

Билет 1.

Философские аспекты (понятие эксперимента)

Все то, на что направлена человеческая деятельность называется **объектом** и вся методология моделирования направлена на получение упорядоченной и обработку информации об объектах. Научно-техническое развитие в любой области идет по следующим образом:

1. Наблюдение и эксперимент
2. Теоретическое исследование
3. Организация производственного процесса

В научных исследованиях большую роль играет **гипотеза** – определенные предсказания, основывающиеся на небольшом количестве опытных данных, наблюдениях, догадках. Быстрая и полная проверка выдвигаемых гипотез может быть проведена в ходе эксперимента. При формировании и правильности проверки гипотезы большое значение в качестве метода суждения имеет аналогия.

Аналогией называется суждение о каком-либо частном сходстве двух объектов. Современная научная гипотеза создается как правило по аналогии с проверенными на практике положениями. Аналогия связывает гипотезу и эксперимент.

Гипотезы и аналогии, отражающие реальный, объективно существующий мир, должны обладать наглядностью или сводиться к удобным для исследования лог. схемам. Такие лог. схемы, упрощающие рассуждения и логические построения или позволяющие проводить эксперименты, уточняющие природу явлений, называются **моделями**.

Модель – это объект, заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала. **Моделирование** – процесс замещения одного объекта другим (объекта оригинала объектом модели). Преимущества моделирования: гораздо дешевле и можно реализовать фантастические условия.

Подобие важный момент при создании модели. Модель идентичная оригиналу – это очень дорого и нерационально. Мат. модели могут не учитывать некоторые физ. воздействия.

Компьютерный эксперимент – процесс использования модели с

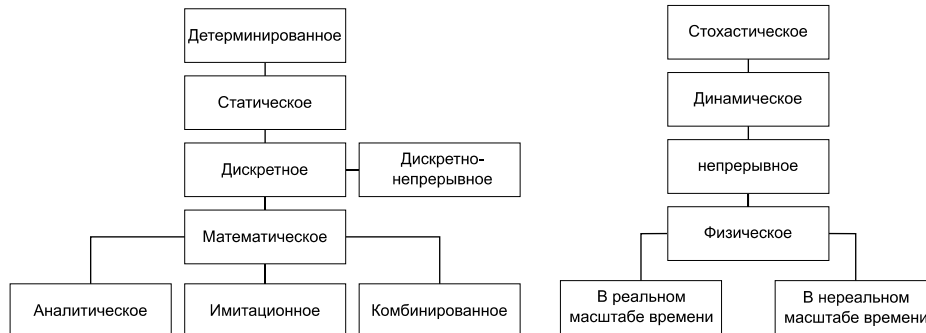
целью получения и анализа интересующей исследователя информации о свойствах моделируемой системы.

Моделирование СМО ЛР4 на языке GPSS.

```
Generate (10 # Poisson(1, 5)) ; генерация поступления заявок  
; первый параметр -- номер датчика генератора случ величин; # – умножение  
Compute TEST LE Q$QueueR,10,Overflow ; отказ в обслуживании  
QUEUE QueueR ; очередь  
SEIZE Comp1 ; занять ОА  
DEPART QueueR ; покинуть очередь  
ADVANCE (10 # Exponential(1, 0, 5)) ; задержка  
RELEASE Comp1 ; освободить ОА  
TRANSFER 0.1,Finish,Compute ; с вероятностью 0.1 повторить заявку  
Overflow SAVEVALUE AMOUNT_LOST,N$Overflow  
Finish Terminate 1  
Start 300
```

Билет 2

Виды моделирования



Детерминированное – моделируем процессы без случайных факторов.

Стохастическое отображает вероятностные процессы и события.

Статическое – описание объекта в какой-либо момент времени.

Динамическое – поведение объекта во времени.

Математическое моделирование – процесс установления реальному объекту некоторого мат. объекта, называемой мат. моделью, и исследовании этого объекта.

Вид математической модели зависит от *природы* рассматриваемого объекта и от *цели моделирования*. Математическая модель описывает реальный объект с некоторой степенью приближения.

Математическое моделирование можно разделить на:

1. **Аналитическое** (функциональные соотношения => большая точность).
Мб исследована 3-я способами:
 - a. Аналитическим – в общ. виде зависимость от характеристик
 - b. Численными методами – решение для опред начал условий
 - c. Качественно – указываем качеств. характер. (н-р, устойчивость)
2. **Имитационное** (нельзя аналит. => имитируем с больш. допущениями):
 - a. Агентное (задаём поведение частей системы => функц. в целом);
 - b. Дискретно-событийное (послед событий => функц системы);
 - c. Системная динамика (граф. причин связей объектов).
3. **Комбинированное** (часть элементов аналитически описываем, часть – имитируем => повышаем адекватность и точность модели)

При **имитационном моделировании** (ИМ) алгоритм, реализующий модель, воспроизводит процесс функционирования системы во времени, причем имитируются элементы системы с сохранением их логических структур и последовательностей протекания во времени. Результаты ИМ – реализации случ. величин => для нахождения характеристик процесса требуется его многократное воспроизведение с последующей статистической обработкой.

Комбинированное моделирование объединяет достоинства аналитического и имитационного. Декомпозиция процесса функционирования системы на подпроцессы. Где возможно – аналит, где нет – имитируем.

Физическое моделирование делиться на моделирование в реальном масштабе времени, т.е. натуральный эксперимент и в нереальном масштабе – в модельном времени.

Управляющие блоки GPSS

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

Технические средства математического моделирования

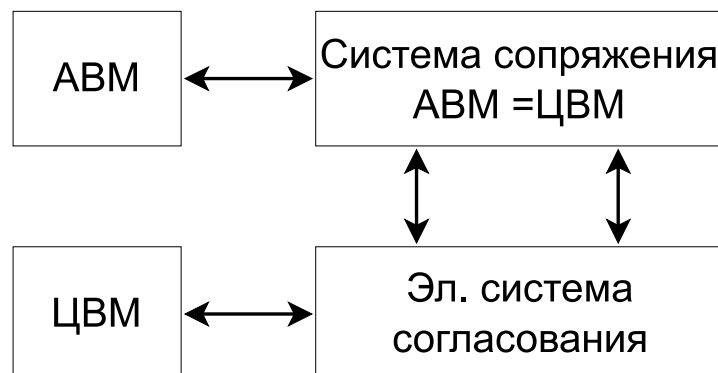
Классы используемых устройств:

- 1) Цифровая выч. техника является дискретной. Недостатки: слишком сложен механизм и ограниченное быстродействие. Процессор – устр-во, предназнач. для выполнения арифм. операций и управления над ними.
- 2) Аналоговая в.м. Скорость аналоговых ограничивается скоростью передвижения электрона в цепи. Но низкая точность из-за помех. Аналоговая техника относится к классу спец. машин и выполняет узкий класс вычислений. АВМ - совокупность электрических элементов, организованных в систему, позволяющих изоморфно моделировать динамику исследования объекта.

Структурная схема АВМ



- 3) Гибридная вычислительная машина. Вычислительная система, использующая как аналоговую, так и дискретную форму, представления сигнала.



Построение выражений в языке GPSS

Выражения могут применяться в переменных и операторах GPSS. При применении в переменных выражения определяются командами GPSS. При применении в операторах GPSS выражения определяются как часть языка PLUS.

GPSS имеет встроенный алгоритмический язык PLUS (Programming Language Under Simulation). Важной областью применения языка PLUS является разработка экспериментов – специальных пользовательских процедур, реализующих специфическую логику управления моделированием

Операторы (блоки) GPSS имеют следующий формат:

<метка> <имя_оператора> <поле_операндов> [<комментарий>]

Каждому объекту соответствуют атрибуты, описывающие его состояние в данный момент времени. Они доступны для использования в течение всего процесса моделирования и называются *системными числовыми атрибутами* (СЧА).

