

Билет 1

1. Философские аспекты (понятие эксперимента)

Методическая основа моделирования - это диалектический метод познания.

Научно техническое развитие в любой области обычно идёт по следующему пути:

- 1) Наблюдение и эксперимент
- 2) Теоретическое исследование
- 3) Организация производственного процесса

В научных исследованиях большую роль играет понятие **«гипотеза»**, т.е. определенные предсказания, основывающиеся на небольшом количестве опытных данных, наблюдениях или догадок.

Быстрая и полная проверка гипотезы может быть проведена в ходе специально поставленного эксперимента. Мы должны связать гипотезу и эксперимент.

Аналогией называется суждение о каком-либо частном сходстве.

Гипотеза и аналогия, отражающей реальный мир, должны обладать наглядностью или сводиться к удобным для исследования логическим схемам. Такие логические схемы, упрощающие рассуждения и логические построения, или позволяющие проводить эксперимент, уточняющий природу явления, называются моделями.

Модель - объект, заместитель объекта оригинала, обеспечивающий изучение некоторых свойств оригинала. Замещение одного объекта другим с целью получения информации о важнейших свойствах оригинала с помощью объекта модели называется **моделированием**.

2. Моделирование СМО с поступлением заявок по пуассоновскому закону и с обслуживающим прибором по экспоненциальному закону на языке GPSS

```

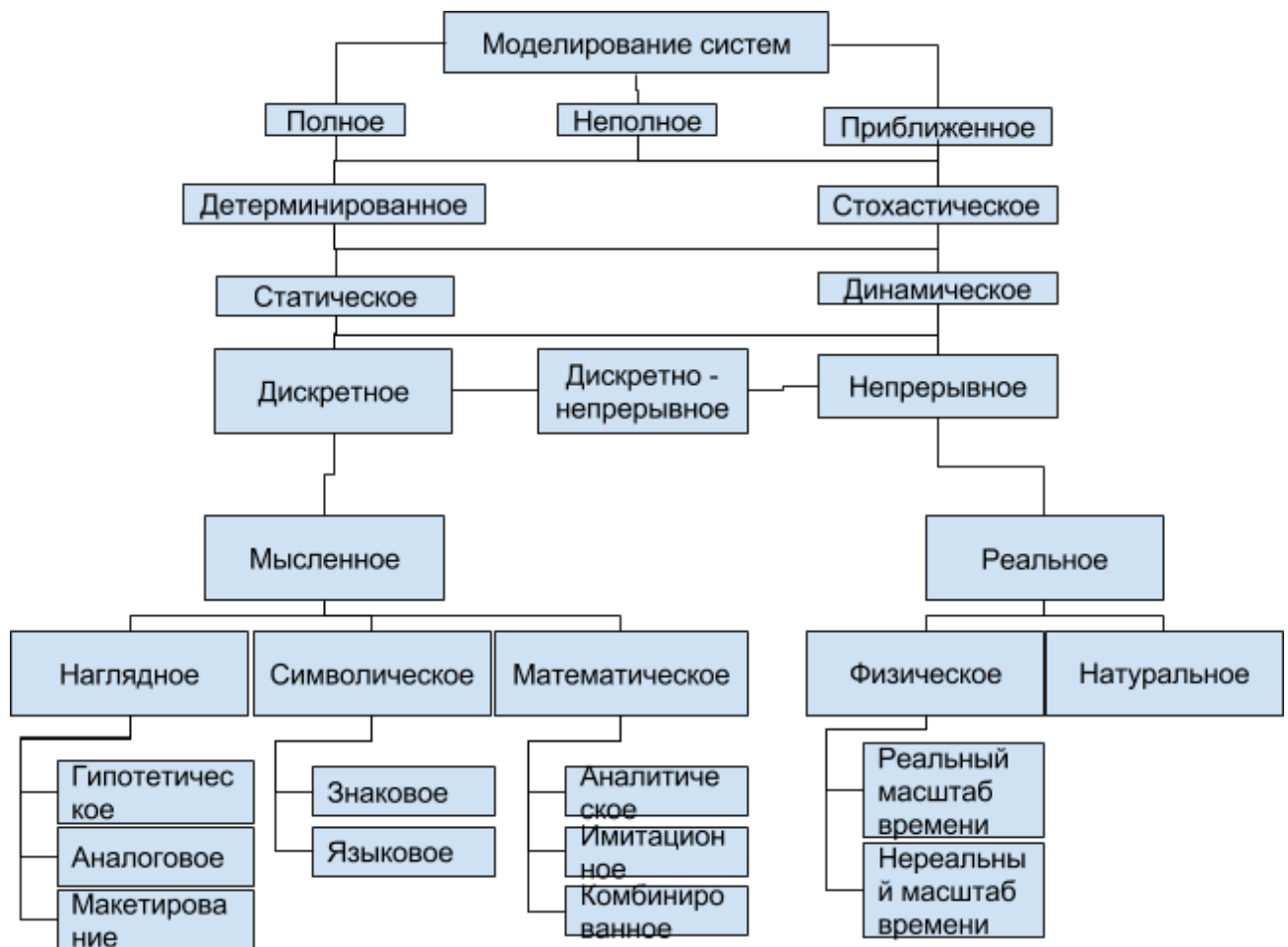
      GENERATE (Uniform(1,5,10)) , , , 1000      ; генерируется 1000 заявок равномерным распределением [5, 10]
QUE     QUEUE      STAND
;-----Захват обработчика и заявки из очереди
      SEIZE      PROCUNIT
      DEPART      STAND
      ADVANCE      (Exponential(1,0,0.1)) , ,      ; обработка по экспоненциальному распределению, бета = 1/lmb = 1/5 = 0.2
      RELEASE      PROCUNIT
      TRANSFER      .2 , , QUE      ; с вероятностью .2 заявка вернется к метке QUE

      TERMINATE 1

RESET
START 1000
|
```

Билет 2

1. Виды моделирования



По глубине моделирования методы делятся на две группы:

- Материальное - воспроизведение геометрических, физических и других свойств объекта
- Идеальное

В зависимости от типа носителя и сигнатуры:

1. Детерминированное - отображает процессы, предполагает отсутствие случайных воздействий
2. Стохастическое - тоже самое, но с/в учитываются
3. Статическое
4. Динамическое - исследование объекта во времени

2. Управляющие блоки

START A,B,C,D * A – начальное значение счётчика завершения. * B – может быть написано NP, признак подавления формирования стандартного отчёта. * C – не используется. * D – включение в отчёт списков текущих и будущих событий, если 1.

SIMULATE A * A – ограничение реального времени в минутах.

RMULT – устанавливает значения ГСЧ.

RESET – сбрасывает всю статистическую информацию, накопленную в процессе прогона модели. При этом состояние аппаратных, динамических и запоминающих объектов, а также ГСЧ сохраняется.

CLEAR – модель приводится к состоянию до первого прогона (сбрасывается всё)

END – завершает сеанс работы с GPSS и возвращает управление в ОС.

Билет 3

1. Технические средства математического моделирования

Принципы фон Неймана

1. Принцип однородности памяти: принципиальное отличие архитектуры "фон Неймана" (принстонской) от "Гарвардской". Команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы. Распознать их можно только по способу использования; то есть одно и то же значение в ячейке памяти может использоваться и как данные, и как команда, и как адрес в зависимости лишь от способа обращения к нему
2. Принцип адресности: структурно память состоит из пронумерованных ячеек, каждая из которых доступна процессору в любое время
3. Принцип программного управления: все вычисления, предусмотренные алгоритмом задачи, должны быть представлены в виде набора последовательно выполняемых команд, множество которых определяют возможности системы
4. Принцип двоичного кодирования: вся информация кодируется двоичными цифрами 0 и 1. Каждый тип информации имеет свой формат данных.

Виды вычислительных машин

Различают два вида: аналоговые и цифровые. Цифровая точна, но дискретна, а аналоговая быстра, но имеет узкий спектр применения. Аналоговые машины в качестве модели работают с величинами электрических цепей, которые идут в соответствии с переменными заданной модели. Преимущественно АВМ применяются для исследования объектов, динамику которых можно описать

дифференциальными уравнениями или ДУ в частных производных. Под АВМ стоит понимать совокупность электрических элементов, позволяющих провести изоморфное моделирование изучаемого объекта. По степени способности решения ИДУ различают малые, средние, большие АВМ.

Как особенный вид выделяются гибридные ВМ, позволяющие в разной степени применять обе формы представления и обработки информации. К таким машинам относятся:

- АВМ с цифровыми методами численного анализа, программируемые с помощью ЦВМ
- АВМ с цифровым управлением и логикой
- АВМ с цифровыми элементами
- ЦВМ с аналоговым арифметическим устройством
- ЦВМ с дифференциальным анализатором

Технические средства -- компьютеры -- используются в двух своих аспектах:

1. как средства расчета по полученным аналитическим моделям (ЦВМ, АВМ)
2. как средства имитационного моделирования

Гибридная вычислительная машина – вычислительная система, использующая как аналоговую, так и дискретную форму представления сигнала.

Гибридная ВМ:

Примеры гибридных ВМ:



- АВМ с цифровыми элементами (вольтметры, запоминающие устройства, но не решающие элементы)
- ЦВМ с аналоговыми арифметическими устройствами

2. Построение выражений в языке GPSS

GPSS World поддерживает широкое использование выражений. Они могут использоваться в PLUS-процедурах или в операторах GPSS (если заключены в скобки). Это означает, что высокий уровень мощности вычислений может быть достигнут уже в операндах блоков и команд. Выражения могут выполнять простые вычисления, вызывая процедуры, которые выполняют операции математического характера или операции над строками, выборку возможных распределений или осуществляют реализацию пользовательских алгоритмов, включая файлы ввода-вывода.

Билет 4

1. Основные понятия в теории моделирования (и определение имитационной модели)

Система — множество элементов, находящихся в отношениях и связях между собой. Часть системы, представление о которой не целесообразно подвергать дальнейшему членению при моделировании — **элемент**.

Сложная система — система, характеризующаяся большим числом элементов и большим числом взаимосвязей. Именно взаимосвязями определяется сложность системы.

Свойства: членимость, многоаспектность, целенаправленность, иерархичность.

Подсистема — часть системы (подмножество элементов и их взаимосвязи), которая имеет свойства системы.

Надсистема — система, по отношению к которой рассматриваемая система является подсистемой.

Структура — это отображение совокупности элементов системы и их взаимосвязей. Понятие структуры отличается от понятия системы тем, что при описании структуры принимают во внимание лишь типы элементов и связей без конкретизации значений их параметров.

Параметр — величина, выражающая свойство системы или ее части, или влияющая на систему через внешнюю среду.

