Министерство общего и профессионального образования Российскрй Федерации

Государственный Университет Управления

Институт информационных систем управления

Кафедра информационных систем

Утверждено

первым проректором ГУУ

проф. Ю.Л. Старостиным

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению лабораторных работ

по дисциплине

«Имитационное моделирование экономических процессов»

для студентов специальности

“Прикладная информатика в управлении” - 351400

Москва 2003

# Цели и порядок выполнения практических занятий

Практические занятия имеют целью закрепление теоретических основ дисциплины "Имитационное моделирование экономических процессов" и освоение студентами современной технологии имитационных исследований, приобретение практических знаний по разработке и исследованию имитационных моделей при работе на базе системы моделирования GPSS World.

Практические занятия проводятся по четырем темам:

1. Построение концептуальных моделей сложных систем
2. Инструментальные и технологические возможности современных систем моделирования
3. Освоение языка имитационного моделирования GPSS
4. Планирование и проведение направленных вычислительных экспериментов на имитационной модели.

Практические занятия по темам 2-4 посвящены освоению языка и системы моделирования GPSS World и построены в форме обучающих компьютерных уроков (уроки 1-7). При выполнении компьютерных уроков следует ознакомиться с лекционным материалом, дополнительной литературой [1-6] и приводимыми ниже пояснениями к компьютерным урокам. Используя настоящие методические указания, студент выполняет уроки 1-8, построенные в виде компьютерных сценариев. В компьютерных уроках слева приведено содержание сценария, а справа - последовательность команд системы моделирования, вводимых студентом с клавиатуры компьютера. Пояснения к компьютерным урокам необходимы для ознакомления с основными командами системы моделирования и использования их в процессе создания и верификации имитационной модели, а также при выводе выходной статистики и обработке результатов имитационного эксперимента.

По результатам изучения теоретических и практических курсов студент самостоятельно выполняет разработку и исследование имитационной модели экономической системы. В работе должно быть отражено содержание выполненных исследований на всех технологических этапах имитационного моделирования. Имитационное моделирование включает технологические этапы, представленные ниже:

1.Формулирование проблемы: описание исследуемой проблемы и определение целей исследования.

2.Разработка концептуальной модели: логико-математическое описание моделируемой системы в соответствии с формулировкой проблемы.

3.Формализация имитационной модели.

4.Создание имитационной модели средствами системы моделирования.

5.Подготовка исходных данных моделирования.

6.Испытание и исследование имитационной модели ( оценка адекватности, верификации; анализ устойчивости, чувствительности, точности результатов моделирования)

7.Планирование и проведение направленного вычислительного эксперимента на имитационной модели.

8.Анализ и интерпретация результатов моделирования.

9.Выводы и принятие решений.

# Тема 1. Разработка концептуальной модели

Студенты самостоятельно выполняют концептуальное описание моделируемой системы в соответствующей проблемной отрасли (например: системы массового обслуживания, производственные системы, склад и т.д.).

На первом этапе составляется содержательное описание объекта моделирования, формулируется проблема и обосновываются цели моделирования.

В результате изучения проблемы обосновывается сущность и место исследуемой проблемы, выявляется полное множество значащих факторов. Важно четко *сформулировать и обозначить проблему,* стоящую перед исследователем. Далее исследуются объективные условия решения проблемы и *обосновываются цели моделирования* и задачи исследования. Выполняется системная классификация задач моделирования. *Обосновывается выбор метода моделирования*. Принимается решение о целесообразности применения метода имитационного моделирования. Приводится описание альтернатив (управленческих решений), которые будут проиграны на модели.

Определяется и изучается объект моделирования, те стороны его функционирования, которые представляют интерес для исследования. Результатом работы на этом этапе является *содержательное описание* объекта моделирования с указанием целей моделирования и аспектов функционирования объекта моделирования, которые необходимо изучить на имитационной модели.

На этапе составления концептуального описания осуществляется переход от реальной системы к ее логической (алгоритмической) схеме функционирования системы, составляется логико-математическое описание моделируемой системы в соответствии с формулировкой проблемы, описание объекта в терминах математических понятий и алгоритмизация функционирования ее компонент.

В ходе составления концептуального описания устанавливается основная структура модели, выделяются подсистемы (существенные элементы) и описывается характер их взаимодействия. Системный аналитик выполняет описание границ моделируемой системы, обсуждается уровень детализации моделируемых процессов. Статическое описание моделируемой системы выполняется в ходе *структурного анализа* моделируемого объекта (может быть представлено в виде функциональных схем и т.п.) Динамическое описание системы позволяет построить *функциональную модель* объекта.

Концептуальное описание моделируемой системы включает следующий состав документации:

- подробная постановка задачи (содержательное описание реальной системы и проблемной ситуации, обоснование целесообразности построения модели, формулировка целей моделирования);

- концептуальное описание объекта моделирования (перечисление основных гипотез, выдвинутых при построении модели);

- функциональная схема объекта моделирования;

- список параметров и переменных модели;

-состав выходных статистик моделирования и соотношения для критериев эффективности;

- функциональные зависимости, используемые в концептуальной модели.

# Тема 2. Основные команды системы моделирования GPSS World. Построение GPSS модели.

#### **Пояснения к урокам 1-2**

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

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

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

Система моделирования использует сеансный метод работы с программой. В одном сеансе можно работать с текущей моделью, которая может быть сохранена в файле с помощью команды SAVE.

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

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

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

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

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

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

Для наблюдения за процессом моделирования применяются различные динамические окна:

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

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

В компьютерных уроках 1,2 мы освоим следующие

##### основные команды системы моделирования

**START A, B, C, D**

Установление счетчика завершений и запуск процесса моделирования

1. счетчик завершения.
2. операнд вывода данных: NP – нет вывода данных, Null - вывод стандартного отчета.
3. не используется.
4. вывод списков: 1 - включение CEC/FEC в стандартный отчет, Null.

**HALT**

Прерывает процесс моделирования и сбрасывает очередь команд. Срочная команда.

**CLEAR A**

Сбрасывает статистику процесса моделирования и удаляет из модели все транзакты.

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

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

**RESET**

Сбрасывает статистику процесса моделирования. Отмечает начало периода измерений.

**REPORT A, B**

Устанавливает имя файла отчета и запрашивает немедленный отчет

A – должен быть Null

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

**SHOW X**

Вычисляет выражение X и отображает результат. Срочная команда.

**EXIT A**

Выход из системы моделирования.