Категории	Типы объектов
Динамическая	Транзакты
Операционная	Блоки
Аппаратная	Одноканальные устройства, памяти (многоканальные устройства), логические ключи (переключатели)
Вычислительная	Переменные, функции, генераторы случайных чисел
Статистическая	Очереди, таблицы
Запоминающая	Ячейки, матрицы ячеек
Группирующая	Числовые группы, группы транзактов, списки

Билет 4

Основные понятия в теории моделирования и определение имитационной модели

Имитационное моделирование предполагает представление модели в виде некоторого алгоритма – выполнение которого имитирует элементы явления, состава процесса с сохранением их логических структур и последовательностей протекания во времени и т. о. представляет собой поведение моделируемой системы.

Система – множество элементов, находящихся в отношениях и связях между собой.

Элемент – часть системы, представления о которой нецелесообразно подвергать дальнейшему членению.

Сложная система – система, характеризующаяся большим числом элементов, и, что наиболее важно, *большим числом взаимосвязей* элементов. Сложная система определяется также видом взаимосвязей элементов, свойствами целенаправленности, целостности, членимости, иерархичности, многоаспектности.

Подсистема – это часть системы (подмножество элементов и их взаимосвязи), которая имеет свойства всей системы.

Надсистема – это система, по отношению к которой рассматриваемая система является подсистемой.

Структура – отображение совокупности элементов системы и их взаимосвязи. 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 свойства или системы, или её части, или влияющие на систему среды.

Основные свойства системы:

1. **Целенаправленность** – свойство искусственной системы, выражающее назначение системы для оценки наилучших вариантов системы.
2. **Целостность** – свойство системы, характеризующее взаимосвязанность элементов и наличие зависимостей выходных параметров от параметров элементов и лин. независимость вых параметров.

3. **Иерархичность** – свойство сложной системы, выражающее возможность и целесообразность её иерархического описания. Между компонентами имеется отношение “целое-часть”.

Моделирование сложных систем имеет 2 чётко различимые задачи:

1. Modeling – создание моделей сложных систем.
2. Simulation – анализ свойств системы на основе исследования их моделей.

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.).

NameTable TABLE A,B,C,D

NameTable – имя таблицы (обязательный параметр)

A – табулируемая величина (что заносим в гистограмму)

B – верхняя граница левого интервала (в 1 столбец попадут $x \in [B)$)

C – ширина интервалов

D – число интервалов.

Инфа накапливается посредством попадания транзакта в блок
TABULATE NameTable

Для очередей NameQTable QTABLE A,B,C,D (аналогично TABLE) и не требуется TABULATE. Этот оператор собирает инфу о том, сколько времени провели транзакты в очереди A. Пример:

```
Generate 10,2; генерация поступления заявок 10 +/- 2
LenTable TABLE Q$QueueR,0,1,12 ; таблица с инфой по длине очереди
TimeTable QTABLE QueueR,0,3,18 ; таблица с инфой о времени в очереди
QUEUE QueueR ; очередь
SEIZE Comp1 ; занять ОА
TABULATE LenTable ; занести инфу о длине очереди в таблицу LenTable
DEPART QueueR ; покинуть очередь
ADVANCE 10,4; задержка 10 +/- 4
RELEASE Comp1 ; освободить ОА
Terminate 1
```

Чтобы увидеть, что получилось запускаем симуляцию, после Window->Simulation Window->Table Window и указываем имя таблицы. Есть ещё графики. Спец операторы не требуются. Создали симуляцию. Window-

>Simulation Window->Plot Window, указали какие СЧА будем рисовать, н-р, текущую длину очереди Q\$QueueR, после чего запускаем симуляцию.

Билет 5

D-схемы

D-схема отображают динамику системы используя диффуры. Подходит для непрерывно-детерминированных процессов. Независимая переменная – обычно время.

В качестве примера возьмем диффур вынужденных колебаний:

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

Если рассматриваемая система не взаимодействует с внешней средой, то $x(t)=0$. Если нет потерь энергии, то $h_1=0$. Данный диффур применим как для колеб контура, так и для мат. и физ. маятника.

Использование D-схем позволяет формализовать процесс функционирования непрерывной детерминированной системы и оценить основные характеристики, применяя аналитический или имитационный подход, реализованный на соответствующем языке моделирования.

Интерактивные графические возможности в GPSS

Список окон window->simulation window системы GPSS с кратким описанием:

- Blocks – интерактив. представ. динамики перемещ. транзактов по блокам.
- Expressions – окно для интерактив. представления значений выражений
- Facilities – интерактив. представ. динамики значений параметров КО
- Logicswitches – представление динамики логических переключателей
- Matrix – для представления динамики значений элементов матрицы
- Plots – для интерактив. представления до 8 графиков выражений
- Queues – для представления динамики изменения очереди
- Savevalues – представление динамики изменения значений сохраняемых величин в процессе моделирования;
- Storages – представ. динамики изменен. значений параметров накопителя
- Table – для представления динамики изменения значений таблицы
- Model (Модель) – окно полноэкранного текстового редактора модели;
- JOURNAL (Журнал) – окно журнала для записи различных сообщений;

Список окон окна Command

- Create Simulation (Создать выполняемую модель)
- Repeat Last Command - Повторить последнюю команду
- START (Пуск) - запуск оттранслированной программы на выполнение
- STEP1 (Шаг 1) - пошаговое выполнение оттранслированной программы;
- HALT - прерывает процесс моделирования;
- CONTINUE - продолжение процесса моделирования;
- CLEAR – возвращение моделирования в первоначальное состояние;
- RESET – сброс статистики в начальное состояние;
- SHOW ... - просмотр искомых параметров команды в окне JOURNAL;
- Custom ... для ввода команд управления во время моделирования.

Билет 6

Типовые математические схемы для сложных систем

В практике моделирования на первоначальных этапах формализации объекта используют так называемые типовые математические схемы, к которым относятся хорошо проработанные и проверенные мат. объекты.

Процесс функционирования системы	Типовая математическая схема	Обозначение
Непрерывно-детерминированный подход	дифференциальные уравнения	D-схема
Дискретно-детерминированный подход	конечные автоматы	F-схема
Дискретно-стохастический подход	вероятностные автоматы	P-схема
Непрерывно-стохастический подход	СМО	Q-схема
Обобщенные (универсальный)	агрегативная система	A-схема

Сложность возрастает сверху вниз. В А-схемах используется иерархический подход. Основными операциями являются композиция и декомпозиция.

Если неизвестны функции многих аргументов – в частных производных, если одна – ОДУ. Называется D-схема т.к. схемы такого вида отображают динамику изучения схемы.

Методика отладки 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 “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

Основные требования при разработке модели и аппаратной реализации

Сущность компьютерного моделирования сложной системы состоит в проведении на компьютере эксперимента с моделью.

Модель – некоторый программный комплекс, описывающий формально или алгоритмически поведение системы.

Основные требования к модели:

- **полнота** - возможность получения необходимых характеристик;
- **гибкость** - возможность воспроизводить различные ситуации при варьировании структуры, алгоритма и параметров модели. **Блочная**

структура – возможность замены, добавления, исключения некоторых частей схемы без переделывания всей модели.

- компьютерная реализация модели должна **соответствовать** имеющимся технологическим **ресурсам** (быстродействие, память и т.д.)

Процесс моделирования, включая разработку и машинную реализацию модели, является итерационным.