При **имитационном** моделировании реализующий модель алгоритм воспроизводит процесс функционирования системы во времени, причем имитируются элементарные явления, составляющие процесс с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состоянии процесса в определенные моменты времени, дающие возможность оценить характеристики системы.

2. Графические возможности gpss (графики СЧА и гистограммы по таблицам)

Статистические таблицы используются для получения частотных распределений определенных аргументов, которыми могут быть некоторые СЧА (например, времени задержки транзакта в модели в целом или в отдельных ее частях; длин очередей; содержимого памяти и т. д.).

M TABLE A,B,C,D

A- табулир величина

B-верхняя граница левого интервала

C-ширина интервала

D-число интервалов

информация о гистограмме накапливается посредством попадания транзакта в блок:

TABULATE A

где A - имя гистограммы.

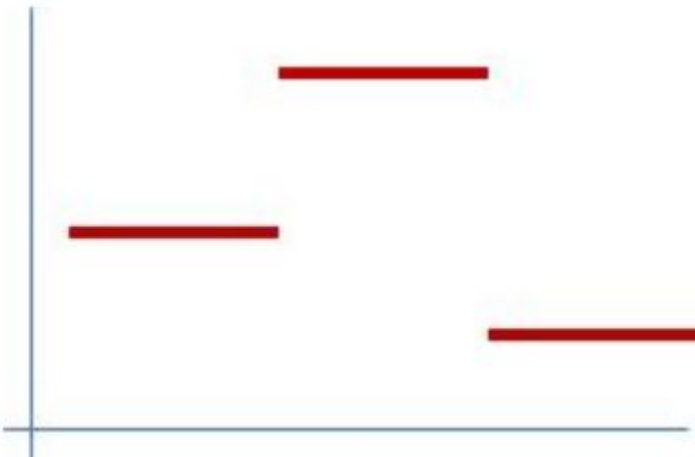
Функция – стандартный числовой атрибут, название и численная зависимость, обозначаемые в виде FN\$name. Описывается пользователем в виде численной зависимости от другого СЧА. Два основных типа – дискретные и непрерывные.

Пример: хранилище S с занятой памятью SMemory. Дискретная функция:

Func1 FUNCTION S\$Mem, D3 ;дискретная по трем точкам, ещё C, M, L, ..., S –

стандартный числовой атрибут

5,12 / 9,20 / 12,6 ; дробная часть различается точкой



до 5 будет идти на уровне 12, до 9 на 20, до 12 на 6 (кусочная). Функция неявно продлевается влево и вправо если пользователь введёт соответствующий аргумент.

На граничных участках берётся «всё ещё» значение (в 5 – 12).

В случае непрерывной – не набор горизонтальных отрезков, а ломаная.

4.5, 12.1 / 5.5, 10.5 / 7, 14 / 9, 15.3



Если задается много точек, то новая точка начинается со следующей строки, причем знак слэша в конце не ставится. Во всех незаданных участках (влево и вправо) функция продлевается горизонтально.

Для обращения к функции: FN\$Func1

Билет 5

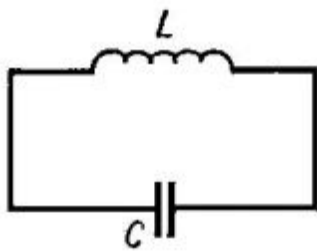
1. D-схема

Процесс, описываемый этой схемой, имеет непрерывно-детерминированный подход и ТМС основана на дифференциальных уравнениях.

В качестве примера возьмем колебательный контур и маятник:

колебательный контур с емкостью C_k и индукцией L_k , или маятник массой M и длиной l .

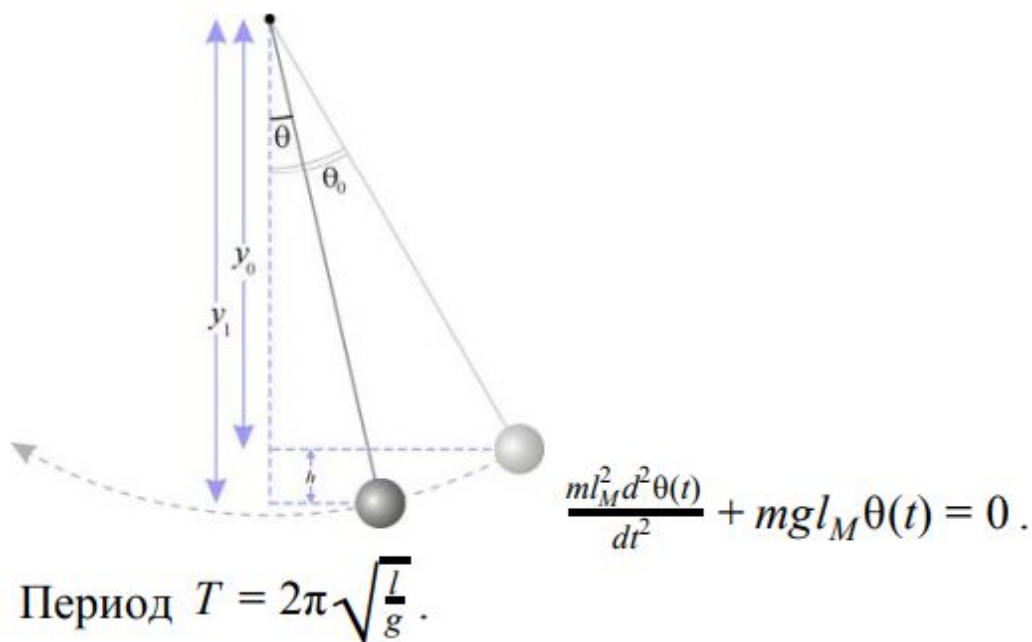
В контуре:



$$\frac{L_k d^2 q(t)}{dt^2} + \frac{q(t)}{C_k} = 0.$$

$q(t)$ - заряд на конденсаторе. Период $T_k = 2\pi \sqrt{L_k C_k}$

В маятнике:



Суть объектов разная, но процессы описываются обыкновенными ДУ. Введем понятия $h_2 = L_k = ml_M$, $h_1 = 0$, $h_0 = 1/C_k = mgl$. Если теперь положить состояние $z(t)=q(t)=\theta(t)$, то можно записать общий вид уравнения, описывающего функционирование данных объектов:

$$\frac{h_2 d^2 z(t)}{dt^2} + \frac{h_1 dz(t)}{dt} + h_0 z(t) = 0$$

Ноль символизирует собой отсутствие внешних воздействий X (включающих в себя заодно и V).

Использование D-схем позволяет формализовать процесс функционирования непрерывно-детерминированных систем и оценить их характеристики, применяя аналитический (иногда имитационный) подход, реализованный в виде соответствующего языка моделирования непрерывных систем с использованием аналоговых или гибридных средств вычислительной техники.

2. Интерактивные графические возможности в GPSS

Статистические таблицы используются для получения частотных распределений определенных аргументов, которыми могут быть некоторые СЧА (например, времени задержки транзакта в модели в целом или в отдельных ее частях; длин очередей; содержимого памяти и т. д.).

M TABLE A,B,C,D

A- табулир величина

B-верхняя граница левого интервала

C-ширина интервала

D-число интервалов

информация о гистограмме накапливается посредством попадания транзакта в блок:
TABULATE A

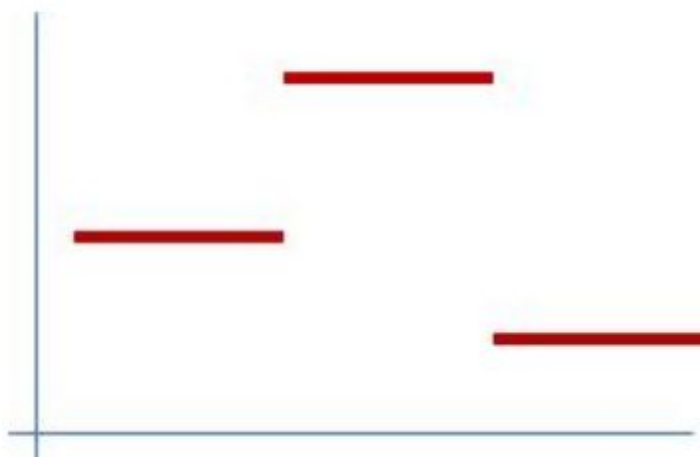
где A - имя гистограммы.

Функция – стандартный числовой атрибут, название и численная зависимость, обозначаемые в виде FN\$name. Описывается пользователем в виде численной зависимости от другого СЧА. Два основных типа – дискретные и непрерывные.

Пример: хранилище S с занятой памятью SMemory. Дискретная функция:

Func1 FUNCTION S\$Mem, D3 ;дискретная по трем точкам, ещё C, M, L, ..., S – стандартный числовой атрибут

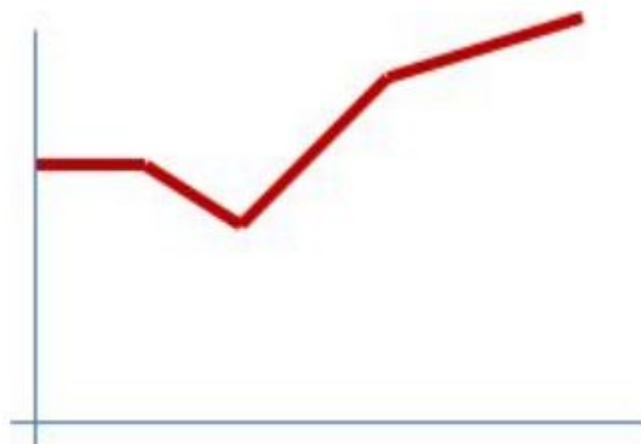
5,12 / 9,20 / 12,6 ; дробная часть различается точкой



до 5 будет идти на уровне 12, до 9 на 20, до 12 на 6 (кусочная). Функция неявно продлевается влево и вправо если пользователь введёт соответствующий аргумент. На граничных участках берётся «всё ещё» значение (в 5 – 12).

В случае непрерывной – не набор горизонтальных отрезков, а ломаная.

4.5, 12.1 / 5.5, 10.5 / 7, 14 / 9, 15.3



Если задается много точек, то новая точка начинается со следующей строки, причем знак слэша в конце не ставится. Во всех незаданных участках (влево и вправо) функция продлевается горизонтально.