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

## УРОК 1

***Основные команды системы моделирования: создание и редактирование программы- имитатора***

* Запустите GPSS World *C: / Program Files/ Minuteman Software/ GPSS World*
* Загрузите программный файл *File/ Open……SAMPLE 1*

Sample.gps в главное окно

Main Window

* Сохраните программу в файле *File/ Save as TMP.gps*
* Pедактируйте файл с помощью *Search/ Find (Replace)…Go to line*
* (вставьте блок BUFFER
* после блока GENERATE) *Edit/ Insert Line …BUFFER*
* Откройте окно блоков *Window/Simulation Window/ Blocks Window*
* Cнимите статистику в окне *View/ Entity Details*

блоков

* Запустите имитацию *Command/ Create Simulation*

*Command/ START 300,NP*

* Наблюдайте в окне блоков прогон *Window/Simulation Window/ Blocks Window*
* Работа со стандартным
* отчетом в Report Window:
* Установить требуемый набор статистики: *Edit/ Setting (Reports)*
* распечатать: *File/ Print*
* Наведите порядок с окнами *Window/Tile, Window/Cascad*
* Сбросить статистику *CLEAR*
* Выход, завершение сеанса *File/Exit*

## УРОК2

***Построение графиков. Динамическое представление переменных (СЧА).***

* Запустите систему GPSS World *File/Open/Sample1.gps*

и пример SAMPLE 1

* Транслируйте модель *Command/Create Simulation*
* Откройте окно для графика и *Window/ Simulation Window/ Plot* *Window*

введите переменные *Label: Очередь*

*Expression: Q$ Barber*

*Title: Очередь к парикмахеру*

* Нажмите кнопку *Plot*

затем *Memorize*

* Запустите имитацию и *Command/ START 100*

наблюдайте вPlot Window

* Прервите имитацию *Command/ Halt*
* И продолжите *Command/ Continue*
* Используйте команду SHOW *Command/SHOW*

для вывода значений переменных *…C1* *…QM$BARBER*

(наблюдайте в Journal Window) .. *1234#5678/345#(67+78)*

*…RN1000*

* Выведите значения переменных *Window/ Simulation* *Window/Expression Window*

в Expression Window *Label: Активный транзакт*

*Expression:* *XN1*

* Нажмите кнопку *View*

затем *Memorize*

* Из списка Memorized Expressions

выберите выражение Q$Barber

и нажмите *View*

* Продолжите имитацию *Continue*
* Наблюдайте статистику в

динамических окнах

* И прервите *Halt*
* (Сбросить статистику) *Command/ RESET (CLEAR)*
* Откройте окно устройств *Window/ Simulation* *Window* */FacilitiesWindow*
* И запустите имитацию *Command/START 1000000,NP*
* Наиболее часто употребляемые *Edit/Setting/ Function Keys*

команды можно зашить в

функциональные клавиши

* Создайте отчет немедленно *Command/Custom*

*REPORT,NOW*

* Остановите имитацию *Halt*
* Сбросьте статистику *RESET или CLEAR*
* Выход *File/Exit*

# 

# Тема З. Освоение языка моделирования GPSS. Основные функциональные объекты GPSS.

**Пояснения к урокам 3-6**

**Содержание базовой концепции структуризации языка моделирования GPSS, описание функциональной структуры GPSS, основных функциональных и статистических объектов, соответствующих моделирующих блоков и стандартных числовых атрибутов приведено в приложениях к настоящим методическим указаниям, которыми следует руководствоваться при выполнении компьютерных уроков 3-6.**

Любая команда на языке GPSS выглядит следующим образом:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № строки (не обязательно) | Метка | Глагол | Операнды (зависят от глагола) | ;комментарии |

В компьютерных уроках используются приемы и команды ручного моделирования для анализа внутренней логической структуры системы моделирования, которые в дальнейшем могут быть полезны для целей верификации и трассировки имитационной модели.

Основные команды ручного моделирования:

**STOP** А, В,С

Устанавливает или снимает условие остановки.

А – номер транзакта

В – номер блока.

C – ON или OFF

###### STEP А

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

А – количество входов в блок.

###### CONTINUE

Продолжает процесс моделирования

Для анализа внутренней логики работы системы моделирования в уроке используются списки текущих и будущих событий, состояние которых отражено в динамических окнах «Кадр СТС» и «Кадр СБС» (см. Приложение 3, пунк 1.3.).

Для анализа выходной статистики в интерактивном режиме по функциональным объектам GPSS используют следующие графические окна: окно устройств *Fasilities* *Window,* окно памяти *Storage Window*, а также средства построения гистограмм **TABLE, QTABLE** (см.Приложение 3, пункт 1.7**.)** и их отображения *Table Window***.**

.

## УРОК 3

***Транзакты. Блоки GENERATE и TERMINATE.***

* Создайте новый *File/New*

программный файл GENERATE 60 ;Создание нового транзакта

TERMINATE 1 ;Уничтожение транзакта

* И сохраните *File/Save as... MYMODEL*
* Посмотрите полученную *Window/Similation Window/ Block Window*

модель в окне блоков

* Транслируйте модель *Command/Create Simulation*

*START 100*

и немедленно получите *Command/Custom*

### отчет REPORT, NOW

* Сбросьте статистику *CLEAR*

и модельное время

* Задайте с помощью средств *Command/Custom*

ручной имитации *STOP*

STOP-условия

* И проводите имитацию по *Command/Create Simulation*

шагам (трассировочные *STEP*

сообщения наблюдайте в

Journal Window)

* Наблюдайте на каждом

шаге списки текущих *Window/Simulation Snapshot/FEC Snapshot*

и будущих событий *Window/Simulation Snapshot/ CEC Snapshot*

(CEC, FEC)

* Используйте команду SHOW *Command/*SHOW

для просмотра переменных *…AC1*

*…TG1*

* Или динамические *Window/Simulation Windows/Expressions*

окна переменных *Время Активный транзакт*

*AC1 XN1*

*(View Memorize)*

* Можно использовать *Window/Simulation Snapshot/User Stops*

также окно User Stops *(View/Settings)*

для задания стоп-условий

и трассировoчных

сообщений

* Снимите стоп- условия *STOP,OFF*
* Продолжите прогон *CONTINUE*

## 

## Урок 4

***Транзакты. Блок ADVANCE.***

* Редактируйте программу *Edit/Insert Line ADVANCE 61*

третьего урока: *GENERATE 60 ;Покупатель приходит*

*ADVANCE 61 ;Покупатели ходят по магазину 61 сек*

*TERMINATE 1 ;Покупатель уходит*

* Транслируйте новую модель *Command/Retranslate*
* Запустите имитацию *Command/*
* Аналогичо уроку 3 *…………… /Custom* *START 1000*

задайте стоп-условия *…………… /STOP*

* И проводите имитацию *Command/ START 1000*

по шагам (команда-STEP) ……..*HALT*

(можно команды зашить ……..*STEP*

в функциональные *View/Settings/Function Keys*

клавиши)

* Наблюдайте списки и *Window/SimulationWindow/*

статистику в *…Blocks Window*

динамических окнах *…Expressions Windo*w

*Window/Simulation Snapshort/FEC*

*…CEC*

* Можно использовать также *Command/Custom*

команду TRACE для *TRACE*

получения трассировочных

сообщений в JournalWindow

* Снимите STOP-условия *STOP, OFF*
* Закончите имитацию *…CONTINUE*
* Сохраните отчет:

1 способ *закройте Report Window*

2 способ *File/ Open Report*

(распечатать) *Print sample x.1.1)*

* Сбросьте статистику *Command/RESET (CLEAR)*
* Выход *File/Exit*

## 

## Урок 5

***Устройства. Блоки SEIZE и RELЕASE.***

* Создайте модель *File/New*

*GENERATE 20 ; Покупатель приходит*

*SEIZE SALESMASN ;Покупатель начинает обслуживаться продавцом*

*ADVANCE 4 1;Покупатель разговаривает с продавцом*

*RELEASE SALESMAN ;Покупатель перестает общаться с продавцом*

*TERMINATE 1; Покупатель уходит*

* Транслируйте *Command / Create Simulation*
* Откройте окно блоков *Window/ Simulation Window/Blocks window*
* Запустите имитацию …*START 1000*
* Откройте окно устройств *Window/Simulation Window/Facilities Window*
* Остановите имитацию ...*HALT*
* Анализируйте статистику *View/Entity Details*
* Моделируйте по шагам *…..STOP ….. STEP*
* Посмотрите списки *Window/Simulation Snapshort/CEC, FEC*
* Создайте отчет и *Command/Custom…REPORT, NOW*

посмотрите

* Сброс *RESET*
* Выход *File/Exit*

## 

## Урок 6

Задание:

*Адаптируйте пример урока 5, введя вместо обслуживающих устройств памяти. Повторите эксперимент.*

## Урок 7

***Очереди. Блоки QUEUE, DEPART. Сбор статистики.***

* Откройте файл *File/Open SAMPQUE.gps*

1 способ

* Запустите имитацию, *Command/Create Simulation*
* наблюдайте статистику *Window/Simulation Window*
* по очередям в QueuesWindow */Queues Window*

*Command/START 1111111*

*(View/Entity Details)*

2 способ

* Прервите имитацию *HALT CLEAR*
* Вставьте в начало *Edit*……

программы *Waittime QTABLE Barber,2,3,20; Wait times Table*

* Транслируйте *Command/Retranslate*

*Command/START 11111111*

* Наблюдайте статистику *Window/Simulation Window*

в Table Window и …*/Table Window*

других окнах *…/Block Window*

*…/Facilities window*

* Выход *Filе/Exit*

**Задание по теме: «Моделирование систем массового обслуживания общего типа в системе моделирования GPSS World».**

ЗАДАНИЕ 1. *Моделирование типовой СМО* (по результатам лабораторного практикума в среде моделирования GPSS World). Изучите учебные модели SAMPLE 1- SAMPLE 9.

Разработайте имитационную модель одноканальной (многоканальной) системы массового обслуживания с очередью. Время между прибытиями посетителей распределено экспоненциально (со средним Х). Время обслуживания распределено равномерно (известно минимальное значение Хмин , максимальное значение Хмакс).

Вариант 1. Смоделировать работу 1000 клиентов,

Вариант 2. Использовать таймер: GENERATE 30

TERMINATE1

Табулировать количество обслуженных посетителей.

Продумать состав и форму представления выходной статистики. Использовать интерактивные окна для построения графиков (Plot Window), средства построения гистограмм (Table Window), и вывода значений переменных (Expression Window), а также Окно устройств (Facilities), и Окно Памяти (Storage Window) и др.

ЗАДАНИЕ 2. *Моделирование СМО общего типа.*

Адаптировать разработанную имитационную модель для следующих случаев:

Вариант 1. Входной поток заявок неоднородный. Например, на автостоянке заправка автотранспорта производится бензином нескольких типов. Время обслуживания зависит от характеристик заявки.

Моделировать обобщенный поток заявок. Изменение режима прихода и обслуживания заявок осуществляется с использованием блока TRANSFER (вероятностное развитие ситуации).

Вариант 2. СМО с отказами.

А) Моделировать нетерпеливого клиента. Если время ожидания в очереди превышает допустимое, клиент покидает систему необслуженным.

Б) Если длина очереди превышает допустимую (например, нехватка места при парковке) клиент покидает систему необслуженным.

Табулировать количество отказов в обслуживании. При описании можно использовать блок TEST.

B) Моделировать ситуацию, когда объект обслуживания закрывается на технологический перерыв (обед, профилактику и т. п.)

С) Обслуживающее устройство выходит из строя (отказ в обслуживании по причине потери работоспособности). Время между отказами и время восстановления распределено экспоненциально.

Табулировать общее количество обслуженных заявок и количество отказов.

Моделируемый процесс в последнем случае включает описание двух процессов:

* процесса обслуживания,
* процесса, связанного с отказами (прерывание устройства через случайное время безотказной работы) и восстановлениями обслуживающего устройства (снятие прерывания после случайного времени восстановления).

# Тема 4. Направленный вычислительный эксперимент на имитационной модели. Обработка результатов эксперимента. Процедура ANOVA.

Вычислительная процедура ANOVA использует статистический метод – дисперсионный анализ для анализа результатов вычислительного эксперимента. Она возвращает средние значения, доверительные интервалы и др. оценки по каждой выборке, составленной по результатам имитационных прогонов, а также расчетное значение F-теста, которое может быть использовано для сравнения двух альтернатив, составленных на основе выборочных данных по результатам имитационного эксперимента.

## Урок 8

Процедура ANOVA

* Ознакомьтесь с примером *File / Open/ANOVA.gps*
* Ознакомьтесь или

подготовьте INCLUDE-файл

«ctlanova.txt» для

проведения эксперимента:

* RESULTS MATRIX, 2,3 *; Создать 3 реплики на двух уровнях*

*;(матрица для хранения результатов ;эксперимента)*

* Cut\_Time EQU 6.8 ;*задаем значение управляемого параметра*
* Treatment EQU 1 ;*- уровень 1*
* RMULT 411 ; *задаем начальные значения ДСЧ*
* Start 100,NP ;*прогон №1*
* MSAVEVALUE RESULTS, 1,1, ;*сохраняем результаты прогона по отклику*

QT$Barber ;*QT $Barber*

* Clear off ;*очистка статистики*
* RMULT 421 ;*меняем начальные значения ДСЧ*
* Start 100,NP ;*прогон №2*
* MSAVEVALUE RESULTS, 1,2, ;*и т.д.*

QT$Barber

* Clear off
* RMULT 431
* Start 100,NP ;*прогон №3*
* MSAVEVALUE RESULTS, 1,3,

QT$Barber

* Clear off
* Cut\_Time EQU 2 ;*изменяем значение* *управляемого параметра*
* Treatment EQU 2 ;*- уровень 2*
* RMULT 411
* Start 100,NP ;*прогон №1*
* MSAVEVALUE RESULTS, 2,1, ;*сохраняем результаты* *прогона по отклику*

QT$Barber ;*QT$Barber*

* Clear off
* RMULT 421
* Start 100,NP ;*прогон №2*
* MSAVEVALUE RESULTS, 2,2,

QT$Barber

* Clear off
* RMULT 431
* Start 100,NP ;*прогон №3*
* MSAVEVALUE RESULTS, 2,3,

QT$Barber

* Транслируйте *Command / Create Simulation*
* Выполните эксперимент) *Command / Custom*
* (автоматически*) INCLUDE “ctlanova.txt” (OK)*
* Наблюдайте результаты эксперимента *Window / Simulation Window /*

*Matrix Window*

* Проанализируйте результаты *Command / Custom*

статистического анализа в окне «Журнал», *SHOW ANOVA (Results,2,0)*

выполненные процедурой ANOVA

(доверительный интервал; F-статистика)

# ПРИЛОЖЕНИЕ 1.

# Базовая концепция структуризации

# языка моделирования GPSS

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

Каждый язык моделирования содержит абстрактные конструкции в понятиях, близких к алгоритмической (концептуальной или формальной) схеме описания моделируемых систем. В рамках этой схемы четко классифицируются элементы. Элементы различных классов различают по характеристикам и свойствам. Описываются связи между элементами системы и внешней средой.