Этот итерационный процесс продолжается до тех пор, пока не будет получена модель, которую можно считать адекватной в рамках поставленной задачи. Вычленяется объект на базе некоторого анализа, синтезируется, и снова начинает анализироваться. ТЗ → Анализ → Синтез (заикливание)

Визуальные средства отображения всякого о модели в процессе работы программы в GPSS

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

- блоки (Blocks) – вход транзактов в блоки
- выражения (Expression) – наблюдение за PLUS выражениями
- устройства (Facilities) – наблюдение за сост устр-в
- логические ключи (Logicswitches)
- матрица (Matrix)
- график (Plot) – графики
- очереди (Queues) – сост. очередей
- ячейки (Savevalues)
- памяти (Storages)
- таблицы (Table) – гистограммы

Билет 8

Основные этапы моделирования сложных систем

Этапы:

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

На **первом** этапе – формализация. От описания к мат. модели.
Последовательность действий:

1. проведение границы между системой и внешней средой;
2. исследование модели объекта с точки зрения выделения основных составляющих (на основании цели моделирования);
3. переход от содержательного описания к формальному описанию системы (к концептуальной модели) этот переход сводится к исключению из рассмотрения некоторых второстепенных признаков;
4. оставшиеся элементы группируются в блоки:
 - a. блоки имитаторы воздействия внешней среды
 - b. блоки-модели функции системы
 - c. блоки-модели функции системы (н-р, сбор статистики)
5. процесс функционирования системы разбивается на подпроцессы, чтобы построение модели отдельных подпроцессов было элементарно и не вызывало особых трудностей (элементарно => до типовых математических моделей).

На **втором** этапе – алгоритмизации и реализация. Мат модель в программную модель. Последовательность действий:

1. разработка схемы модели алгоритма (блочная схема модели);
2. разработка диаграммы классов;
3. программирование модели;
4. отладка программы (отладка, тестирование);
5. проверка достоверности на примерах;
6. составление технической документации.

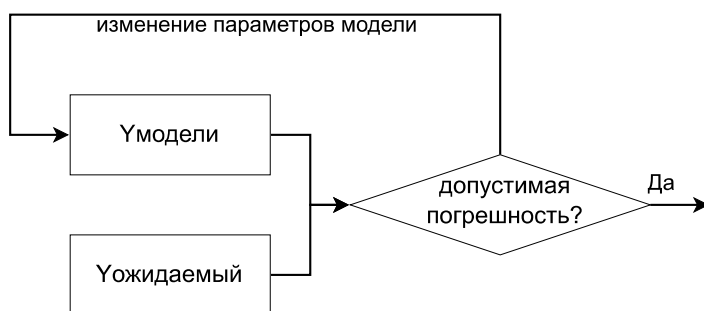
На **третьем** этапе – получение и интерпретация результатов => сделать выводы о характеристиках процессов функционирования исследуемой схемы и об адекватности модели.

Последовательность действий:

1. Планирование эксперимента с моделью (стратегическое и тактическое). *Стратегическое планирование* - разработка условий проведения эксперимента, определение режимов, обеспечивающих наибольшую информативность эксперимента.

Тактическое планирование обеспечивает достижение заданных точности и достоверности результатов.

2. проведение рабочих расчетов с калибровкой модели



3. статистическая обработка результатов отчета

4. интерпретирование результатов модели

5. подведение итогов.

Способ моделирования СМО в GPSS

Для представления собственно обслуживания используются определенные элементы. Такими элементами могут быть либо люди, либо какие-то предметы. Независимо от этого подобные элементы в GPSS называют объектами аппаратной категории, к которой относят одноканальные (ОКУ) и многоканальные устройства (МКУ) и логические ключи (переключатели). При моделировании возможны следующие режимы организации функционирования ОКУ:

1. занятие **SEIZE** и его освобождение **RELEASE**;
2. прерывание **PREEMPT - RETURN** обслуживания;
3. недоступность **FUNAVAIL** и восстановление доступности **FAVAIL**.

Если ОКУ занят, то новый транзакт должен либо подождать в очереди, либо перейти в другое место, либо если это прерывание – прервать обслуживание тек транзакта. Поскольку ОКУ в модели может быть много, для различия им дают символические имена.

Два или более обслуживающих устройств могут быть смоделированы на GPSS двумя или более ОКУ, располагаемыми рядом, т. е. параллельно. Так нужно поступать, когда отдельные устройства являются разнородными (н-р, разная интенсивность обслуживания). Однако часто различные параллельно работающие устройства являются однородными. GPSS представляет для моделирования однородных параллельных устройств специальное средство, именуемое многоканальным устройством. МКУ может быть использовано несколькими транзактами одновременно. Ограничений на число МКУ в модели нет. Для различия им дают имена. МКУ определяется до его использования командой:

NameStorage STORAGE A, где NameStorage – имя МКУ, A – число каналов.

В модели можно организовать функционирование МКУ в двух режимах:

1. занятие **ENTER** и освобождение **LEAVE** МКУ;
2. недоступность/неисправность **SAVAIL** МКУ и восстановление **SUNAVAIL**.

Билет 9.

Проверка адекватности модели

Проверка адекватности - анализ соразмерности модели и равнозначности системы. Адекватность нарушается из-за идеализации внешних условий и пренебрежения случайными факторами.

Модель адекватна с системой, если вероятность того, что отклонение параметров Δy не превышает некоторой предельной величины δ больше допустимой вероятности.

$$\Delta y = |y_{\text{оригинала}} - y_{\text{модели}}|, \delta = |y_{\text{оригинала}} - y_{\text{модели}}| / y_{\text{оригинала}}$$

На практике не работает, т.к. нет значений $y_{\text{оригинала}}$ и значения могут быть случайными величинами.

На практике нужно проводить экспертную оценку результатов моделирования по всем параметрам. Проверка:

- 1) моделируемых элементов
- 2) модели входных воздействий
- 3) концептуальной модели
- 4) как измерили/оценили значения выходных характеристик
- 5) программной модели

Проверка приводит к изменениям модели:

- 1) глобальные - возникают в случае обнаружения методических ошибок концепт. или мат. модели
- 2) локальные - связаны с уточнением параметров и алгоритмов => Замена воздействий на более точные эквивалентные.

- 3) параметрические - изменение калибровочных параметров

Зафиксировать область применения модели – множество условий при соблюдении которых точность результатов моделирования находится в допустимых пределах.

QUEUE, RELEASE

QUEUE A,[B] - этот блок осуществляет сбор статистики об очереди. **A** - номер очереди, в которую заносится транзакт. При входе транзакта в данный блок текущая длина очереди увеличивается на число единиц, указанное в поле **B** по умолчанию 1. Обновляется (если необходимо) максимальная длина очереди и счетчик общего числа единиц прошедших через очередь увеличивается на то же число единиц. Для выхода из очереди используется **DEPART A,[B]**, который уменьшает длину очереди на **B** единиц.

Транзакты перемещаются в модели от блока к блоку. Если транзакт занимает ОКУ (одноканальное устройство), то для этого он пытается войти в соответствующий блок, описывающий это ОКУ.

ОКУ обладает следующими свойствами:

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

Вход транзакта в блок **SEIZE** моделирует занятие ОКУ. После обслуживания вход того же транзакта в другой блок моделирует освобождение ОКУ - состояние ОКУ меняется с "занято" в "незанято".

RELEASE A - переводит статус ОКУ в состояние «свободен». **A** - содержит имя ОКУ

Пример (ADVANCE с распределением Пуассона и $\lambda=3$):

```
GENERATE 10,5
QUEUE Queue1
SEIZE 1
DEPART Queue1
ADVANCE (POISSON(1, 3));  $P(X=k)=\frac{\lambda^k}{k!}e^{-\lambda}$ 
RELEASE 1
QSTAT QTABLE Queue1,0,5,100
TERMINATE 1
```


Билет 10

Моделирование на системном уровне. Схема вычислительной установки

Уровни проектирования:

1. **Системный**. - рассматриваются процессор, память, внешние устройства. Используются типовые матсхемы - СМО, А-схемы, сети Петри, вероятностные и конечные автоматы.
2. **Функционально-логический** - интересуется реализация правильного функционирования целиком какого-либо блока
 - а. подуровень регистровых передач. Элемент моделирования - регистры, сумматоры; всё разделяется на комбинационные и последовательные схемы. Используются конечные автоматы
 - б. логический уровень. Моделируем работы ЛЭ. Используются конечные автоматы.
3. **Схемо-технический** - Рассматривается правильность срабатывания отдельных элементов. В основном модели ОДУ и ДУ
4. **Конструкторский** – Решение вопросов связаны с компоновкой системы (конструирование многослойных плат), вопросы отвода тепла, вентиляции, оптимального размещения интегральных схем (оптимизация длины проводников). Используется ДУ.