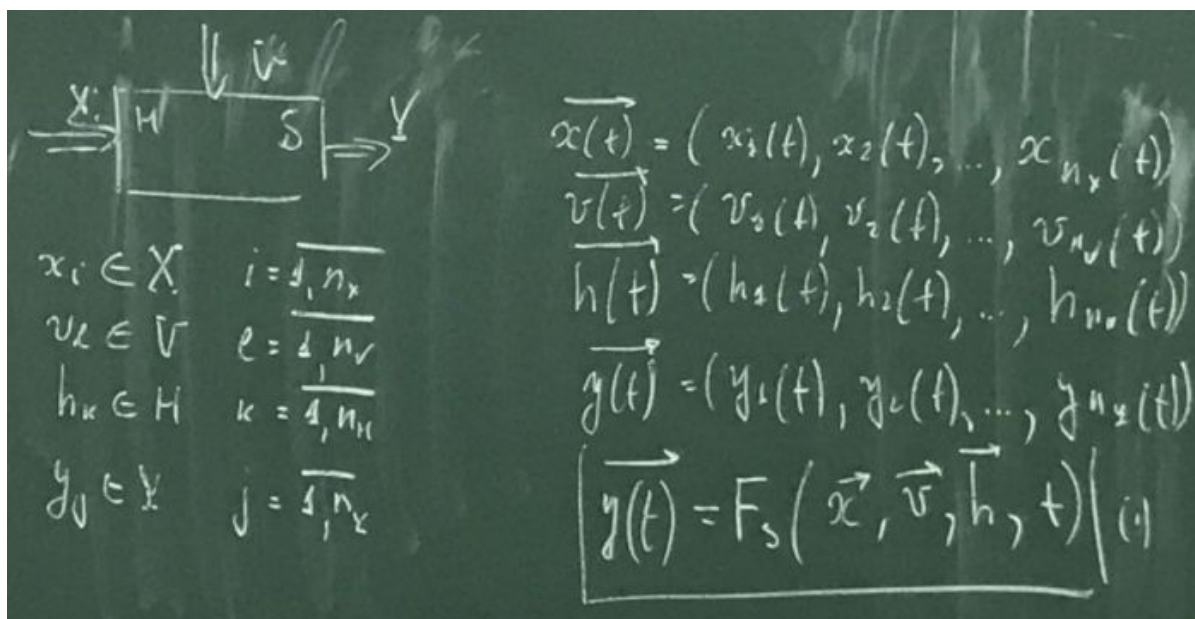
Для обращения к функции: FN\$Func1

Билет 6

1. Типовые математические схемы для сложных систем

Модель объекта моделирования можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих следующие подмножества:

2. совокупность входных воздействий x
3. совокупность воздействий внешней среды v
4. совокупность внутренних собственных параметров h
5. совокупность выходных характеристик системы y (под схемкой множества)



В общем случае эти множества не пересекаются и содержат параметры стохастического и детерминированного характера.

При моделировании функционирования системы x , v и h являются независимыми (экзогенными) переменными, которые в векторной форме могут быть записаны в следующей форме, а выходные характеристики являются эндогенными (зависимыми). Процесс функционирования описывается в итоге оператором F_s , который преобразует независимые переменные в зависимые. (справа сверху от схемы)

Выражение 1 описывает закон функционирования системы, в общем случае закон задается в виде функции, функционала, логических условий, в алгоритмическом или табличном виде.

Алгоритм -- это метод получения y с учетом x , v , h . Состояние -- свойства системы в конкретный момент времени, характеризуются некоторым вектором.

Если рассматривать процесс функционирования системы как последовательную смену состояний, то состояния могут быть интерпретированы как координаты точки в k -мерном фазовом пространстве. Причем, каждой реализации процесса будет соответствовать некоторая фазовая траектория.

Совокупность всех возможных состояний называется пространством состояний процесса моделирования.

В общем случае, время может быть непрерывным или дискретным (квантованным на интервале $[t_0, t]$).

Под математической моделью реальной системы будем понимать конечное множество переменных x , v , h вместе с математическими связями между ними и характеристиками $y(t)$.

Практика моделирования: на первоначальных этапах формализации объектов используются т.н. математические типовые схемы, к которым относятся:

Процесс функционирования	Типовая математическая схема	Обозначения
непрерывно-детерминированный подход	дифференциальные уравнения	D
дискретно-детерминированный	конечные автоматы	F
дискретно-стохастический процесс	вероятностные автоматы	P
непрерывно-стохастический	системы массового обслуживания	Q
универсальный подход	агрегативные системы	A

2. Методика отладки в GPSS

Для пошагового выполнения модели с целью ее отладки можно воспользоваться командой STEP (выполнить шаг). Операнд в поле А команды задает количество входов активного транзакта в блоки, которое производится при каждом выполнении команды. Обычно этот операнд равен 1, и каждое выполнение команды STEP приводит к продвижению активного транзакта к следующему блоку. Отладку с использованием команды STEP удобно проводить, находясь в окне блоков. Для продолжения моделирования после прерывания следует ввести в командную строку команду CONTINUE (продолжить). Подробнее ниже. GPSS имеет в своем составе развитые средства отладки ИМ, доступ к которым осуществляется из пункта главного меню «Window → Simulation Window» GPSS реализует пошаговую отладку модели с одновременным отображением процесса перемещения транзактов между блоками ИМ в окне «BLOCK ENTITIES». Для этого в главном меню необходимо выбрать пункт «Window → Simulation Window → Block Window» Для управления процессом моделирования в панели инструментов окна «BLOCK ENTITIES» предусмотрены кнопки «Continue», «Halt» и «Step»

Билет 7

1. Основные требования при разработке модели и аппаратной реализации

Суть машинного моделирования состоит в проведении на компьютере эксперимента с моделью, которая представляет собой, как правило, программный комплекс, описывающий формальным или алгоритмически поведение элементов системы в процессе ее функционирования.

Функционирование элементов — поведение в их взаимодействии друг с другом и внешней средой.

Основные требования к модели

1. Полнота модели. Должна предоставлять пользователю возможность получения необходимого набора характеристик , оценок системы с требуемой точностью и достоверностью.
2. Гибкость модели. Должна давать возможность воспроизведения различных ситуаций при варьировании структуры, алгоритмов и параметров модели. Структура должна быть блочной.
3. Компьютерная реализация модели должна соответствовать имеющимся тех средствам.

2. Визуальные средства отображения всякого о модели в процессе работы программы в gpss (или что-то похожее)

Для наблюдения за процессом моделирования и действием на него команд на этапе тестирования и верификации используются делить графических окон. Окна подразделяются по типам объектов:

- блоки
- выражения
- устройства
- логические ключи
- матрица
- график
- очереди
- ячейки
- памяти
- таблоицы

Окно блоки показывает выход транзактов в блоки. Оно позволяет с помощью мыши или клавиатуры устанавливать и удалять контрольные точки и визуально отслеживать передвижение транзактов.

Окно выражения предназначено для наблюдения за изменениями любого количества Plus-выражений.

С помощью окна график одновременно можно наблюдать любое количество многоцветных графиков.

Окно таблица представляет собой динамическую гистограмму, полезную для наблюдения за сбором данных, поиска выбросов и оценки сходимости к порождающему вероятностному распределению.

Окна блоки, устройства, логические ключ, очереди, ячейки, памяти имеют подробный обзор и общий обзор. Открываются они всегда с подробным образом.

Билет 8

1. Основные этапы моделирования сложных систем

1. Построение концептуальной модели и её формализация
2. Алгоритмизация и компьютерная реализация модели.
3. Получение и интерпретация результатов моделирования.

Первый этап -- формирование модели и построение формальной схемы.
Переход от содержательного описания к математической модели.

1. Проведение границы между системой и внешней средой
2. Выделение основных составляющих функционирования системы
3. Исключение из рассмотрения некоторых второстепенных элементов

4. Группировка блоков в группы: имитаторов событий внешних воздействий, модели процесса функционирования и вспомогательных блоков
5. Разбитие ПФ на подпроцессы до достижения элементарности построения модели

Второй этап -- этап алгоритмизации и компьютерной реализации.

1. Разработка схемы алгоритма
2. Разработка схемы программы
3. Выбор технических средств для реализации
4. Процесс программирования и отладки
5. Тестирование достоверности
6. Составление ТД

Третий этап -- этап проведения рабочих расчетов по готовой программе.

1. Планирование эксперимента с моделью
2. Проведение собственных расчетов (контрольная калибровка модели)
3. Статистическая обработка результатов расчетов
4. Интерпретация результатов моделирования
5. Составление ТД

2. Способ моделирования СМО в GPSS

```

GENERATE (Uniform(1,5,10)) , , , 1000 ; генерируется 1000 заявок равномерным распределением [5, 10]
QUE     QUEUE     STAND

;-----Захват обработчика и заявки из очереди
SEIZE   PROCUNIT
DEPART  STAND
ADVANCE (Exponential(1,0,0.1)) , , ; обработка по экспоненциальному распределению, бета = 1/lmb = 1/5 = 0.2
RELEASE PROCUNIT
TRANSFER .2 , , QUE ; с вероятностью .2 заявка вернется к метке QUE

TERMINATE 1

RESET
START 1000

```

Билет 10

1. Структурный уровень проектирования

Выделяют следующие уровни проектирования:

1. Структурный. В качестве типовых матсхем - системы массового обслуживания, А-схемы, сети Петри, вероятностные автоматы
2. Функционально-логический уровень проектирования

- а. подуровень регистровых передач. Элемент моделирования - регистры, сумматоры; всё разделяется на комбинационные и последовательные схемы.
 - б. логический уровень. Моделируем работы ЛЭ, когда они синтезированы на уровне одной кремниевой подложки. Используются цифровые автоматы, Ф-схемы
3. Конструкторский
 - а. ОДУ, ДУ в частных производных. Характерен тем, что решаются в основном задачи отвода тепла, вентиляции, оптимального размещения интегральных схем (оптимизация длины проводников) либо вопросы технологичности многослойной печатной платы.

Моделирование на системном уровне

При моделировании новой или модернизации действующей ВС и сетей необходимо предварительно оценить эффективность их функционирования с учетом различных вариантов структурной организации. Эти варианты могут отличаться составом и характеристиками модулей (моделей устройств), структурой межмодульных связей, режимами работы, алгоритмами управления и т.д. Именно в таких случаях используются модели ВС.

Вычислительное Средство - комплекс аппаратных и программных средств, которые в совокупности выполняют определенные обработочные функции.

Операционная Система – набор ручных и автоматических процедур, которые позволяют группе людей эффективно использовать вычислительную установку.

Коллектив пользователей – сообщество таких людей, которые используют ВС для удовлетворения своих нужд по обработке информации.

Входные сигналы (программы, данные, команды), которые создаются коллективом пользователей, называются **рабочей нагрузкой**.



Индекс производительности – описатель, который используется для представления производительности системы. Различают количественные и качественные индексы производительности.