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

**Содержание базовой концепции структуризации языка GPSS**

**1)** Это блочно-ориентированная концепция, разработанная с ориентацией на описание СМО (СМО – система массового обслуживания) (см. Приложение 2).

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

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

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

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

**2)** GPSS – система дискретного типа.

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

GPSS позволяет описывать процессы с дискретными событиями.

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

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

В GPSS рассматриваются 2 класса событий:

* *основные* (те события, которые можно запланировать, то есть рассчитать момент их появления заранее, до их наступления, например момент появления заявки на входе)
* *вспомогательные* (те события, которые происходят вследствие появления основных событий. Вспомогательные события осуществляются в результате взаимодействия таких абстрактных элементов как блоки и транзакты, например, смена состояния прибора обслуживания со «свободен» на «занято»)

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

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

**Функциональная структура GPSS**

рассматривается на двух уровнях.

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

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

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

**1 уровень**

##### **Аппаратно ориентированные объекты:**

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

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

* возможны их задержки в некоторых точках модели (связанные с обслуживанием, ожиданием в очереди),
* изменение маршрутов и направления движения,
* создание копии транзактов и др.

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

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

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

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

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

* **Памяти** – предназначены для моделирования объектов, обладающих ёмкостью. Аналогия с многоканальными СМО - память может обслуживать одновременно несколько транзактов. При этом транзакт занимает определённую часть памяти.
* **Логические переключатели** – принимают значение включено/выключено, позволяют изменять пути следования транзактов в модели.

Устройства, памяти, логические переключатели относятся к **аппаратно-ориентированным объектам GPSS.**

#### **Статистические объекты GPSS**: используются тогда, когда надо собирать cтатистику:

* **Очереди.** В процессе движения транзакты могут задерживаться в определенных точках модели. Если необходимо собирать информацию о длине очереди транзактов и времени задержки транзактов используют соответствующие статистические объекты.
* **Таблицы.** Таблицы обрабатывают статистическую информацию, строят гистограмму распределений по любой переменной.

#### **Вычислительные объекты GPSS:**

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

**2 уровень**

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

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

*GPSS является системой интерпретирующего типа с собственным языком.*

Таким образом, составляется на языке GPSS и реализуется *функциональная блок-схема.*

Существуют 2 особых блока: GENERATE, имеющий только выход, через него транзакты входят в модель, и блок TERMINATE, имеющий только вход – удаляет транзакты из модели. Любой процесс на языке моделирования GPSS имеет вид:

GENERATE

(Блоки модели)

TERMINATE

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

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

В настоящее время появились различные обобщения рассмотренной концепции структуризации, когда структура моделируемого процесса изображается в виде потока, проходящего через обслуживающие устройства и другие элементы СМО: сети очередей, графы потоков, структурно-стохастические графы и др. Дуги на графах интерпретируются как потенциальные потоки заявок между обслуживающими устройствами. Пути на графах соответствуют маршрутам движения заявок в системе обслуживания.

# ПРИЛОЖЕНИЕ 2.

# Экскурс по Системам Массового Обслуживания.

***Система массового обслуживания (СМО)*** *– объект, в котором выполняется последовательность операций, включает совокупность приборов обслуживания, которые связаны определенным логическим порядком. В соответствии с этой логикой происходит движение материальных носителей – заявок на обслуживание от канала (ОУ) к каналу (ОУ).*

*Структура систем массового обслуживания* представляется в следующем виде*:*

*Входной Обслуживающее Выходной*

*поток заявок Очередь устройство поток заявок*

*источник поглотитель*

*заявок заявок*

Заявка характеризуется моментом появления на входе системы, статусом по отношению к другим заявкам, некоторыми параметрами, определяющими потребности во временных ресурсах на обслуживание

Постоянно поступающие заявки на обслуживание образуют *поток заявок – совокупность заявок, распределенную во времени.*

Поток заявок может быть однородным (с точки зрения обслуживания все заявки равноправны) и неоднородным.

Основной параметр потока заявок – промежуток времени между моментами поступления 2-х соседних заявок.

Поток заявок может быть стационарным и нестационарным (например, изменяться от времени суток).

Поток заявок рассматривается как *случайный процесс, характеризующийся функцией распределения периода поступления заявок* (например, простейший – стационарный, ординарный, поток без последействия, поток Эрланга).

Элемент системы, в котором происходят операции, называется обслуживающим устройством. В момент выполнения операций он занят, иначе - свободен. Если ОУ (канал) свободен, то заявка принимается к обслуживанию.

Обслуживание каждой заявки каналом означает задержку в нем заявки на время, равное периоду обслуживания. После обслуживания заявка покидает прибор обслуживания. Таким образом, *ОУ характеризуется временем обслуживания заявки (время занятости канала).*

При случайном характере поступления заявок образуются *очереди.*

## Заявки принимаются к обслуживанию

* в порядке очереди (FIFO, очереди с приоритетами и др.),
* в случайном порядке в соответствии с заданными распределениями, по минимальному времени получения отказа
* и др.

Реальный процесс функционирования СМО следует представлять в виде последовательности фаз обслуживания, выполняемых различными устройствами. Примеры многофазного обслуживания: обслуживание покупателей в магазине (прилавок, касса; технологический процесс – обработка деталей на станках). Причем эти многофазные системы могут иметь сложную структуру (стохастические сети).

Обслуженная заявка покидает прибор обслуживания и покидает систему (поглотитель заявок), либо движется дальше в соответствии с технологической схемой работы системы.

*Типы СМО:*

* *с ожиданием*
* *без ожидания* (с отказами)
* *с ограничением на длину очереди (или с ограниченным ожиданием)*
* *без ограничения;*
* *с упорядоченной очередью*
* *с неупорядоченной очередь*
* *с приоритетами*

*- без приоритетов.*

Любая модель строится для того, чтобы оценить какие-то показатели качества.