Моделирование на системном уровне.

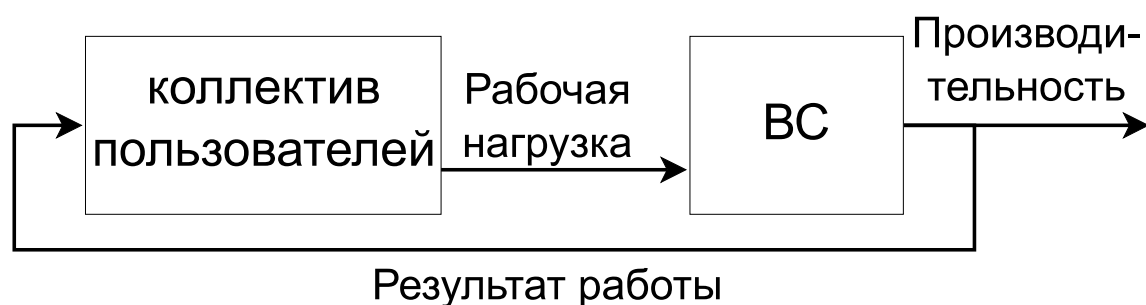
Объект моделирования - вычислительная система (ВС).

ВС - комплекс аппаратных и программных средств, которые в совокупности выполняют некоторые рабочие функции.

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, структурой межмодульных связей, режимами работы, алгоритмами управления и т.д.

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

Рабочая нагрузка - входные сигналы (программы, данные, команды), которые создаются коллективом пользователей.



Индекс производительности – описатель, который используется для представления производительности системы.

Индексы производительности:

1. Качественные
 - а. легкость использования системы
 - б. мощность системы команд
2. Количественные
 - а. пропускная способность $\frac{\text{объём инфы.}}{\text{ед. время}}$
 - б. время ответа – время между предъяв вход. данных и появлением выходной информации.
 - с. коэффициент использования оборудования $\frac{\text{время использ.}}{\text{время моделир}}$

Концепт. модель ВС включает сведения о выходных и конструктивных параметрах системы, её структуре, особенностях работы каждого элемента и ресурса, характера взаимодействия между ресурсами, постановка прикладных задач, определение цели моделирования.

Основные задачи:

1. Определение принципов организации ВС.
2. Выбор архитектуры, уточнение функций ВС и их декомпозиция
3. реализация аппаратным и программным путем.
4. Разработка структурной схемы – определение состава устройств и способов их взаимодействий.
5. Определение требований к выходным параметрам устройств и ТЗ на разработку устройств для функц.-лог. уровня проектирования.

Table, tabulate

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.).

NameTable TABLE A,B,C,D

NameTable – имя таблицы (обязательный параметр)

A – табулируемая величина (что заносим в гистограмму)

B – верхняя граница левого интервала (в 1 столбец попадут $x \in [b)$)

C – ширина интервалов

D – число интервалов.

Инфа накапливается посредством попадания транзакта в блок
TABULATE NameTable

Для очередей NameQTable QTABLE A,B,C,D (аналогично TABLE) и не требуется TABULATE. Этот оператор собирает инфу о том, сколько времени провели транзакты в очереди A. Пример:

```
Generate 10,2; генерация поступления заявок 10 +- 2
LenTable TABLE Q$QueueR,0,1,12 ; таблица с инфой по длине очереди
TimeTable QTABLE QueueR,0,3,18 ; таблица с инфой о времени в очереди
QUEUE QueueR ; очередь
SEIZE Comp1 ; занять ОА
TABULATE LenTable ; занести инфу о длине очереди в таблицу LenTable
DEPART QueueR ; покинуть очередь
ADVANCE 10,4; задержка 10 +- 4
RELEASE Comp1 ; освободить ОА
Terminate 1
```

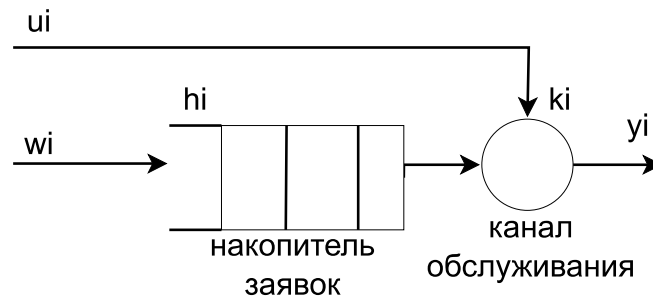
Чтобы увидеть, что получилось запускаем симуляцию, после
Window->Simulation Window->Table Window и указываем имя таблицы.

Билет 11.

Q-схемы

При непрерывно-стохастическом подходе в качестве типовых математических схем применяются Q-схемы. Q-схемы используются для моделирования систем массового обслуживания (СМО).

Элементарный акт обслуживания состоит из ожидания обслуживания и обслуживания заявки. Это можно изобразить в виде i-того прибора обслуживания (обслуживающего аппарата – ОА):



Прибор обслуживания Π_i состоит из:

- накопителя заявок H_i емкостью L_i^H
- канала обслуживания K_i

В обслуживающем приборе два входных потока:

- поток заявок W_i в накопитель H_i
- поток обслуживания U_i на вход канала K_i

Центральным понятием является понятие потока заявок. Для СМО характерно случайное появление заявок на обслуживание и случайное распределение времени на обработку заявки.

Поток событий – последовательность событий, происходящих одно за другим в случайные моменты времени. Поток событий однороден, если он характеризуется только моментами поступлений этих событий. Поток неоднороден, если задается ещё и набором признаков события (н-р, приоритет). Если интервалы между сообщениями случайны и независимы, то поток с ограниченным последствием.

ПС **ординарен**, если вероятность того, что на момент времени t придется 2 и более событий, пренебрежительно мала. Среднее число сообщений, поступивших за $\Delta t \rightarrow 0$, где Δt примыкает к моменту времени t , равно $p_1(t, \Delta t)$. Среднее число сообщений, поступивших за интервал Δt

$\lim_{\Delta t \rightarrow 0} \frac{p_1(t, \Delta t)}{\Delta t} = \lambda(t)$ – интенсивность ординарного потока. ПС стационарен, если $\lambda(t) = \text{const}$ = среднее число событий в единицу времени.

Процесс функционирования ОА можно представить как изменение состояний его элементов во времени. Если в Q-схеме, образованной комбинацией многих устройств обслуживания Π_i , каналы различных устройств объединены параллельно, то это многоканальная Q-схема, если последовательно – многофазная. Оператор сопряжения R описывает связь элементов между собой.

Q-схема задается кортежем $Q=(W,U,R,H,Z,A)$: W-поток заявок, U-поток обслуживания, R-оператор сопряжения, H-множество собственных параметров, Z-множество состояний, A-алгоритм функционирования.

Параметры Q-схемы: количество фаз, количество каналов в каждой фазе, количество накопителей, емкости накопителей.

Блоки SEIZE и RELEASE

Транзакты в GPSS перемещаются от блока к блоку. Если транзакт занимает одноканальное устр-во, он входит в соотв-щий блок, описывающий это устр-во.

Блок SEIZE (вид): метка SEIZE <имя устройства: A>.

Блок RELEASE (вид): метка RELEASE <имя устройства: A>.

Если транзакт входит в блок SEIZE, то устройство A становится занятым, пока транзакт не пройдет соотв. блок RELEASE A, который освобождает устройство. Если устройство, указанное в поле A блока SEIZE, уже занято каким-либо транзактом, то транзакт становится в очередь и ожидает освобождения ОА. Транзакты, задержанные (заблокированные) перед блоком SEIZE, остаются в списке текущих событий и при освобождении устройства обрабатываются с учетом приоритетов и очередности поступления. Каждое устройство имеет следующие СЧА: F - состояние устройства (0 - свободно, 1 - занято); FR - коэффициент использования в долях 1000; FC - число занятий устройства; FT - целая часть среднего времени занятия устройства.