Качественные:

- легкость использования системы
- мощность системы команд

Количественные:

- пропускная способность – объем информации, обрабатываемый в единицу времени.
- время ответа (реакции) – время между предъявлением системе входных данных и появлением соответствующей выходной информации.
- коэффициент использования оборудования – отношение времени использования указанной части оборудования в течение заданного интервала времени к длительности этого интервала.

Концептуальная модель ВС включает сведение о выходных и конструктивных параметрах системы, её структуре, особенностях работы каждого элемента и ресурса, постановка прикладных задач, определение цели моделирования.

Основные задачи:

2. Определение принципов организации ВС.
3. Выбор архитектуры, уточнение функций ВС и их разделение на подфункции, реализация аппаратным и программным путем.
4. Разработка структурной схемы – определение состава устройств и способов их взаимодействий.
5. Определение требований к выходным параметрам устройств и формирования технического задания на разработку устройств для функционально-логического уровня проектирования.

6. Table, tabulate

NAME TABLE A,B,C,D

- NAME - Entity Label for this entity. Required. The field must be Name. The length of a Table name is limited to 32 characters.
- A - Table argument. Optional. The data item whose frequency distribution is to be tabulated. The operand must be Name, Number, String, ParenthesizedExpression, or SNA. Ignored by ANOVA, but must be specified when used by TABULATE Blocks.
- B - Upper limit of first frequency class. The maximum argument which causes the first frequency class to be updated. Required. The operand must be Number or String.
- C - Size of frequency classes. The difference between the upper limit and lower limit of each frequency class. Required. The operand must be Number or String.
- D - Number of frequency classes. Required. The operand must be PosInteger.

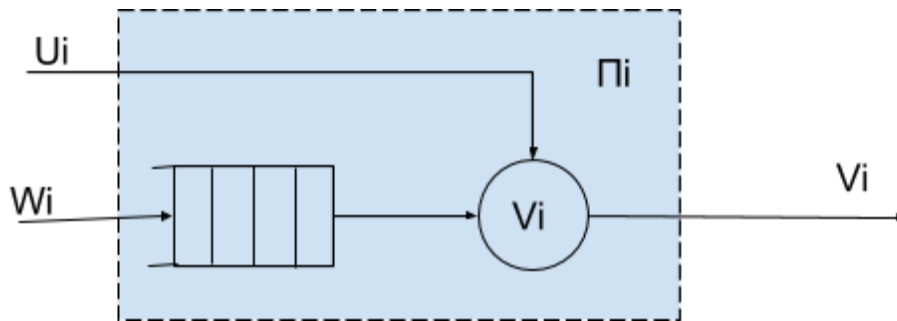
TABULATE Sales

- When a Transaction enters this TABULATE Block, the Table Entity named Sales is found. Sales must have been defined in a TABLE Command. Then the statistics associated with the table are updated with no weighting.

Билет 11

1. Q-схемы

Q-схема реализует непрерывно-стохастический подход, используется для моделирования систем массового обслуживания.



Самым главным понятием является понятие потока заявок.

Для СМО характерно случайное появление заявок на обслуживание и случайное же распределение времени на обработку заявки.

В любом акте обслуживания можно выделить две составляющие: ожидание обслуживания и процесс обслуживания.

Поток событий -- последовательность событий, происходящих одно за другим в какие-то случайные моменты времени.

Поток событий однороден, если он характеризуется только моментами поступлений этих событий и задается некоторой последовательностью $t: \{t_n\} = t_0 < t_1 < t_2 < \dots < t_n$. Поток неоднороден, если он задается еще и набором признаков события $f: \{t_n, f_n\}$.

Если интервалы между сообщениями случайны и независимы, то мы имеем поток с ограниченным последствием.

Поток событий ординарен, если вероятность того, что на момент времени t придется 2 и более события, пренебрежительно мала по сравнению с вероятностью иного исхода.

ПС стационарен, если вероятность появления того или иного числа событий на интервале времени зависит лишь от длины этого интервала, а не от того, где этот самый интервал был взят.

Для стационарного потока интенсивность потока равна среднему числу событий в единицу времени.

Для ординарного потока число сообщений на участке dt , примыкающему к моменту времени t , будет равно $P(>1)[t, dt] + P(1)[t, dt] \sim P(1)[t, dt]$. Соответственно, предел при $t \rightarrow 0$ от $P(1)/dt$ даёт нам $\lambda(t)$ -- интенсивность ординарного потока.

Процесс функционирования прибора можно представить как изменение состояний его элементов во времени. Таким образом, вектор состояний может

быть либо вектором накопителя, либо вектором канала: $_{Zi} = (Zi^N, Zi^K)$
Что касается канала — то он находится в двух состояниях — канал свободен и канал занят.

В практике моделирования элементарной Q-схемы объединяются, при этом, если каналы обслуживания различных приборов соединены параллельно, то имеет место многоканальное обслуживание. А если последовательно, то многофазное. Следовательно, для задания QS необходимо использовать оператор сопряжения R, отражающий взаимосвязь элементов структуры между собой. Различают разомкнутые и замкнутые Q-схемы.

Параметрами для Q-схемы будут:

- количество фаз
- количество каналов в каждой фазе
- количество накопителей в каждой фазе
- емкость накопителя (если она равна нулю, то система с потерями, иначе — система с ожиданием.)

Еще кое что:

- необходимо описать алгоритм её функционирования, который определяет набор правил поведения заявок в системе.
- неоднородность заявок, отражающая состояние системы в то или иное время учитывается с помощью введения классов приоритетов.

Весь набор алгоритмов поведения заявок в Q-схеме можно передать в виде оператора Q:

$Q = (U, W, H, Z, R, A)$

где

2. W - входные потоки
3. U - поток обслуживания
4. R - оператор сопряжения элементов
5. H - собственные параметры
6. Z - состояния системы
7. A - оператор алгоритмов поведения и обслуживания заявок.

8. Блоки SEIZE и RELEASE

Транзакты в GPSS перемещаются от блока к блоку. Если в какой-то момент транзакт занимает одноканальное устройство (ОКУ), то он должен войти в соответствующий блок, описывающий ОКУ.

Для SEIZE:

1. Если ОКУ уже используют, то транзакт встать в очередь и должен ждать, пока он освободится.
2. Если ОКУ не используется, то транзакт войдет в блок, а статус ОКУ изменится на «занят».

RELEASE соответственно переводит статус ОКУ в состояние «свободен». В качестве параметров обоим блокам передается имя ОКУ.

Билет 13

1. марковские случайные процессы. Многоканальная смо с отказами

Все уравнения Колмогорова соответствуют следующим правилам:

1. В левой части находится производная вероятности состояния по времени, а в правой части столько членов, сколько и стрелок, связанных с этим состоянием.
2. В правой части для исходящих стрелок используется знак минус, а для входящих -- плюс.
3. Каждый член равен произведению плотности вероятности перехода интенсивности, умноженную на вероятность того состояния, из которого выходит стрелка.

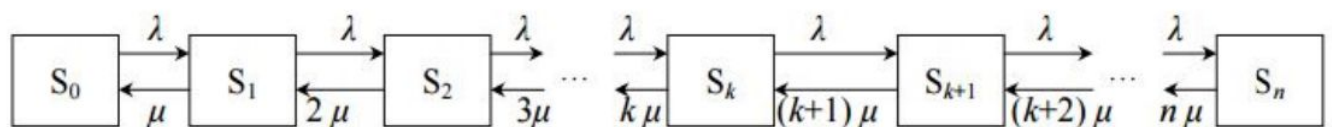
Состояние системы будем интегрировать по числу занятых каналов.

S_0 – все каналы свободны.

S_1 - занят один канал, остальные свободны.

S_k - занято k каналов, остальные свободны.

S_n - все S_n каналов заняты.



Проставим интенсивности:

1. стрелка слева направо переводит поток заявок с интенсивностью λ
2. стрелка справа налево -- с интенсивностью $n \cdot \mu$, где i -- номер текущего состояния.

По уравнениям Колмогорова получаем:

$$p'_0 = -\lambda * p_0 + \mu * p_1$$

$$p'_1 = -\lambda * p_1 - \mu * p_1 + \lambda * p_0 + 2 * \mu * p_2$$

$$p'_k = -\lambda * p_k - k * \mu * p_k + \lambda * p_{k-1} + (k + 1) * \mu * p_{k+1}$$

$$p'_n = -n * \mu * p_n + \lambda * p_{n-1}$$

Предельные вероятности состояний p_0, \dots, p_n характеризуют установившийся режим работы СМО. Следовательно, все левые части у уравнений равны 0.

- Предельные вероятности состояний p_0, \dots, p_n характеризуют установившийся режим работы СМО. Это значит, что левые части всех уравнений равны 0.

$$p_0 = \frac{1}{1 + \frac{(\frac{\lambda}{\mu})}{1!} + \dots + \frac{(\frac{\lambda}{\mu})^n}{n!}}$$

$$p_k = \frac{(\frac{\lambda}{\mu})^k}{k!} * p_0$$

$$\frac{\lambda}{\mu} = \rho$$

- Зная все вероятности p_0, \dots, p_n можно найти характеристики СМО:

- 1) Вероятность отказов: $p_{\text{отк}} = p_n = \frac{\rho^n}{n!} * p_0$

- 2) Вероятность принятия: $q = 1 - p_{\text{отк}}$

- 3) Среднее число обработанных заявок: $= \lambda * q = \lambda(1 - p_{\text{отк}})$

- 4) Производительность: $\lambda = \frac{1}{\text{время обработки}}$

- 5) Если время входа мало по сравнению с временем обработки

$$\text{время обработки} = \frac{1}{\mu}, \mu - \text{технические характеристики компьютера}$$

2. Блоки синхронизации транзактов

MATCH A -- одностная очередь, один приходит, а тот что перед ним был -- выходит.

ASSEMBLE A -- число A транзактов объединяется в ансамбль, и в результате после набора числа A транзактов выходит первый его транзакт, а остальные уничтожаются.