*Показатели качества обслуживания:*

* *общее количество обслуженных заявок за какой- либо промежуток времени,*
* *пропускная способность – среднее число заявок, обслуженных в единицу времени,*
* *доля заявок обслуженных,*
* *доля заявок, получивших отказ,*
* *время пребывания заявки в системе (от момента поступления заявки в систему до момента завершения ее обслуживания),*
* *среднее время обслуживания (функция распределения времени обслуживания),*
* *средняя длина очереди,*
* *среднее время ожидания*
* *загрузка каналов - коэффициент использования (как доля времени, в течение которого ОУ было занято) – характеризует степень простоя ОУ.*

Чем аппарат аналитического моделирования СМО (теория массового обслуживания) отличается от имитационного моделирования?

Аналитические методы весьма стеснительны для решения практических задач: например, выдвигается предположение о простейшем потоке заявок (для разных фаз обслуживания он может быть не простейшим), однотипных устройствах и т.д .

В имитационном моделировании все ограничения снимаются (например, могут использоваться произвольные законы распределения для описания временных параметров, различные схемы (порядок обслуживания) и т.д., объекты исследуется не обязательно в стационарном режиме (например, возможно изучение переходного режима, когда показатели отличаются от асимптотических значений).

### Сущность метода имитационного моделирования для СМО:

Используются специальные алгоритмы, позволяющие вырабатывать случайные реализации потоков событий и моделировать процессы функционирования обслуживающих систем. Далее осуществляется многократное воспроизведение, реализация случайных процессов обслуживания и статистическая обработка на выходе - оценка показателей качества обслуживания.

# ПРИЛОЖЕНИЕ 3.

# Функциональная структура GPSS

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

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

**1.1.Блоки**

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

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

TERMINATE

GENERATE

блоки модели

##### СВЯЗАННЫЕ СЧА:

N$j - общее число входов в блок j ;

W$j - текущее число транзактов в блоке j.

**1.2.Транзакты**

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

МОДЕЛИРУЮЩИЕ БЛОКИ:

Основные блоки:

**GENERATE** A,B,C,D,E

Вводит транзакты в модель.

А - среднее значение интервала времени (необязательный операнд);

В - разброс или модификатор среднего значения (необязательный операнд);

С - время появления первого транзакта (необязательный операнд);

D - общее число генерируемых транзактов (необязательный операнд);

Е - уровень приоритета каждого транзакта (необязательный операнд).

**TERMINATE** А

Удаляет активный транзакт из процесса моделирования.

А – величина уменьшения счетчика завершения (необязательный операнд).

**ADVANCE** А,В

Задерживает транзакт.

А - среднее время задержки;

В - разброс или модификаторсреднего значения (необязательный операнд).

Блоки, изменяющие значения параметров транзактов:

**ASSIGN** А,В,С

Изменяет значение параметра транзакта.

А - номер изменяемого параметра;

В - новое значение параметра;

С – номер функции (необязательный операнд).

**INDEX** А,В

Обновляет параметр активного транзакта.

А - номер параметра;

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

**MARK** А

Записывает в параметр активного транзакта значение абсолютного модельного времени.

А – номер параметра (необязательный операнд).

**PRIORITY** А,В

Устанавливает приоритет активного транзакта.

А – новое значение приоритета;

В – BU (необязательный операнд).

Блоки, управляющие движением транзактов:

**GATE** 0 А,В

Изменяет маршрут движения транзактов в зависимости от состояния некоторого объекта.

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

А - номер проверяемого объекта;

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

**LOOP** А,В

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

А – параметр, содержащий число;

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

**TEST** 0 А,В,С

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

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

А – проверяемое значение;

B – контрольное значение;

С – номер блока назначения (необязательный операнд).

**TRANSFER** А,В,C,D

Обеспечивает переход активного транзакта к новому блоку.

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

B - номер или метка блока (необязательный операнд);

C – номер или местоположение блока (необязательный операнд);

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

Блоки, работающие с транзактами в списке пользователя:

**LINK** А,В,С

Управляет размещением активного транзакта в списке пользователя.

А - номер списка пользователя;

В – упорядочивание списка;

С – местоположение следующего блока (необязательный операнд).

**UNLINK О** А,В,С,D,E,F

Управляет удалением активного транзакта из списка пользователя.

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

А – номер списка пользователя;

В – номер блока назначения для удаленных транзактов;

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

D – проверяемое значение (необязательный операнд);

E – контрольное значение (необязательный операнд);

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

Блоки, связанные с трассировкой транзактов:

**TRACE**

Устанавливает индикатор трассировки активного транзакта.

**UNTRACE**

Сбрасывает индикатор трассировки активного транзакта.

Блоки, работающие с копиями транзактов:

**ASSEMBLE** А

Ожидает и уничтожает транзакты.

А – счетчик транзактов.

**GATHER** А

Накапливает транзакты, являющиеся членами семейства.

А - счетчик транзактов, которые должны быть накоплены.

**MATCH** А

Пара сопряженных блоков MATCH заставляет транзакты ожидать друг друга.

А – номер блока, который проверяется на наличие транзакта.

**SPLIT** А,В,С

Создает транзакты того же семейства, что и активный транзакт.

А – количество создаваемых транзактов;

В – номер блока (необязательный операнд);

С – номер параметра (необязательный операнд).

СВЯЗАННЫЕ СЧА:

A1 – указатель семейства активного транзакта;

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

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

М1 –время пребывания транзакта в системе;

Pj – значение параметра с номером j;

PR – приоритет активного транзакта;

XN1 - номер активного транзакта.

**1.3. Списки**

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

Существует несколько видов списков:

1) Список будущих событий

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

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

2) Список текущих событий

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

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

3) Список пользователя

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

4) Cписок прерываний

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

5) Список синхронизации

В список синхронизации помещаются транзакты, ожидающие объединения с другими транзактами в блоках GATHER и ASSEMBLE или находящиеся в блоках MATCH. После выполнения условий синхронизации транзакты возвращаются в список текущих событий. Кроме рассмотренных списков, обработка которых происходит без участия программиста, в GPSS/PC существуют списки пользователя, управление которыми осуществляется с помощью блоков LINK и UNLINK.

СВЯЗАННЫЕ СЧА:

CA$j - среднее число транзактов в списке j;

CCSj - общее число входов транзактов в список j;

СН$j - текущее число транзактов в списке j;

CM$j - максимальное число транзактов в списке j;

CT$j - среднее время пребывания транзакта в списке j.

**1.4. Устройства**

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

МОДЕЛИРУЩИЕ БЛОКИ:

**SEIZE** А

Занимает устройство.

А **-** номер устройства.

**RELEASE** А

Освобождает устройство.

А **-** номер устройства (числовое или символьное имя освобождаемого устройства).

**PREEMPT** А

Переводит устройство в прерванное состояние.

А - номер прерываемого устройства.

###### RETURN А

Удаляет транзакт из прерванного устройства.

А **-** номер устройства (числовое или символьное имя освобождаемого устройства).

**FAVAIL** А

Обеспечивает перевод устройства с номером А в доступное состояние

**FUNAVAIL** А,В,C,D,E,F,G,H

Обеспечивает перевод устройства в недоступное для входа транзактов состояние.