Билет 13.

Марковские случайные процессы. Многоканальная СМО с отказами.

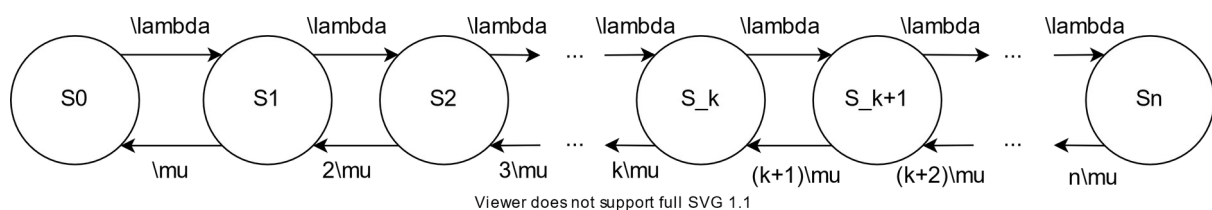
Случайный процесс, протекающий в некоторой системе, называется **Марковским**, если для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от состояния системы в настоящем и не зависит от того, каким образом система пришла в это состояние (не зависит от прошлого).

Для Марковского процесса как правило составляют уравнение Колмогорова, общий вид: $F(P'(t), P(t), \Lambda) = 0$. Для стационарного распределения это уравнение принимает вид: $p = P(P(t), \Lambda) = P(\Lambda)$. Тогда выходная характеристика будет $Y = P(\Lambda)$ – это называется базисной

моделью. С учётом, что интерфейсная модель $Y=Y(X, V, H)$, то $\Lambda=\Lambda(X, V, H)$.

Стационарное -- распределение вероятностей не меняется с течением времени. Вывод уравнений Колмогорова для СМО в общем виде -- в левой части производная вероятности состояния, а в правой столько членов, сколько стрелок связано с этим состоянием: “+” если стрелка в состояние, “-” если из. Каждый член равен плотности вероятности перехода * вероятность состояния, из которого стрелка.

Дана многоканальная СМО с отказами:



на слева направо систему переводит один и тот же поток заявок с интенсивностью λ . Одна заявка обслуживается с интенсивностью μ .

Состояния системы:

S_0 - все каналы свободны

S_k - занято k каналов, остальные свободны

S_n - все n каналов заняты

Уравнения Колмогорова для этой СМО:

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

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

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

Предельные вероятности состояний $p_0 \dots p_n$ характеризуют установившийся режим работы СМО $\Rightarrow p'_i = 0$ при $t \rightarrow \infty$. Зная все вероятности состояний $p_0 \dots p_n$ можно найти характеристики СМО:

1. Вероятность отказа

$$p_n = \frac{\rho^n}{n!} p_0 = \frac{\rho^n}{n!} \cdot \frac{1}{\sum_{k=0}^n \frac{\rho^k}{k!}}$$

$\rho = \frac{\lambda}{\mu}$ -- среднее число заявок приходящих в СМО за среднее время обслуживания одной заявки.

2. Относительная пропускная способность $q = 1 - p_n$

3. Среднее число заявок, обрабатываемых в единицу времени $A = \lambda q$

Блоки синхронизации транзактов.

метка **MATCH A**, где A -- метка другого оператора **MATCH**, называемого сопряженным с данным. Транзакт, поступивший в блок **MATCH**, блокируется на нем до тех пор, пока в сопряженный блок **MATCH** не войдет транзакт, принадлежащий тому же ансамблю. Когда в двух сопряженных блоках **MATCH** окажется по транзакту из одного ансамбля*, оба транзакта одновременно пропускаются дальше.

*транзакты генерируются блоком **GENERATE** или порождаются блоком **SPLIT**. Транзакт, прошедший через блок **SPLIT**, называемый транзактов-родителем, порождает какое-то количество транзактов - тогда все транзакты, порожденные одним родителем и сам родитель, принадлежат одному ансамблю.

метка **ASSEMBLE A** -- когда транзакт, принадлежащий какому-то ансамблю, приходит в блок **ASSEMBLE**, он блокируется там до тех пор, пока в этот блок не придет еще A-1 транзактов того же семейства, после чего из блока выходит первый транзакт, остальные уничтожаются.

метка **GATHER A** -- аналогично, но выходят все A транзактов.

Билет 14.

Марковские процессы. Уравнения Колмогорова.

Случайный процесс, протекающий в некоторой системе, называется Марковским, если он обладает следующим свойством: для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от состояния системы в настоящем и не зависит от того, каким образом система пришла в это состояние (не зависит от прошлого).

Для Марковского процесса как правило составляют уравнение Колмогорова, общий вид: $F(P'(t), P(t), \lambda) = 0$. λ -- вектор коэффициентов, присущих системе.

Для стационарного* распределения это уравнение принимает вид: $P(P(t), \lambda) = 0$, что позволяет вывести $P = P(\lambda)$. Зависимость $Y = Y(P(\lambda))$ -- это зависимость выходных параметров от некоторых внутренних параметров

системе, называемая базисной моделью. Для вывода зависимости $Y=Y(X, V, H)$ надо связать внутренние параметры модели λ с:

- конструктивными параметрами X
- неуправляемыми параметрами V
- и собственно структурой системы

В результате чего можно вывести зависимость $\lambda=\lambda(X, V, H)$, называемой интерфейсной моделью.

Вывод уравнений Колмогорова для СМО в общем виде -- в левой части производная вероятности состояния, а в правой столько членов, сколько стрелок связано с этим состоянием: “+” если стрелка в состояние, “-” если из. Каждый член равен плотности вероятности перехода * вероятность состояния, из которого стрелка. Рисуи граф и для него уравнения Колмогорова.

Блоки аппаратной категории GPSS: PREEMPT, RETURN.

метка **PREEMPT A, [B, C, D, E]**, где

A -- имя устройства,

B -- режим прерывания (по умолчанию прерывание без приоритета, PR -- с приоритетом)

C -- метка блока, куда направляется прерванный транзакт

D -- номер параметра прерванного транзакта, в который заносится оставшееся время выполнения

E -- режим удаления прерванного транзакта (по умолчанию не удаляем, RE -- удаляем)

Если устройство A свободно и в блок PREEMPT приходит транзакт, он захватывает устройство. Если транзакт захватил устройство посредством блока PREEMPT, то освобождать его он должен в блоке RETURN.

Если устройство занято:

- если режим работы приоритетный -- PR: прервать обслуживание предыдущего транзакта, обслужить транзакт с наибольшим приоритетом.
- если режим прерывания захвата -- по умолчанию: прервать обслуживание предыдущего транзакта, обслужить вновь пришедший транзакт.