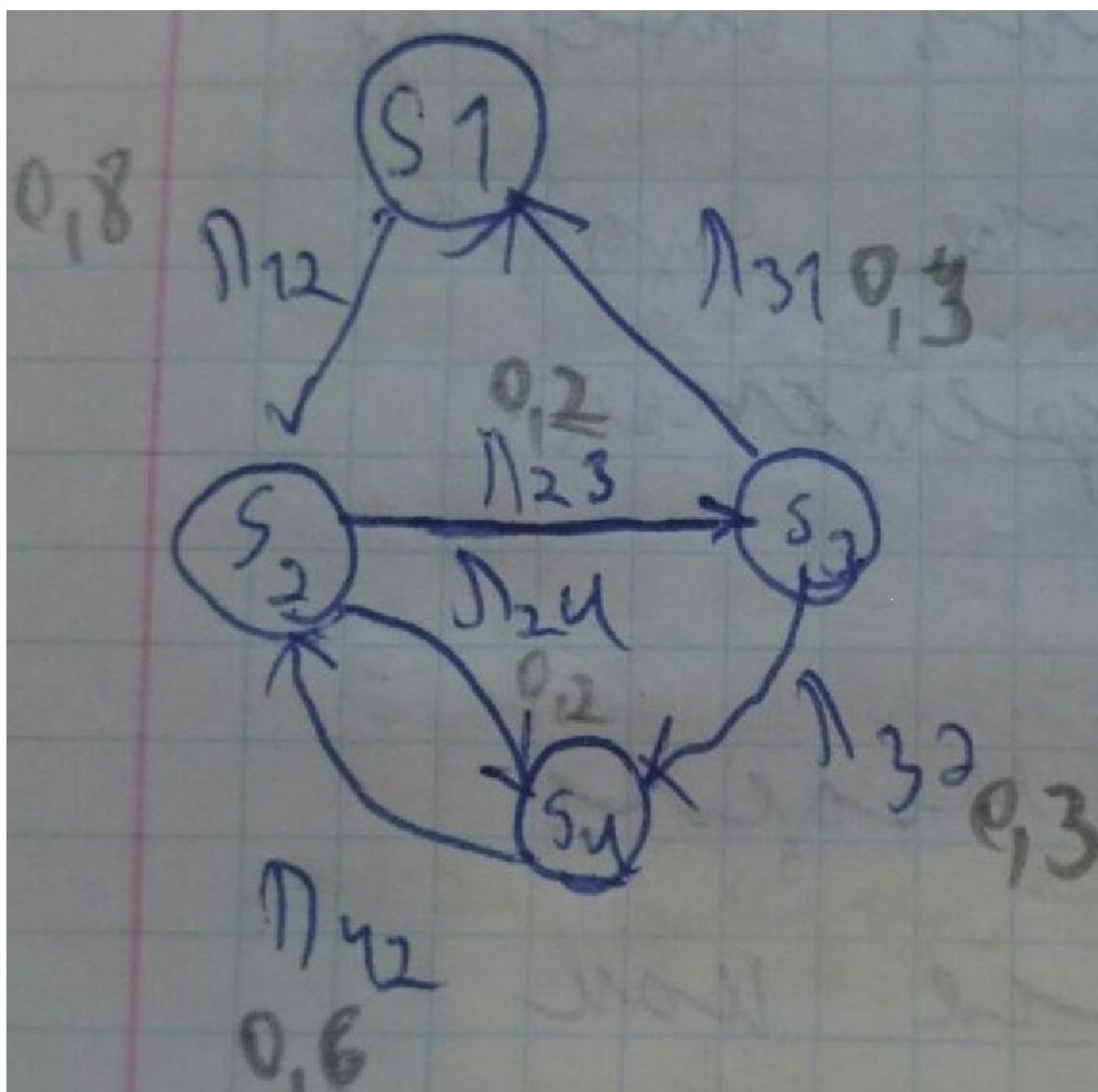
GATHER A -- аналогично, только без уничтожения транзактов

Билет 14

1. Марковские процессы. Уравнения Колмогорова.

Все уравнения Колмогорова соответствуют следующим правилам:

1. В левой части находится производная вероятности состояния по времени, а в правой части столько членов, сколько и стрелок, связанных с этим состоянием.
2. В правой части для исходящих стрелок используется знак минус, а для входящих -- плюс.
3. Каждый член равен произведению плотности вероятности перехода интенсивности, умноженную на вероятность того состояния, из которого выходит стрелка.



Пусть λ -- плотность вероятности. Найдём вероятность нахождения системы в состоянии $S1 = P1(t)$.

Придадим t малое приращение dt и найдем вероятность того, что в момент времени $t+dt$ система будет находиться в состоянии $S1$.

Вероятность того, что система не вышла из $S1$: $p1 \times (1 - \lambda_{12} \times dt)$ Вероятность того, что система перешла из $S3$ в $S1$: $p3(t) \times \lambda_{31} \times dt$

Тогда в итоге имеем: [вероятность первого] + [вероятность второго] = $p1(t+dt)$

- Первое состояние:

$$\lim_{\Delta t \rightarrow 0} \frac{p_1(t + \Delta t) - p_1(t)}{\Delta t} = -\lambda_{12} * p_1(t) + \lambda_{31} * p_3(t)$$

$$p_1'(t) = -\lambda_{12} * p_1(t) + \lambda_{31} * p_3(t)$$

- Второе состояние:

$$p_2(t + \Delta t) = p_2(t) * (1 - \lambda_{24} * \Delta t - \lambda_{23} * \Delta t) + p_1(t) * \lambda_{12} * \Delta t + p_4(t) * \lambda_{42} * \Delta t$$

$$\lim_{\Delta t \rightarrow 0} \frac{p_2(t + \Delta t) - p_2(t)}{\Delta t} = -\lambda_{23} * p_2(t) - \lambda_{24} * p_2(t) + \lambda_{12} * p_1(t) + \lambda_{42} * p_4(t)$$

$$p_2'(t) = -\lambda_{23} * p_2(t) - \lambda_{24} * p_2(t) + \lambda_{12} * p_1(t) + \lambda_{42} * p_4(t)$$

(Аналогично, для третьего и четвертого)

Интегрируя пределы, можно получить искомые вероятности как функции от времени. Начальные условия берутся в зависимости от начального состояния системы. Учитывается условие нормировки.

2. Блоки аппаратной категории GPSS: PREEMPT и RETURN.

PREEMPT A B C D E -- фиксирует использование устройства на более высоком уровне чем SEIZE, приостанавливает обслуживание транзакта, захватившего устройство ранее, дает возможность прерванному транзакту захватить устройство после того, как закончится обслуживание прервавшего транзакта. Если при выполнении PREEMPT выяснилось, что устройство обслуживает прерывание, то блок не выполняется и транзакт задерживается до тех пор, пока не освободится устройство.

RETURN A -- сбрасывает владение над устройством и идет дальше.

Билет 15

1. Способы получения последовательности случайных чисел

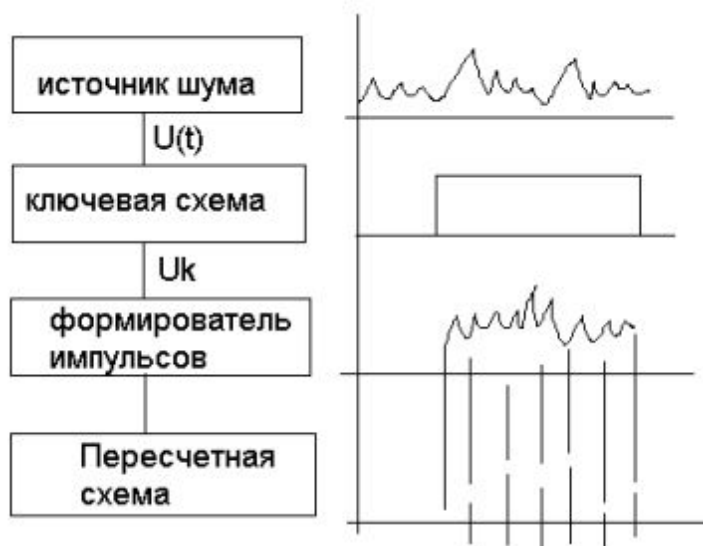
При имитационном моделировании системы одним из основных вопросов является стохастических воздействий. Для этого метода характерно большое число операций со случайными числами или величинами и зависимость результатов от качества последовательностей случайных чисел. На практике используется 3 основных способа:

1. Аппаратный (физический)
2. Табличный (файловый)
3. Алгоритмический (программный)

Аппаратный способ. Аппаратный представляет из себя шум.

Случайные числа вырабатываются случайной электронной приставкой (генератор случайных чисел), служащей, как правило, в качестве одного из внешних устройств. Реализация этого способа обычно не требует дополнительных вычислений, а необходима только операция обращения к ВУ. В качестве физического эффекта, лежащего в основе таких генераторов чаще всего используют шумы в электронных приборах.

Т.е. необходимо: источник шума, ключевая схема, формирователь импульсов, пересчетная (см. рис.)



Табличная схема. Случайные числа оформляются в виде таблицы и помещаются во внешнюю или оперативную память.

Алгоритмический способ. Способ основан на формировании случайных чисел с помощью специальных алгоритмов.

2. Блоки изменения параметров транзактов в GPSS

ASSIGN A,B,C

- номер параметра
- значение для добавления или замены
- номер модификатора-функции

ALTER -- изменяет приоритет или атрибуты сообщений данной группы.

MARK A - изменяет время рождения транзакта

Билет 17

1. Моделирование потоков событий

Самым главным понятием является понятие потока заявок.

Для СМО характерно случайное появление заявок на обслуживание и случайное же распределение времени на обработку заявки.

В любом акте обслуживания можно выделить две составляющие: ожидание обслуживания и процесс обслуживания.

Поток событий -- последовательность событий, происходящих одно за другим в какие-то случайные моменты времени.

Поток событий однороден, если он характеризуется только моментами поступлений этих событий и задается некоторой последовательностью $t: \{t_n\} = t_0 < t_1 < t_2 < \dots < t_n$. Поток неоднороден, если он задается еще и набором признаков события $f: \{f_n, f_n\}$.

Если интервалы между сообщениями случайны и независимы, то мы имеем поток с ограниченным последствием.

Поток событий ординарен, если вероятность того, что на момент времени t придется 2 и более события, пренебрежительно мала по сравнению с вероятностью иного исхода.

ПС стационарен, если вероятность появления того или иного числа событий на интервале времени зависит лишь от длины этого интервала, а не от того, где этот самый интервал был взят.

Для стационарного потока интенсивность потока равна среднему числу событий в единицу времени.

Для ординарного потока число сообщений на участке dt , примыкающему к моменту времени t , будет равно $P(>1)[t,dt] + P(1)[t,dt] \sim P(1)[t,dt]$. Соответственно, предел при $t \rightarrow 0$ от $P(1)/dt$ даёт нам $\lambda(t)$ -- интенсивность ординарного потока.

Процесс функционирования прибора можно представить как изменение состояний его элементов во времени. Таким образом, вектор состояний может быть либо вектором накопителя, либо вектором канала: $_{Zi} = (Zi^N, Zi^K)$ Что касается канала — то он находится в двух состояниях — канал свободен и канал занят.

2. Блоки SPLIT/ASSEMBLE

ASSEMBLE A -- число A транзактов объединяется в ансамбль, и в результате после набора числа A транзактов выходит первый его транзакт, а остальные уничтожаются.