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

СВЯЗАННЫЕ СЧА:

F$j - состояние устройства с номером j: О - если устройство свободно, и 1 - если

устройство занято;

FC$j – количество транзактов, занимавших устройство;

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

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

FT$j - среднее время использования устройства j одним транзактом;

FV$j – если устройство находится в доступном состоянии, то1.

**1.5. Памяти**

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

МОДЕЛИРУЮЩИЕ БЛОКИ:

**ENTER** A,B

Помещает транзакт в память.

А - имя памяти символическое или числовое;

В - число занимаемых единиц памяти (необязательный операнд).

**LEAVE** А,В

Выводит транзакт из памяти.

А - номер памяти;

В - число освобождаемых единиц, памяти (необязательный операнд).

**SAVAIL** А

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

**SUNAVAIL** А

Обеспечивает недоступное состояние памяти.

СВЯЗАННЫЕ СЧА:

R$j – емкость неиспользованной памяти j;

S$j - емкость используемой памяти j;

SA$j - среднее заполнение памяти j;

SC$j - число входов в память j;

SE$j - память j пуста;

SF$j - память j заполнена;

SM$j - максимальное заполнение памяти j;

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

SV$j - память j в доступном состоянии;

SR$j - коэффициент использования памяти j . Доля общего использования, представленная средним объемом используемой памяти в памяти j;

**1.6. Очереди**

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

МОДЕЛИРУЮЩИЕ БЛОКИ:

**QUEUE** А,В

Помещает транзакт в конец очереди.

А - номер очереди (числовое или символьное имя очереди);

В - число добавляемых к очереди элементов (необязательный операнд).

**DEPART** A,B

Удаляет транзакт из очереди.

А - номер (имя) очереди;

В - число удаляемых из очереди элементов (необязательный операнд).

СВЯЗАННЫЕ СЧА:

Q$j - текущая длина очереди j;

QA$j - средняя длина очереди j;

QC$j - число входов в очередь j;

QM$j - максимальная длина очереди j;

QT$j - среднее время пребывания в очереди j, включая нулевые входы;

QX$j - среднее время пребывания в очереди j, без нулевых входов.

QZ$j - число входов в очередь с нулевым временем пребывания (транзакт прошел

через блок QUEUE, не задерживаясь в очереди);

**1.7. Таблицы**

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

МОДЕЛИРУЮЩИЕ БЛОКИ:

**TABULATE** A,B

Заносит значение в таблицу.

А – имя или номер таблицы;

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

**NAME TABLE** A,B,C,D

Определяет таблицу.

NAME – метка объекта;

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

В - верхняя граница нижнего интервала гистограммы;

С - ширина интервалов;

D - число интервалов (частотных классов);

###### NAME QTABLE A,B,C,D

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

NAME – метка объекта;

А – имя очереди;

В - верхняя граница нижнего интервала гистограммы;

С - ширина интервалов;

D - число интервалов (частотных классов);

СВЯЗАННЫЕ СЧА:

TB$j - среднее значение фиксируемой в таблице j переменной;

TC$j - число входов в таблицу j;

TD$j - стандартное среднеквадратическое отклонение табулируемой переменной.

# ПРИЛОЖЕНИЕ 4.

# Системные числовые атрибуты

Системные числовые атрибуты являются переменными состояния процесса моделирования и доступны для использования в течение всего процесса моделирования. Они возвращают числовые или строковые значения и могут быть применимы в операндах операторов GPSS и выражениях.

Часть СЧА поддерживаются автоматически, другие требуют вычисления после их вызова. Например, как СЧА доступен счетчик количества входа транзактов в блок. Всего существует свыше 50 классов СЧА.

Следующие СЧА допустимы для использования в операндах и выражениях команд и операторов. Во всех случаях *Еntпит* необходимо заменять одним из спецификаторов объектов. Это может быть имя (перед именем должен стоять символ «$») или номер. В случае косвенной адресации перед именем или номером должен стоять символ «\*».

**Допустимые СЧА**

**A1** – Семейство активного транзакта. Целочисленное значение.

**АС1** - Значение абсолютного модельного времени. (Модельное время с момента последней команды CLEAR). Вещественное значение.

**ВV***Еntпит -* Значение булевой переменной *Епtпит.* Вещественное значение.

**С1** - Значение относительного модельного времени. (Модельное время с момента последней команды RESET). Вещественное значение.

**CA***Еntпит -* Среднее содержимое списка пользователя. Взвешенное по времени среднее количество транзактов, помещенных в список пользователя *Еntпит.* Вещественное значение.

**CC***Еntпит* -Общее число входов транзактов в список пользователя. Число всех транзактов, находившихся в списке пользователя *Епtпит.* Целочисленное значение.

**СН***Еntпит* - Текущее содержимое списка пользователя. Текущее количество транзактов, находящихся в списке пользователя *Еntпит.* Целочисленное значение.

**СМ***Еntпит* - Максимальное содержимое списка пользователя. Максимальное количество транзактов, одновременно находившихся в списке пользователя *Еntпит.* «Верхняя отметка уровня». Целочисленное значение.

**СТ***Еntпит -* Среднее время пребывания одного транзакта в списке пользователя. Средняя продолжительность пребывания транзактов в списке пользователя *Епtпит.* Вещественное значение.

**F***Еntпит* - Состояние устройства. Если в данный момент устройство *Еtпит* занято, *Еntпит* возвращает 1 . Иначе F*Еntпит* возвращает 0. Целочисленное значение.

**FC***Еntпит* -Счетчик использования устройства. Количество транзактов, занимавших устройство *Еntпит* с помощью блоков SEIZE и PREEMPT. Целочисленное значение.

**FI** *Еntпит -* Устройство прервано. Если устройство *Еntпит* в данный момент было занято более приоритетным транзактом, FI*Еп1пит* возвращает 1. Иначе F*Еntпит* возвращает 0. Целочисленное значение.

**FN***Еntпит* - Функция. Результат вычисления значения функции *Еntпит.* Вещественное значение.

**FR***Еntпит -* Коэффициент использования устройства. Отношение времени, в течение которого устройство *Еntпит* было занято, к общему времени моделирования. FR*Еntпит* выражается в долях от тысячи и возвращает значения, изменяющиеся от 0 до 1000 включительно. Может быть нецелочисленным. Вещественное значение.

**FT***Еntпит -* Среднее время удержания устройства. Среднее время удержания устройства *Еntпит* одним транзактом. Вещественное значение.

**FV***Еп1пит -* Устройство доступно. FV*Entпит* возвращает 1, если устройство *Епtпит* в состоянии «доступно», и 0, если устройство в состоянии «не доступно». Целочисленное значение.

**GN***Еntпит* -Счетчик числовой группы. GN*Еntпит* возвращает количество элементов числовой группы *Епtпит.* Целочисленное значение.

**GT***Епtпит -* Счетчик группы транзактов. GT*Entnum* возвращает количество элементов группы транзактов *Епtпит.* Целочисленное значение.