Билет 15

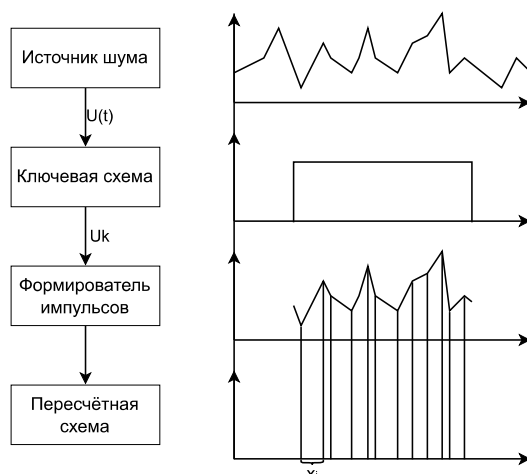
Способы получения последовательности случайных чисел

На практике используется три основных способа:

1. Аппаратный (спец. устройства)
2. Табличный (файл со случайными числами)
3. Алгоритмический (матан – программный)

Аппаратный способ:

При этом способе случайные числа формируются специальным устройством. Источником случайных чисел чаще всего являются шумы в электронных приборах. Временные расстояния между шумовыми всплесками, превышающими подобранный уровень ограничения, фиксируются как случайные числа из равномерного распределения $R[0, 1]$ **В аппаратном способе нельзя воспроизвести еще раз такую же последовательность.**



Способ	Достоинства	Недостатки
Аппаратный (считываем шум)	1) запас чисел не ограничен 2) мало вычислений 3) не занимает память	1) Могут быть неслучайны/или деградировать 2) Нельзя повторно воспроизвести послед ПСЧ 3) спец устройства 4) меры по обеспечению стабильности
Табличный (таблица с ПСЧ)	1) однократная проверка 2) возможность повторного	1) Запас чисел ограничен 2) Занимает место в памяти

	воспроизводства послед. ПСЧ	
Алгоритмический (мат. функция генерации ПСЧ)	1) однократная проверка 2) возможность повторного воспроизводства послед. ПСЧ 3) мало занимает памяти 4) нет внешних устройств	1) Запас ограничен периодом (комбинируй несколько ГСЧ) 2) вычислительные затраты

Блоки изменения параметров транзактов в GPSS

ALTER, ASSIGN, MARK

ASSIGN A,B,[C]

Операндом А задается номер параметра, которому присваивается значение. Операнд В определяет значение, которое следует добавить, вычесть или которым следует заменить значение в параметре, заданном операндом А. Операнд С задает номер модификатора-функции. При использовании операнда С значение операнда В умножается на значение модификатора-функции. ASSIGN 1,754.3

Блок **ALTER** изменяет приоритет или атрибуты сообщений, принадлежащих к данной группе. Изменять значение заданного атрибута для определенного числа членов группы.

Блок **MARK** [A] изменяет время рождения транзакта или заносит значение текущего абсолютного модельного времени в параметр транзакта. Блок всегда принимает транзакты. Если операнд А не определен, то блок заменяет значение времени рождения транзакта на текущее значение абсолютного условного времени. Время рождения используется для вычисления СЧА М1 (время пребывания транзакта в модели, равное разности текущего абсолютного условного времени и времени рождения транзакта).

Билет 16(25)

Алгоритмический способ получения последовательности псевдослучайных чисел

Алгоритмы:

- Выделение значения дробной части многочлена первой степени ($Y_n = A \cdot n + B$)
- **Способ срединных квадратов** (фон Неймана).
Каждое последующее число образуется возведением предыдущего в квадрат и отбрасыванием начальной и конечной цифр. Не используется т.к. между числами имеется сильная корреляция и при неудачном начальном значении последовательность может вырождаться (будет постоянной).
- **Линейный-конгруэнтный метод.**
 $x_{i+1} = (a x_i + b) \bmod M$ Обычно выбирают некоторую степень системы счисления p^q
 $a = 8t + 3$, где t любое целое положительное число.
- **Метод Фибоначчи.**
 $X_{n+1} = (X_n + X_{n-k}) \bmod M$ Рекомендуется $k > 15$

Достоинства

- Однократная проверка
- Многократное воспроизведение последовательности
- Места в RAM мало занимают
- Не требуется внешнее устройство

Недостатки (и их решения):

- Запас чисел ограничен периодом (используют несколько ГСЧ)
- Затраты машинного времени

Для неравномерного распределения $R[0, 1]$

$$F(t) = \int_{-\infty}^t f(x) dx = R = Y = F^{-1}(R)$$

Язык GPSS. Блоки. Объекты языка

Категории	Типы объектов
Динамическая	Транзакты
Операционная	Блоки
Аппаратная	Одноканальные устройства, памяти (многоканальные устройства), логические ключи (переключатели)
Вычислительная	Переменные, функции, генераторы случайных чисел
Статистическая	Очереди, таблицы
Запоминающая	Ячейки, матрицы ячеек
Группирующая	Числовые группы, группы транзактов, списки

Динамическими объектами являются *транзакты*, которые создаются в определенных точках модели, продвигаются планировщиком через блоки, а затем уничтожаются.

Объекты аппаратной категории - это абстрактные элементы, на которые может быть декомпозирована реальная система.

Операционные объекты, т. е. блоки, задают логику функционирования модели системы и определяют пути движения транзактов между объектами аппаратной категории.

Вычислительная категория служит для описания таких ситуаций в процессе моделирования, когда связи между компонентами моделируемой системы посредством процесса наиболее просто и компактно выражаются в виде математических (аналитических и логических) соотношений.

Объекты запоминающей категории обеспечивают обращения к сохраняемым значениям.

К статистическим объектам относятся очереди и таблицы. Учет очередей составляет одну из основных функций планировщика.

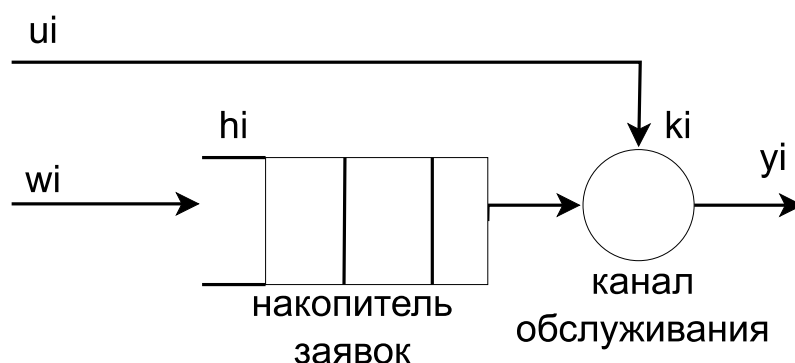
К группирующей категории относятся три типа объектов: числовая группа, группа транзактов и списки.

Билет 17

Моделирование потоков событий

Q-схема реализует непрерывно-стохастический подход, используется для моделирования СМО. Для СМО характерно случайное появление заявок на обслуживание и случайное распределение времени на обработку.

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



i -й прибор обслуживания, состоит из накопителя заявок и канала обслуживания. w_i - поток заявок в накопитель, u_i - поток заявок в канал.

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

ПС **однороден**, если он характеризуется только моментами поступлений этих событий. Поток **неоднороден**, если он задается еще и набором признаков события f : $\{t_n, f_n\}$ n -р, приоритетом.

Если интервалы между сообщениями случайны и независимы, то это 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. Интенсивность ординарного ПС

$$= \lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{p_1(t, \Delta)}{\Delta t}. \text{ Среднее число сообщений, поступающих за } \Delta t \rightarrow 0: p_1(t, \Delta).$$

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 интервал был взят. Для стационарного ПС интенсивность потока равна среднему числу событий в единицу времени и $\lambda(t) = \lambda = const$.

Процесс функционирования - процесс изменения состояний элементов. Переход в новое состояние - изменение количества заявок.

В практике моделирования элементарные Q-схемы обычно объединены, при этом если каналы различных устройств соединены параллельно то имеет место многоканальное обслуживание, если параллельно - многофазное. Для соединения необходимого использовать некоторый оператор сопряжения R.

Параметры Q-схемы:

- количество фаз
- количество каналов в каждой фазе
- количество накопителей в каждой фазе
- емкость накопителя (если $= 0$, то система с потерями, если $\rightarrow \infty$ — система с ожиданием, иначе - смешанного типа)

Для задания Q-схемы необходимо описать алгоритм функционирования – набор правил поведения заявок в системе.

В общем случае Q-схема описывается кортежем $Q=(w,u,R,H,Z,A)$, где w – входные данные, u – поток обслуживания, R – оператор сопряжения, H – множество внутренних параметров, Z – множество состояний для определений закона функционирования, A – алгоритм функционирования.

Блоки SPLIT/ASSEMBLE

Блок **ASSEMBLE A** - для объединения определенного числа транзактов, являющихся членами одного ансамбля. **A** - число объединяемых ансамблей. Транзакты, принадлежащие одному ансамблю, будут задерживаться в блоке **ASSEMBLE** до тех пор, пока не поступит заданное число транзактов этого ансамбля. В результате на выходе блока появляется один (первый) транзакт ансамбля, а остальные транзакты уничтожаются. В одном блоке **ASSEMBLE** могут накапливаться транзакты разных ансамблей, транзакты одного ансамбля могут накапливаться в разных блоках **ASSEMBLE**.