SPLIT A,B,C,D -- создат A копий транзакта

- входящих в блок B
- записываемых в параметры C
- с числом параметров D

Билет 23

1. Событийный принцип протяжки модельного времени

Событийный принцип. Характерное свойство моделируемых систем обработки информации – состояние отдельных устройств изменяется в дискретные моменты времени, совпадающие с: моментами времени поступления сообщений в систему; временем окончания решения задач; временем возникновения аварийных сигналов и т.п. Моделирование и продвижение текущего времени в системе удобно проводить с использованием событийного метода. Состояние всех блоков имитационной модели анализируется лишь в момент появления какого-либо события.

Моменты наступления следующих событий определяются минимальным значением из списка будущих событий. Список представляет собой совокупность моментов ближайшего изменения состояния каждого из блоков системы.

Достоинство: не пропустим ни одного события

Недостаток: при большом количестве событий (сложная система) список необходимо просматривать постоянно (можно держать отсортированным).

2. Классификация блоков GPSS

Блоки, используются для описания функций и управляют движением транзактов. У каждого блока имеется 2 стандартных числовых атрибута(СЧА):

Wn–счетчик входа в блок. Как правило номер транзакта, входящего в данный блок.

Nn–Общий счетчик транзактов, поступивших в блок с начального момента моделирования.

Практически все изменения состояния модели возникают в результате поступления транзактов в соответствующие блоки. После выполнения блока транзакт либо движется к следующему блоку, либо задерживается, либо уничтожается. Блоки изменяют атрибуты транзакта.

1. Блоки осуществляющие модификацию атрибутов транзакта
 - Временная задержка ADVANCE
 - Генерация и уничтожение транзактов GENERATE TERMINATE
SPLIT ASSEMBLE
 - Синхронизация движения нескольких транзактов MATCH GATHER

- Изменение параметров транзакции ASSIGN INDEX MARK
 - Изменение приоритета PRIORITY
2. Блоки, изменяющие последовательность передвижения транзактов, т.е. блоки передачи управления.

TRANSFER, LOOP, TEST, GATE

1. Блоки, связанные с группирующей категорией.

JOIN, REMOVE, EXEMINE, SCAN, ALTER

1. Блоки, сохраняющие значения для дальнейшего использования.

SAVEVALUE, MSAVEVALUE

1. Блоки, организующие использование объектов аппаратного устройства.

SEIZE - RELEASE, PREEMPT - RETURN, F\S AVAIL – UNAVAIL, ENTER – LEAVE, LOGIC

1. Блоки, обеспечивающие получение статистических результатов. QUEUE, DEPART, TABULATE, TABLE
2. Специальные блоки

HELP, TRACE, UNTRACE, PRINT, ...?

1. Блоки для организации цепей

LINK, UNLINK

9.Вспомогательные блоки

LOAD, SAVE

Каждый блок определяется с помощью отдельной команды. В общем случае: сначала идет нумерация, затем обязательно поле метки, затем поле операции, поле операндов, затем, если необходимо, комментарий (через ;). Т.е.:

<Нумерация><Метка><Оператор><Операнды><Комментарии>

Билет 24

1. Классификация языков имитационного моделирования



Непрерывное моделирование сводится к представлению ДУ, с помощью которой устанавливается связь между входной и выходной функциями.

GASP -- комбинированный, на основе Fortran. Может описать события, зависящие от времени и от состояния.

Последняя группа используется в основном для имитационного моделирования, при этом используются разные способы описания динамического поведения системы.

2. Язык GPSS. Построение и основные принципы

General Purpose Simulation System - язык общецелевой системы моделирования.

Возможности:

- Многозадачность
- Использование вирт. памяти
- Интерактивность
- GUI
- Визуализация процесса моделирования

Для GPSS моделью сложной дискретной системы является описание её элементов и логических правил их взаимодействия в процессе функционирования. Можно выделить конечный набор абстрактных элементов "объектов", причем набор логических правил также ограничен и может быть описан небольшим числом операций.

В процессе моделирования объекты взаимодействуют друг с другом путем изменения атрибутов и преобразования их значений - в результате "событий".

Транзакты моделируют прохождение по системе соответствующих единиц потока. Такое движение может быть разбито на цепь элементарных событий, происходящих в определенные моменты времени.

Основная задача симулятора - выявление моментов наступления этих событий и выполнение определенных действий при их наступлении.

Билет 25

1. Алгоритмический способ получения последовательности псевдослучайных чисел

Алгоритмы:

1. Выделение значения дробной части многочлена первой степени ($Y_n = A \cdot n + B$)
2. фон Нейман. Каждое последующее число образуется возведением предыдущего в квадрат и отбрасыванием начальной и конечной цифр
3. Лимер $g_{n+1} = (K \cdot g_n + C) \bmod M$. Наибольший период: при $k = 3+8i$ или $5+8i$ (комплексные числа)
4. Метод Фибоначчи на основе одноименных чисел и натуральной арифметике

Достоинства

- Однократная проверка
- Многократное воспроизведение последовательности
- Места в RAM не занимает
- Не требуется внешнее устройство

Недостатки

1. Запас чисел ограничен периодом
2. Затраты машинного времени

2. Язык GPSS. Блоки. Объекты языка

Объекты языка подразделяются на 7 категорий и 14 типов:

Категория	Тип
Динамическая	Транзакт
Операционная	Блоки
Аппаратная	Устройства памяти, ключи, (?) одноканальные устройства
Вычислительная	Переменные, арифметические\логические функции, (?) генераторы случайных чисел
Статистическая	Очереди, таблицы
Запоминающие	Ячейки, матрицы ячеек
Группирующие	Списки, группы

Динамические объекты - транзакты (сообщения), которые представляют собой единицы исследуемых потоков и производят ряд определенных действий, продвигаясь

по фиксированной структуре, представляющей собой совокупность объектов других категорий.

Операционные объекты задают логику функционирования системы и определяют маршрут движения транзактов между объектами аппаратной категории.

Аппаратные объекты – абстрактные элементы, на которых может быть декомпозировано оборудование реальной системы. Воздействуя на эти объекты, транзакты могут изменять их состояния и влиять на движение других транзактов.

Вычислительная категория – служит для описания таких ситуаций в процессе моделирования, когда связи между компонентами моделируемой системы наиболее просто и компактно выражаются в виде математических соотношений.

Статистический объект - очередь\таблица, служащая для оценки поведения системы.

Объекты запоминающей категории обеспечивают обращения к сохраняемым значениям. Ячейки сохраняемых величин и матрицы ячеек сохраняемых величин используются для сохранения некоторой числовой информации. Любой активный транзакт может произвести запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считать любой транзакт.

Группирующая категории состоит из трёх типов объектов: числовых групп, групп транзактов и списков.

Билет 26

1. Типы СМО для моделирования сложных систем

Признаки классификации:

1. закон распределения входного потока заявок
2. числа обслуживающих приборов
3. закон распределения времени обслуживания в обслуживающих приборах
4. число мест в очереди
5. дисциплина обслуживания

Для обозначения СМО принята система кодирования $A|B|C|D|E$, где

- А–закон распределения интервалов времени между поступлениями заявок.

Наиболее часто используемое обозначение:

- М–экспоненциальное
- Е–эрлангово
- Н–гипер-экспоненциальное
- (?)–гамма
- D–детерминированное
- G–произвольное
- В–закон распределения времени обслуживания в приборах. Приняты те же обозначения, как и для интервалов между появлениями заявок.

- C – число обслуживающих приборов. Для одноканальной – 1, для многоканальной – L .
- D – число мест в очереди.
 n или r – конечно
 $<\text{опущено}>$ – неограниченно
- E – дисциплина обслуживания. Наиболее часто используются следующие варианты дисциплины обслуживания: FIFO (может опускаться), LIFO, RANDOM

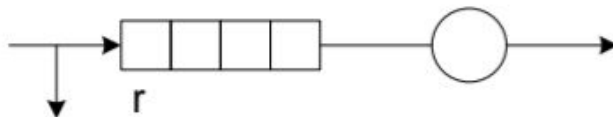
Пример: $M/M/1$ – СМО с одним ОА, бесконечной очередью, экспоненциальными законами распределения времени между поступлениями заявок и времени обслуживания, дисциплина обслуживания FIFO.

$G/G/1$ - Одноканальные системы с ожиданием:

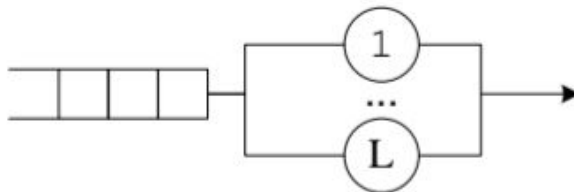
$G/G/1$ - Одноканальные системы с ожиданием:



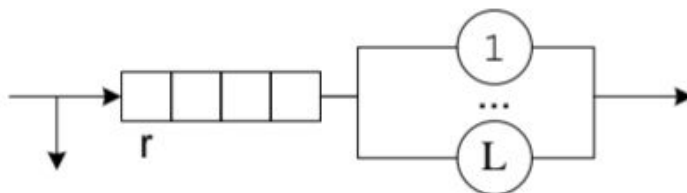
$G/G/1/r$ - Одноканальная система с потерями:



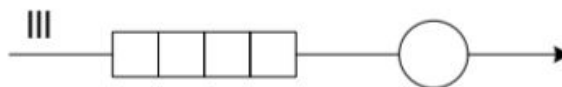
$G/G/L$ - Многоканальная система с ожиданием



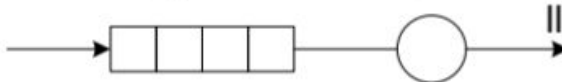
$G/G/1/r$ – Многоканальная система с потерями



Gr/G/1 – одноканальная система с групповым поступлением заявок:



G/Gr/1 – одноканальная система с групповым обслуживанием



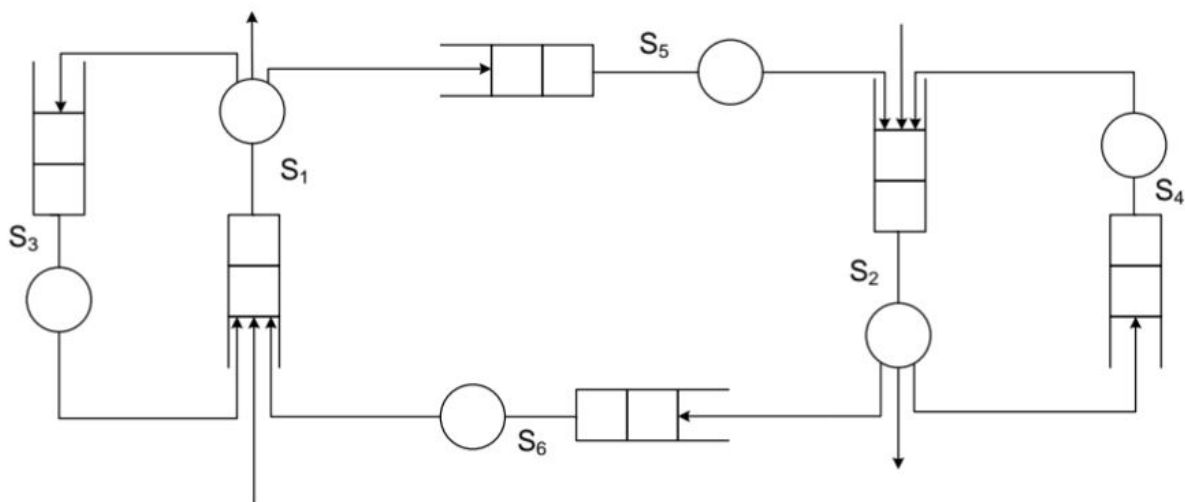
Для моделирования вычислительных систем и сетей наиболее часто используются следующие типы СМО:

1. Одноканальное СМО с ожиданием. Представляет собой один обслуживающий прибор с бесконечной очередью. Является наиболее распространенной при исследовании СДС. Формализует функционирование практически любого числа узла вычислительной сети.
2. Одноканальная СМО с потерями. Один обслуживающий прибор с конечным числом мест в очереди. Используется при моделировании каналов передачи в вычислительных сетях.
3. Многоканальные СМО с ожиданием. Представляют собой несколько параллельно работающих обслуживающих приборов с общей параллельной очередью. Используется при моделировании групп абонентских терминалов, работающих в диалоговом режиме.
4. Многоканальные СМО с потерями. Наиболее часто используются при моделировании работы каналов.
5. Одноканальные СМО с групповым поступлением заявок. Также как и одноканальная СМО с групповым обслуживанием заявки используются для моделирования центров коммутации.

Вычислительные сети в целом могут быть исследованы с помощью сетей массового обслуживания.

Различают сети:

- 1) **Открытые.** Сеть массового обслуживания, состоящая из m узлов, причем хотя бы в один из узлов сети поступает извне входящий поток заявок и обязательно имеется сток заявок из сети. Для открытых сетей характерно то, что интенсивность поступления заявок в сеть не зависит от состояния сети, т.е. от числа уже поступивших. Такие сети используются как правило, для исследования функционирования вычислительной сети, работающей в неоперативном режиме



S1, S2 – моделируют работу узлов коммутации

S3, S4 – моделируют работу серверов

S5, S6 – моделируют работу межузловых каналов

В сети циркулируют два потока заявок. Каждая заявка поступает на вход соответствующего узла коммутации, где определяется место её обработки. Затем заявка передается на «свой» сервер или по каналу связи на соседний сервер, где заявки обрабатываются. После чего возвращается к источнику и покидает сеть.

2) **Замкнутые** – называются сети МО с множеством узлов без источника и стока, в которой циркулирует постоянное число заявок.

Замкнутые сети МО используются для моделирования таких вычислительных систем, источниками информации для которых служат абонентские терминалы, работающие в диалоговом режиме. В этом случае каждая группа абонентских терминалов представляется в виде многоканальной системы МО с ожиданием и включается в состав устройств сети.

Различают простой и сложный режим диалога.

- При простом: абоненты не производят никаких действий кроме посылки заданий в вычислительную сеть и обдумывания полученного ответа.

Схема (самостоятельно): группы абонентов, каналы связи с абонентами, узлы коммутации, серверы и каналы межузловой связи.

Абоненты с терминалов посылают запросы, которые по каналам связи поступают на узлы коммутации. А оттуда на обработку на свой или соседний сервер.

- При сложном режиме диалога работа абонентов представляется в виде совокупности операций некоего процесса, называемого **технологическим**. Каждая операция технологического процесса, моделируется соответствующей системой массового обслуживания. Часть операций предусматривает обращение к вычислительной системе, а часть может и не обращаться.

3) **Смешанные** – называются сети МО в которой циркулирует несколько различных типов заявок (трафик). Причем относительно одних типов заявок сеть замкнута, а относительно других открыта. С помощью смешанных сетей МО моделируют такие вычислительные сети часть которых работает в диалоговом режиме, а часть в неоперативном. Причем для диалоговых

абонентов также различают простой и сложной режим работы. Так же смешанными сетями МО моделируются вычислительные системы в которых сервер дополнительно загружается задачами, решаемыми на фоне работы сети.

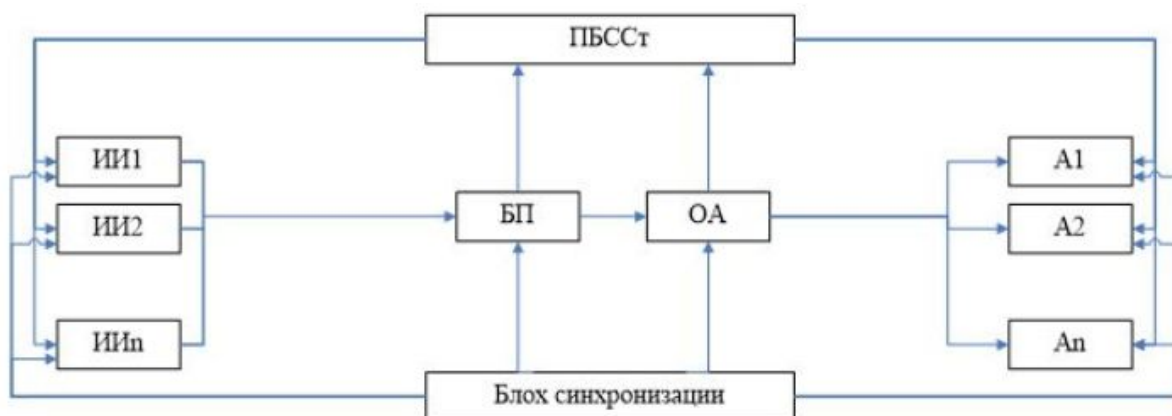
2. Отладка в GPSS

Для пошагового выполнения модели с целью ее отладки можно воспользоваться командой STEP (выполнить шаг). Операнд в поле A команды задает количество входов активного транзакта в блоки, которое производится при каждом выполнении команды. Обычно этот операнд равен 1, и каждое выполнение команды STEP приводит к продвижению активного транзакта к следующему блоку. Отладку с использованием команды STEP удобно проводить, находясь в окне блоков. Для продолжения моделирования после прерывания следует ввести в командную строку команду CONTINUE (продолжить). GPSS имеет в своем составе развитые средства отладки ИМ, доступ к которым осуществляется из пункта главного меню «Window → Simulation Window» GPSS реализует пошаговую отладку модели с одновременным отображением процесса перемещения транзактов между блоками ИМ в окне «BLOCK ENTITIES». Для этого в главном меню необходимо выбрать пункт «Window → Simulation Window → Block Window» Для управления процессом моделирования в панели инструментов окна «BLOCK ENTITIES» предусмотрены кнопки «Continue», «Halt» и «Step».

Билет X1

1. Методика построения программной модели

Для разработки программной модели исходная система должна быть представлена как **стохастическая СМО**, так как информация от внешней среды поступает в случайные моменты времени, и в общем случае длительность обработки различных типов информации может быть различна. То есть, внешняя среда является **генератором сообщений**, а комплекс вычислительных устройств -- ОУ.



ИИ -- источники информации. БП -- буферная память, FIFO/LIFO-очередь сообщений. Вследствие различности времени обработки и появления сообщения возможно сложение данных, ожидающих обработки. А -- абоненты. ПБССт -- программный блок

сбора статистики. Блок синхронизации -- блок, отвечающий за управлением активацией тех или иных фрагментов системы.

2. Язык GPSS. Блоки создания и уничтожения транзактов

GENERATE A,B,C,D,E

- время между поступлением транзактов
- модификатор интервала а
- начальная задержка
- число порождаемых транзактов
- приоритет транзакта

TERMINATE A удаляет транзакты из системы

Билет X2

1. Проверка адекватности и корректности модели

Проверка адекватности модели некоторой системы заключается в анализе её соизмеримости, а также равнозначности системы.

Адекватность часто нарушается из-за идеализации внешних условий и режимов функционирования, пренебрежением случайными факторами. Простейшей мерой адекватности является отклонение некой характеристики оригинала от модели.

Считают, что модель адекватна с системой, если вероятность того, что отклонение не превышает определенного значения, выше допустимой нормы.

Однако на деле нельзя использовать такой подход, так как отсутствует информация по выходным характеристикам объекта. Нужно проводить экспертный анализ разумности результатов моделирования.

Если по результатам проверки выделяется недопустимое несогласование модели и системы, возникает необходимость во внесении изменений.

Типы изменений

1. Глобальные -- методические ошибки в концепте или матем. модели.
2. Локальные -- уточнение алгоритмов
3. Параметрические (калибровочные)

2. Язык GPSS. QUEUE, RELEASE.

QUEUE A,B

Этот блок осуществляет сбор статистики об очереди. Номер очереди в которую заносится транзакт задается в поле А. При записи нового транзакта в очередь определяется длина интервала времени, в течение которого длина очереди оставалась неизменной.

При входе транзакта в данный блок текущая длина очереди увеличивается на число единиц, указанное в поле В. Затем происходит сравнение с максимальной длиной очереди, достигнутой до этого момента времени. Если оно больше старого значения, то оно его заменяет. Кроме того, счетчик общего числа единиц прошедших через очередь увеличивается на тоже число единиц.

RELEASE A

В GPSS элементами, которые требуют обслуживания, являются транзакты. Они перемещаются в модели от блока к блоку. Если в какой-то момент активности транзакт занимает ОКУ, то для этого он входит (или пытается войти) в соответствующий блок, описывающий это ОКУ. Блок должен обладать следующими свойствами:

- если ОКУ уже используют, транзакт не может войти в блок и должен ждать в очереди;
- если ОКУ не используют, транзакт может войти в блок, статус ОКУ изменяется на "занято".

Блок, обладающий этими свойствами, является блоком SEIZE (занято). Вход транзакта

в блок SEIZE моделирует занятие ОКУ. После обслуживания вход того же транзакта в другой блок моделирует освобождение ОКУ. Назначением этого блока является изменение состояния ранее занятого ОКУ с "занято" в "незанято". Этим блоком является блок RELEASE (освободить).

