в параметре <Число> или <Имя>);

PR– приоритет транзакта;

M1– резидентное время транзакта(абсолютное модельное время минус время появления транзакта в модели)

Графическое отображение блоков GPSS