SPLIT A, B, C, D создает **A** копий текущего транзакта. Копии входят в блок, указанный в поле **B**, а текущий транзакт продолжает движение в следующий блок. Номера создаваемых копий транзактов записываются в параметры, номера которых указываются в поле **C**. **D** - число параметров у транзактов-копий. Если поле **D** не задано, то число и тип параметров у копий такие же, как и у исходного транзакта.

Пример: SPLIT 4, THE Создает 4 копии транзакта, которые посылаются в блок по метке THE, а текущий транзакт продолжает движение в следующий блок.

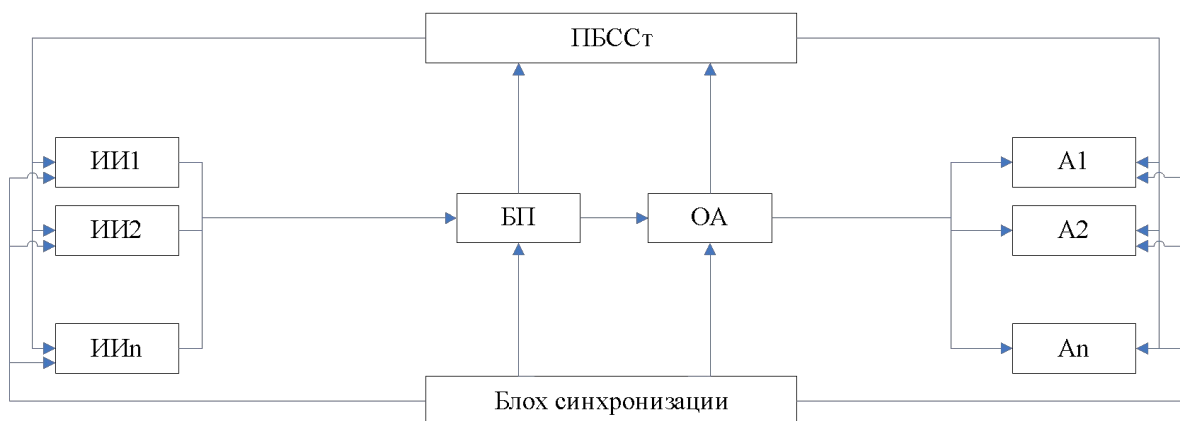
Билет 18

Методика построения программной модели(+)

Методика построения программной модели ВС.

Для разработки программной модели исходная система должна быть представлена как *стохастическая система массового обслуживания*. Это можно объяснить следующим: информация от внешней среды поступает в случайные моменты времени, длительность обработки различных типов информации может быть в общем случае различна. Т.о. внешняя среда является генератором сообщений. А комплекс вычислительных устройств (ВС) – обслуживающими устройствами.

Обобщенная структурная схема ВС.

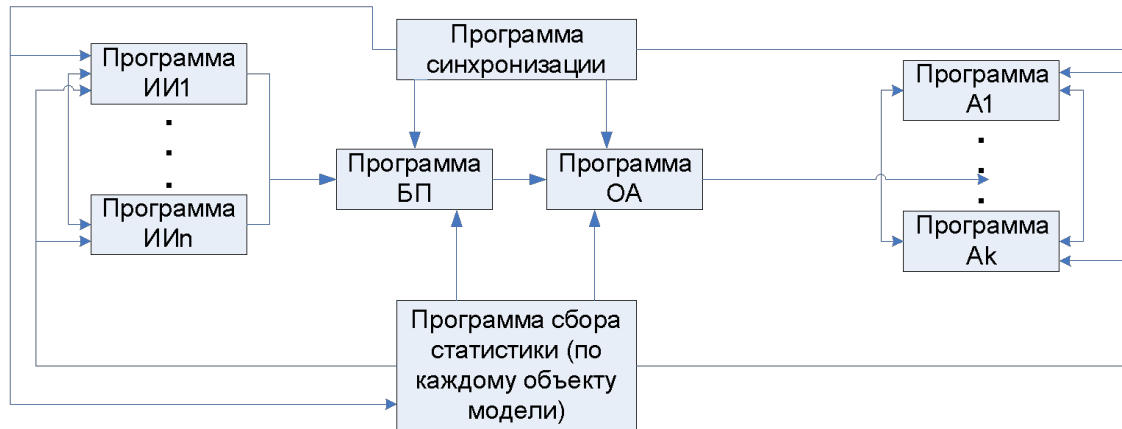


ИИ – источники информации – выдают на вход буферной памяти (БП) независимые друг от друга сообщения. Закон появления сообщений – произвольный, но задан наперед.

В **БП** (буферной памяти) сообщения записываются «в навал» и выбираются по одному в обслуживающий аппарат (ОА) по принципу FIFO/LIFO. Длительность обработки одного сообщения в **ОА** в общем случае так же может быть случайной, но закон обработки сообщений должен быть задан. Т.к. быстродействие ОА ограничено то на входе системы в БП возможно сложение данных ожидающих обработки.

А – абоненты.

Программная модель из этой системы создается следующим образом:



Должна быть обязательно программа сбора статистики (**ПБССт** – программный блок сбора статистики). Причем статистику программа должна собирать по каждому из объектов модели. Так же должна быть программа, которая позволит "оживить" систему – это **программа синхронизации (блок синхронизации)**, которая покажет когда и в какое время будут активизированы те или иные фрагменты модели.

Язык GPSS. Блоки создания и уничтожения транзактов

GENERATE A,B,C,D,E

Функцией данного блока является создание транзактов входящих в систему. В поле **A** задается среднее время между поступлением отдельных транзактов. Как и в блоке **ADVANCE**, это поле может быть модифицировано с помощью модификатора находящегося в поле **B** (также интервал или функция). В поле может быть записан NULL. Если при вычислении времени появления в системе 1-ого транзакта, оно получилось равным 0, то симулятор полагает его равным 1. Задаваемый модификатором интервал не должен превосходить среднего, записанного в поле A. В поле **C** записывается начальная задержка. Заданное в этом поле число без модификации определяет интервал времени до создания данным блоком первого транзакта. По отношению к **A** оно может быть любым. Поле **D** задает число транзактов, которое должно быть создано блоком **GENERATE**. Если это поле пусто, то блок генерирует неограниченное число транзактов. В поле **E** задается приоритет присваиваемый генерируемому транзакту. Если поле пусто, то нулевой приоритет.

TERMINATE A

Удаляет транзакты из системы. Он используется для обозначения окончания пути транзакта. Поле **A** указывает изменяет ли этот блок содержимое счетчика завершения в момент поступления транзакта и, если изменяет, то на сколько единиц.

Билет 23.

Событийный принцип протяжки модельного времени.

Характерное свойство моделируемых систем -- состояния отдельных устройств изменяются в дискретные моменты времени:

- моменты поступления заявок в систему
- моменты времени окончания решения задач
- моменты возникновения аварийных ситуаций

Моделирование и продвижение текущего времени в системе может быть проведено с использованием событийного метода: состояние блоков имитационной модели анализируется только в момент появления события. Моменты наступления следующих событий определяются минимальным значением из списка будущих событий. Список будущих событий -- набор ближайших моментов изменения состояния каждого из блоков системы. Достоинство: меньшие вычислительные затраты по сравнению с итеративным методом, не будет пропущено ни одного события. Недостаток: при большом количестве событий, необходимо постоянно просматривать список будущих событий для поиска минимума, либо хранить сразу в отсортированном порядке.

Классификация блоков GPSS.

По назначению вводится следующая классификация блоков:

1. Блоки, осуществляющие модификацию атрибутов транзактов.