ДОП.ВОПРОС: ADVANCE с распределением Пуассона и $\lambda = 3$

ADVANCE A,B

Блок задает среднее время выполнения операций в моделируемой системе, а так же разброс времени относительно среднего. Задержка – целое число. Для задания времени пребывания в блоке ADVANCE пользователь указывает среднее время в поле А, а модификатор в поле В. Если поле задержки постоянно, то поле В может быть пустым. А если нулевое, то и поле А может отсутствовать. Модификаторы могут быть двух типов:

1. Модификатор «интервал», используется, когда время задержки транзакта распределено равномерно в некотором заданном интервале. Например: ADVANCE 5,2 (т.е. интервал от 3 до 7)
2. Модификатор «функция», когда интервал отличается от равномерного и приходится с помощью этого блока находить данное время. Указываем среднюю величину, а дальше функцию, на значение которой должна быть умножена данная величина.

Билет Х3

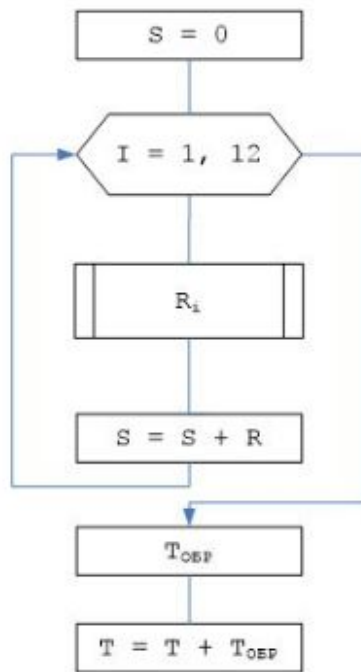
1. Моделирование работы обслуживающего аппарата

Программа-имитатор ОА - это комплекс, вырабатывающий случайные отрезки времени, соответствующие длительностям обслуживания требований. Например, если

требования от источника обрабатываются в ОА по нормальному закону с параметрами M и σ , то длительность обработки i -требования будет:

$$T_{обр} = M_x + \left(\sum_{i=1}^{12} R_i - 6 \right) \cdot \sigma_x$$

Схема алгоритма имитатора.



где

1. R_i - случайное число с равномерным законом распределения
2. $T_{обр}$ - время обработки очередного сообщения
3. T - время освобождения ОА

2. Язык GPSS. Объекты запоминающей категории, ячейки.

Объекты запоминающей категории обеспечивают обращения к сохраняемым значениям. Ячейки сохраняемых величин и матрицы ячеек сохраняемых величин используются для сохранения некоторой числовой информации. Любой активный транзакт может произвести запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считать любой транзакт. Матрицы могут иметь до шести измерений.

Ячейки используются для записи и хранения в процессе моделирования текущих значений СЧА.

Занесение информации в ячейку производится блоком SAVEVALUE, имеющим формат

SAVEVALUE A,B,C

где A – имя ячейки (может сопровождаться в конце знаком плюс $+$ или минус $-$), B – присваиваемое значение, C – тип ячейки. Если после A стоит знак $+$ или $-$, то значение поля B прибавляется или вычитается из текущего содержимого ячейки A . Если знак не указан, то значение поля B присваивается ячейке A . Поле C определяет

тип ячейки и может принимать значения: ХН – полусловная, ХF – полнословная, ХL – с плавающей точкой. При отсутствии поля С по умолчанию принимается полнословная ячейка.

Начальное значение ячейки по умолчанию равно нулю. Для изменения начального значения применяется оператор инициализации INITIAL Ячейка1, Значение1,..., Ячейка К, ЗначениеК

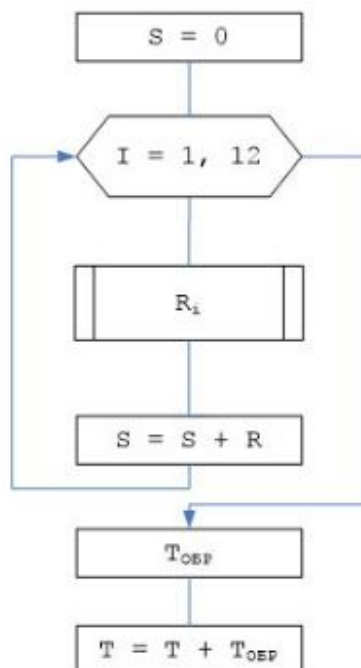
Билет Х4

1. Моделирование работы абонентов

Абонент - частный случай ОА, с той лишь разницей, что он может как обрабатывать заявки, так и вырабатывать их. Для моделирования работы абонентов необходимо вырабатывать длительности обслуживания требований. А генерация заявок может имитироваться с помощью генератора сообщений по наперед заданному закону.

$$T_{обр} = M_x + \left(\sum_{i=1}^{12} R_i - 6 \right) \cdot \sigma_x$$

Схема алгоритма имитатора.



где

1. R_i - случайное число с равномерным законом распределения
2. $T_{обр}$ - время обработки очередного сообщения
3. T - время освобождения ОА

2. Язык GPSS. Ячейки, матрицы ячеек

Ячейки используются для записи и хранения в процессе моделирования текущих значений СЧА.

Занесение информации в ячейку производится блоком SAVEVALUE, имеющим формат

SAVEVALUE A,B,C

где А – имя ячейки (может сопровождаться в конце знаком плюс + или минус -), В – присваиваемое значение, С – тип ячейки. Если после А стоит знак + или -, то значение поля В прибавляется или вычитается из текущего содержимого ячейки А. Если знак не указан, то значение поля В присваивается ячейке А. Поле С определяет тип ячейки и может принимать значения: ХН – полусловная, ХF – полнословная, ХL – с плавающей точкой. При отсутствии поля С по умолчанию принимается полнословная ячейка.

Начальное значение ячейки по умолчанию равно нулю. Для изменения начального значения применяется оператор инициализации INITIAL Ячейка1, Значение1,..., Ячейка К, ЗначениеК

Наряду с ячейками в моделях можно использовать матрицы ячеек. В отличие от простых ячеек матрицы перед использованием должны быть описаны. Для описания матрицы применяется строка описания матрицы. В поле метки этой строки записывается имя описываемой матрицы, в поле операции - слово MATRIX, в поле операндов - параметры матрицы: в поле А записывают любое слово или оставляют поле пустым, в поле В указывают число строк матрицы, в поле С - число столбцов.

NAME MATRIX A,B,C

В начальный момент любая матрица содержит только нулевые значения. Обращение MX\$NAME(A,B).

Запись значения в ячейку матрицы:

MSAVEVALUE NAME,B,C,v\$ABC

Может пригодиться

delta-t принцип протяжки времени

Принцип dt – заключается в последовательном анализе состояний всех блоков в момент $t+dt$ по заданному состоянию блоков в момент t . При этом новое состояние определяется в соответствии с их алгоритмическим описанием с учётом действующих случайных факторов, задаваемыми распределениями вероятностей. В результате этого анализа принимается решение о том, какие общесистемные события должны имитироваться программной моделью на данный момент времени. Основной недостаток принципа: значительные затраты вычислительных ресурсов на реализацию модели, а при недостаточно малом dt появляется опасность пропуска отдельных событий в системе, что приводит к искажению результатов моделирования. Характерное свойство системы обработки информации – то, что состояние отдельных устройств изменяется в дискретные моменты времени, совпадающие с моментами вступления сообщений в систему.

Метод вложенных цепей Маркова.

Вложенные цепи Маркова образуются следующим образом. В исходном случайном процессе выбираются такие случайные процессы, в которых характеристики образуют Марковскую цепь. Моменты времени обычно являются случайными и зависят от свойств исходного процесса. Затем обычными методами теории Марковских цепей исследуются процессы только в эти характерные моменты. Случайный процесс называется **полумарковским** (с конечным или счетным множеством состояний) если

заданы переходы состояний из одного состояния в другое и распределение времени пребывания процессов в каждом состоянии. Например, в виде функции распределения или функции плотности распределения.

Метод статистических испытаний. Метод Монте-Карло.

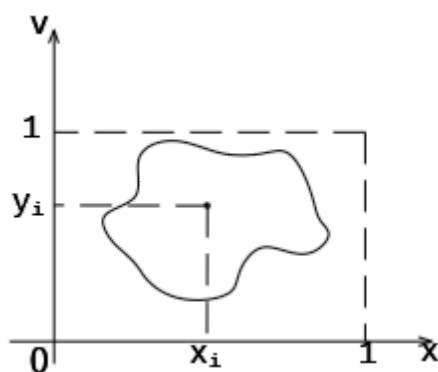
В СМО поток заявок редко бывает Пуассоновским и еще реже наблюдается распределенный закон.

Для произвольных потоков событий переводящих систему из состояния в состояние. Аналитические решения получены только для отдельных частных случаев. Когда построение аналитической модели является по той или иной причине трудно осуществимым ставится метод статистических испытаний.

**Когда этот метод нашел реальное применение: с развитием компьютеров.*

Идея метода: вместо того чтобы описывать случайные явления с помощью аналитической зависимости производится т.н. «розыгрыш», т.е. моделирование «случайного» явления с помощью некоторой процедуры дающей «случайный» результат. Проведя такой розыгрыш достаточно большое количество раз, получаем **статистический материал**, т.е. множество реализаций случайного явления. Дальше эти результаты могут быть обработаны статическими методами математической статистики.

Метод Монте-Карло был предложен в 1948 году Фон-Нейманом как метод численного решения некоторых математических задач.



Введем в некотором единичном квадрате случайную величину. Задача стоит в определении её площади.

Суть метода:

1. Вводим в некотором единичном квадрате любую поверхность S .
2. Любым способом получаем 2 числа x_i, y_i , подчиняющиеся равномерному закону распределения случайной величины на интервале $[0, 1]$.
3. Полагаем, что одно число определяет координату x , второе – координату y .
4. Анализируем принадлежность точки (x, y) фигуре. Если принадлежит, то увеличиваем значение счетчика на 1.
5. Повторяем n раз процедуру генерации 2х случайных чисел с заданным законом распределения и проверку принадлежности точки поверхности S .
6. Определяем площадь фигуры как количество попавших точек, к количеству сгенерированных.

Фон-Нейман доказал, что погрешность $\varepsilon \leq \sqrt{\frac{1}{n}}$.

Преимущество метода статистических испытаний: в его универсальности, которая обуславливает его возможность статистического исследования объекта, причем

всестороннего. Но для реализации этого исследования необходимы довольно полные статистические сведения о параметрах элемента.

Недостаток:

Большой объем требующихся вычислений, равный количеству обращений к модели. Поэтому вопрос выбора величины n имеет важнейшее значение. Уменьшая n – повышаем экономичность расчетов, но одновременно ухудшаем их точность.