**LS***Еntпит -* Состояние логического ключа. LS*Еntпит* возвращает 1, если логический ключ *Епtпит* в установленном состоянии, и 0, если логический ключ в сброшенном состоянии. Целочисленное значение.

**МВ***Еntпит-* Соответствие в блоке МАТСН. МВ*Епtпит* возвращает 1, если в сопряженном блоке МАТСН присутствует транзакт, принадлежащий тому же семейству, что и активный транзакт. В противном случае МВ*Епtпит* возвращает 0. Целочисленное значение.

**МР***Еntпит* - Транзитное время параметра. Текущее абсолютное модельное время за вычетом значения, содержащегося в параметре *Раrатеtеr.* Вещественное значение.

**МХ***Еntпит*(m,n) - Ячейка матрицы. Возвращает значение ячейки в строке m, столбце n матрицы *Епtпит.* Для матриц, имеющих более 2 измерений, все остальные индексы считаются равными 1 . В отличие от СЧА класса МХ, PLUS-выражения могут обращаться к любому элементу многомерных матриц.

**М1** - Транзитное время. М1 возвращает абсолютное модельное время за вычетом времени входа транзакта в систему. Вещественное значение.

**N***Еntпит* - Счетчик входов в блок. Возвращает общее количество транзактов, которые входили в блок *Епtпит.* Целочисленное значение.

**P***Parameter* или \**Parameter* - Значение параметра. Р*Раrатеtеr* или *\*Раrатеtеr* возвращает значение параметра *Раrатеtеr* активного транзакта. Целочисленное, вещественное или строковое значение. Для косвенной адресации используется вариант *\*Раrатеtеr.*

**РR** - Приоритет транзакта. Значение приоритета активного транзакта. Целочисленное значение.

**Q***Еntпит —* Текущее содержимое очереди. Текущее значение количества транзактов в очереди *Епtпит.* Целочисленное значение.

**QA***Еntпит* - Среднее значение длины очереди. Взвешенное по времени количество транзактов в очереди *Еntпит.* Вещественное значение.

**QC***Еntпит -* Общее количество входов в очередь. Сумма всех входов транзактов в очередь *Епtпит.* Целочисленное значение.

**QM***Еntпит -* Максимальное содержимое очереди. Максимальное количество транзактов, находившихся в очереди *Еntпит.* «Верхняя отметка уровня». Целочисленное значение.

**QT***Еntпит -* Среднее время пребывания в очереди. Взвешенное среднее время пребывания транзактов в очереди *Епtпит.* Вещественное значение.

**QX***Еntпит -* Среднее время пребывания в очереди за исключением нулевых входов. Взвешенное среднее время пребывания транзактов в очереди *Еntпит,* не считая входов со временем пребывания, равным 0. Вещественное значение.

**QZ***Еntпит* *-* Количество входов с нулевым временем пребывания в очереди. Количество входов транзактов, время пребывания в очереди *Епtпит* которых равно нулю. Целочисленное значение.

**R***Еntпит -* Объем свободной памяти. Содержимое памяти *Епtпит,* доступное для входов транзактов. Целочисленное значение.

**RN***Еntпит -* Случайное число. RN*Еп1пит* возвращает случайное целое число от 0 до 999 из генератора случайных чисел *Епtпит.* Целочисленное значение.

**S***Еntпит -* Объем занятой памяти. *SЕп1пит* возвращает количество элементов памяти *Епtпит,* занятых транзактами. Целочисленное значение.

**SA***Еntпит -* Среднее значение занятой памяти. *SAEntпит* возвращает взвешенное по времени среднее количество занятых элементов памяти *Епtпит.* Вещественное значение.

**SC***Епtпит -* Счетчик использования памяти. Общее количество использовавшихся элементов памяти *Епtпит.* Целочисленное значение.

**SE***Еntпит -* Память пуста. SE*Еntпит* возвращает 1,если все элементы ячейки памяти *Еntпит* пусты, и 0, если нет. Целочисленное значение.

**SF***Епtпит -* Память заполнена. SF*Епtпит* возвращает 1, если память *Еntпит* полностью заполнена, и 0, если нет. Целочисленное значение.

***SR****Entnum -* Коэффициент использования памяти. Доля общего использования памяти *Entnum,* представленная отношением среднего количества используемых элементов памяти *Entnum* к общему количеству элементов памяти *Entnum. SREntnum* выражается в долях от тысячи и возвращает значения, изменяющиеся от 0 до 1000 включительно. Может быть нецелочисленным. Вещественное значение.

**SM***Entnum* - Максимальное количество занятых элементов памяти *Entnum.* «Верхняя отметка уровня». Целочисленное значение.

**ST***Entnum*  - Среднее время использования одного элемента памяти *Entnum.* Вещественное значение.

**SV***Entnum* - Память в доступном состоянии. SV*Entnum* возвращает 1, если память *Entnum* в состоянии «доступно», и 0, если память *Entnum в* состоянии «не доступно». Целочисленное значение.

**TB***Entnum* - Среднее значение невзвешенных аргументов таблицы *Entnum.* Вещественное значение.

**TC***Entnum* - Количество невзвешенных аргументов таблицы *Entnum.* Целочисленное значение.

***TD****Entnum -* Стандартное отклонение невзвешенных аргументов таблицы *Entnum.* Вещественное значение.

**TG1** - Текущее значение счетчика завершения. TG1 возвращает значение, уменьшаемое блоками TERMINATE с положительным значением операнда А. Это значение задается оператором START и указывает на завершение процесса моделирования, когда становится равным нулю. Целочисленное значение.

***V****Entnum -* Результат вычисления целочисленной переменной *Entnum* или переменной с плавающей точкой *Entnum.* Вещественное значение.

***W****Entnum -* Текущий счетчик блока. Текущее количество транзактов в блоке *Entnum.* Целочисленное значение.

***X****Entnum -* Ячейка. Возвращает значение ячейки *Entnum.* Целочисленное, вещественное или строковое значение.

**XN1** - Номер активного транзакта. Возвращает номер активного транзакта. Целочисленное значение.

**Z1** - Свободная системная память. Величина, возвращаемая операционной системой. Целочисленное значение.

Особые правила, которые действуют в режиме совместимости с GPSS/PC, будут рассмот­рены далее.

……

# ПРИЛОЖЕНИЕ 5.

# Блоки GPSS

**ADOPT** - Изменяет номер семейства.

**ADVANCE***-* Помещает транзакт в список будущих событий.

**ALTER**- Проверяет и изменяет транзакты в группе.

**ASSEMBLE** - Ожидает и уничтожает связанные транзакты, полученные в результате разделения.

**ASSIGN** - Изменяет параметр транзакта.

**BUFFER** - Помещает транзакт в конец списка текущих событий.

**CLOSE**- Закрывает поток данных.

**COUNT** - Помещает число объектов, удовлетворяющих условию, в параметр транзакта.

**DEPART -** Уменьшает содержимое очереди.

**DISPLACE** - Меняет транзакту следующий по порядку блок в последовательности блоков.

**ENTER** - Занимает элементы памяти или ожидает их освобождения.

**EXAMINE** - Проверяет на принадлежность к группе.

**EXECUTE** - Выполняет действие, определенное заданным блоком.

**FAVAIL** - Изменяет состояние устройства на «доступно».

**FUNAVAIL** - Изменяет состояние устройства на «не доступно».

**GATE** - Проверяет заданное условие и изменяет поток транзактов.

**GATHER** - Ждет связанные транзакты.

**GENERATE** - Создает транзакт и помещает его в список будущих событий.

**INDEX** - Изменяет параметр транзакта.

**INTEGRATION** - Включает/выключает интегрирование переменной пользователя.

**JOIN** - Добавляет новый член в числовую группу или группу транзактов.

**LEAVE** - Освобождает элементы памяти.

**LINK** - Помещает транзакт в список пользователя.

**LOGIC** - Изменяет логический ключ.

**LOOP** - Уменьшает значение параметра и переводит транзакт к другому блоку, если пара­метр не равен нулю.

**MARK** - Помещает в параметр транзакта значение модельного времени.

**МАТСН** - Ожидает, пока транзакт, принадлежащий тому же семейству, не войдет в связанный блок МАТСН.

**MSAVEVALUE** - Присваивает значение элементу матрицы.

**ОРЕN** - Инициализирует поток данных.

**PLUS -** Вычисляет значение РШ5-выражения и сохраняет результат в параметре.

**РRЕЕМРТ** - Вытесняет транзакт, занявший устройство.

**PRIORITY** - Изменяет приоритет транзакта.

**QUEUE** - Увеличивает содержимое очереди.

**READ** - Считывает следующую строку данных из потока данных.

**RELEASE** - Освобождает устройство.

**REMOVE** - Исключает член числовой группы или группы транзактов.

**RETURN** - Освобождает устройство.

**SAVAIL** - Изменяет состояние памяти на «доступно».

**SAVEVALUE** - Присваивает значение ячейке.

**SСАN** - Проверяет группу транзактов, помещает результат в параметр.

**SEEK** - Изменяет указатель строки в потоке данных.

**SEIZE** - Занимает устройство или ожидает его освобождения.

**SELЕСТ**- Помещает номер выбранного объекта в параметр транзакта.

**SPLIT** - Создает связанные транзакты, разделяя исходный.

**SUNAVAIL** - Изменяет состояние памяти на «не доступно».

**ТАВULАТЕ** - Обновляет таблицу.

**ТЕRMINАТЕ** - Уничтожает транзакт, уменьшает счетчик завершения.

**TEST** - Проверяет арифметическое условие и изменяет поток транзактов.

**ТRАСЕ** - Устанавливает индикатор трассировки активного транзакта.

**ТRANSFER** - Перемещает транзакт в указанный блок.

**UNLINK** - Исключает транзакт из списка пользователя.

**UNTRACE** - Снимает индикатор трассировки активного транзакта.

**WRITE** - Передает значение величины в поток данных.

# ПРИЛОЖЕНИЕ 6.

# Команды GPSS WORLD

**BVARIABLE** - Определяет булеву переменную.

**CLEAR** - Сбрасывает статистику и удаляет транзакты.

**CONDUCT**- Выполняет зарегистрированный PLUS-эксперимент.

**CONTINUE** - Продолжает процесс моделирования после остановки.

**EQU** - Присваивает значение переменной пользователя.

**ЕХIТ** - Завершает сеанс работы с GPSS World.

**FUNCTION** - Определяет функцию.

**FVARIABLE** - Определяет переменную с плавающей точкой.

**HALT** - Останавливает процесс моделирования и сбрасывает очередь команд.

**INCLUDE** - Считывает и транслирует дополнительные файлы модели.

**INITIAL** - Инициализирует или изменяет логический ключ, ячейку или матрицу.

**INTEGRATE** -Автоматически вычисляет интеграл для переменной пользователя.

**МАТRIX** - Определяет матрицу.

**QTABLE** - Определяет таблицу статистики очереди (Q-таблицу).

**REPORT** - Устанавливает имя файла отчета или запрашивает немедленный отчет.

**RESET** - Сбрасывает статистику процесса моделирования.

**RMULT** - Устанавливает начальные числа первых семи генераторов случайных чисел.

**SHOW** - Вычисляет выражение и отображает результат.

**START**- Устанавливает счетчик завершения и запускает процесс моделирования.

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

**SТОР** - Устанавливает условие остановки, основанное на количестве попыток входа в блок.

**STORAGE** - Определяет память.

**ТАВLЕ** - Определяет таблицу.

**VARIABLE** - Определяет переменную.

# ЛИТЕРАТУРА

1. Шрайбер Т.Дж. Моделирование на GPSS,М., Машиностроение, 1980г. 592 с.

2.Система моделирования GPSS/ PC для ПЭВМ, Выпуск 69/В.А.Орлов и др.- М: Международный центр научной и технической информации, МЭИ, 1990 г.

3.GPSS WORLD REFERENCE MANUAL / htpp://www. Minutemansoftware. Com.

4. GPSS WORLD TUTORIAL MANUAL/ htpp: //www. Minutemansoftware. Com.

5.Руководство пользователя по GPSS World/ Перевод с английского/.- Казань. Изд-во «Мастер Лайн», 2002.-384с.

6. Учебное пособие по GPSS World/ Перевод с английского/.- Казань. Изд-во «Мастер Лайн», 2002.-272с.

**СОДЕРЖАНИЕ**

[Цели и порядок выполнения практических занятий 2](#_Toc37340440)

[Тема 1. Разработка концептуальной модели 3](#_Toc37340441)

[Тема 2. Основные команды системы моделирования GPSS World. Построение GPSS модели. 4](#_Toc37340442)

[УРОК 1 6](#_Toc37340443)

[УРОК2 6](#_Toc37340444)

[Тема З. Освоение языка моделирования GPSS. Основные функциональные объекты GPSS 8](#_Toc37340445)

[УРОК 3 9](#_Toc37340446)

[Урок 4 10](#_Toc37340447)

[Урок 5 10](#_Toc37340448)

[Урок 6 11](#_Toc37340449)

[Тема 4. Направленный вычислительный эксперимент на имитационной модели. Обработка результатов эксперимента. Процедуре ANOVA. 14](#_Toc37340450)

[Урок 7 14](#_Toc37340451)

[ПРИЛОЖЕНИЕ 1. Базовые концепции языка GPSS 16](#_Toc37340452)

[ПРИЛОЖЕНИЕ 2. Экскурс по СМО 20](#_Toc37340453)

[ПРИЛОЖЕНИЕ 3. Функциональная структура GPSS 23](#_Toc37340454)

[ПРИЛОЖЕНИЕ 4. Системные числовые атрибуты 31](#_Toc37340455)

[ПРИЛОЖЕНИЕ 5. Блоки GPSS 35](#_Toc37340456)

[ПРИЛОЖЕНИЕ 6. Команды GPSS 37](#_Toc37340457)

[ЛИТЕРАТУРА 38](#_Toc37340458)