1.1. Временная задержка:

1.1.1. ADVANCE (при использовании функций, потребуется производить масштабирование)

1.2. Генерирование и уничтожение транзактов:

1.2.1. GENERATE (второй параметр – модификатор)

1.2.2. TERMINATE

1.2.3. SPLIT – берёт транзакт и создаёт копии одного «ансамбля» транзактов

1.2.4. ASSEMBLE

1.3. Синхронизация движения нескольких транзактов:

1.3.1. MATCH

1.3.2. GATHER

1.4. Изменение параметров транзактов:

1.4.1. ASSIGN

1.4.2. INDEX

1.4.3. MARK

1.5. Изменение приоритета

1.5.1. PRIORITY

2. Блоки, изменяющие последовательность передвижения транзактов (блоки передачи управления)
 - 2.1. TRANSFER
 - 2.2. LOOP
 - 2.3. TEST (устройство – занято ли, память – свободна ли)
 - 2.4. GATE
3. Блоки, связанные с группирующей категорией
 - 3.1. JOIN
 - 3.2. REMOVE
 - 3.3. EXAMINE
 - 3.4. SCAN
 - 3.5. ALTER
4. Блоки, сохраняющие необходимые значения для дальнейшего использования
 - 4.1. SAVEVALUE
 - 4.2. MSAVEVALUE
5. Блоки, организующие использование объектов аппаратной категории
 - 5.1. Устройств:
 - 5.1.1. SEIZE
 - 5.1.2. RELEASE
 - 5.1.3. PREEMPT
 - 5.1.4. RETURN
 - 5.1.5. FAVAIL (facilities avail)
 - 5.1.6. MFAVAIL
 - 5.1.7. FUNAVAIL
 - 5.2. Памяти:
 - 5.2.1. ENTER
 - 5.2.2. LEAVE
 - 5.2.3. SAVAIL
 - 5.2.4. SUNAVAIL
 - 5.3. Ключи:
 - 5.3.1. LOGIC
6. Блоки, обеспечивающие получение статистических результатов
 - 6.1. QUEUE
 - 6.2. DEPART
 - 6.3. TABLE
 - 6.4. TABULATE
7. Специальные TRACE, SELECT, HELP
8. Блоки для организации цепей
 - 8.1. LINK

8.2. UNLINK

9. Вспомогательные блоки

9.1. WRITE

9.2. SAVE

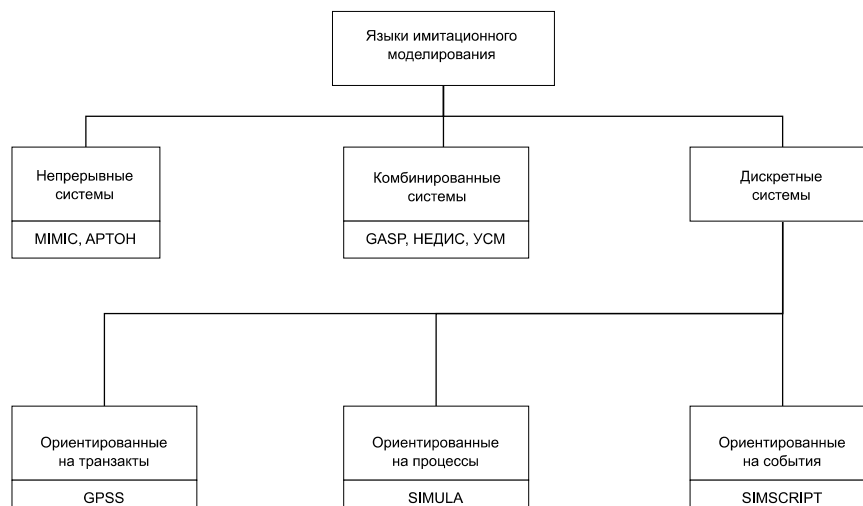
9.3. LOAD

9.4. REPORT

9.5. UPDATE

Билет 24

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.

Язык GPSS. Построение и основные принципы

General Purpose Simulation System - язык общецелевой системы моделирования. Возможности:

- Многозадачность
- Использование виртуальной 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 26.

Типы СМО для моделирования сложных систем

СМО классифицируется по:

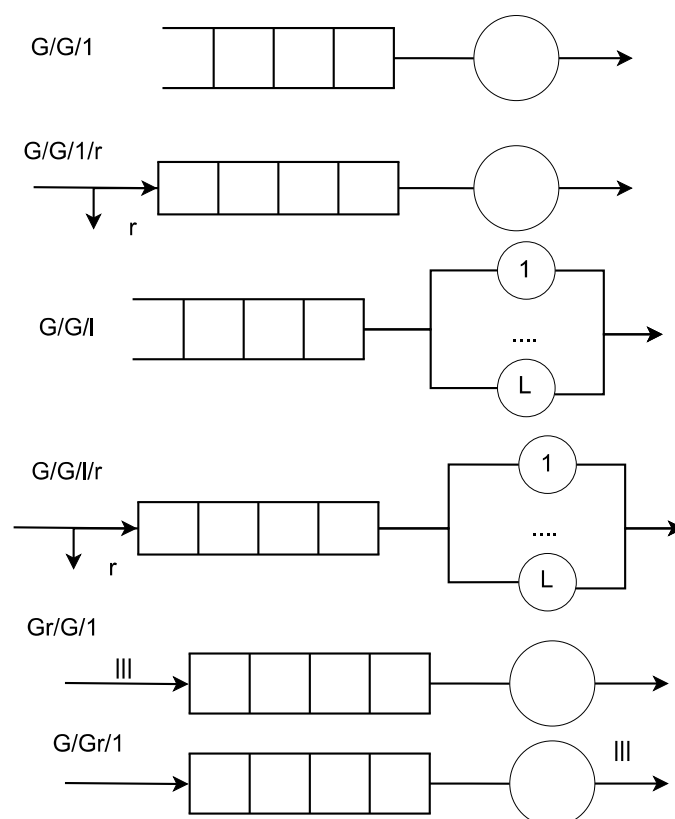
1. закону распределения входного потока
2. число обслуживающих приборов
3. закону распределения времени (в обслуживающих приборах)
4. числу мест в очереди
5. дисциплине обслуживания

Обозначается как **A/B/C/D/E**, где

1. **A** - закон распределения интервалов времени между поступлениями заявок. Примеры: **G** - random, **M**-экспоненциальное, **E**-эрлангово, **H**-гипер-экспоненциальное, **(Г)**-гамма, **D**-детерминированное (равные интервалы времени), **Gr** - групповое
2. **B** - закон распределения времени обслуживания (те же обозначения)
3. **C** - число обслуживающих приборов. Для одноканальной – **1**, для многоканальной – **l**.
4. **D** – число мест в очереди **r**. Если пропущено - **неограничено**
5. **E** - Дисциплина обслуживания (**fifo** (по умолчанию), **lifo**, **random**).

Примеры СМО:

1. **G/G/1** Одноканальное СМО с ожиданием. Представляет собой один обслуживающий прибор с бесконечной очередью. Является наиболее распространенной при исследовании СДС. Формализует функционирование практически любого узла вычислительной сети.
2. **G/G/1/r** Одноканальная СМО с потерями. Один обслуживающий прибор с конечным числом мест в очереди. Используется при моделировании каналов передачи в вычислительных сетях.
3. **G/G/I** 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. **G/G/I/r** Многоканальные СМО с потерями. Наиболее часто используются при моделировании работы каналов.
5. **Gr/G/1** Одноканальные СМО с групповым поступлением заявок. Также как и одноканальная СМО с групповым обслуживанием заявки используются для моделирования центров коммутации.



Отладка в GPSS

Отладка модели производится в области работы с документами текстового редактора. Блок модели, который будет выполнен на следующем шаге, выделяется желтым прямоугольником. В начале моделирования — это

блок образования первого транзакта. Чтобы выполнить текущий блок и перейти к следующему блоку, необходимо выполнить команду «Шаг вперед». Отладчик позволяет выполнять моделирование по шагам, последовательно перемещаясь по блокам, а также двигаться «скачками», по контрольным точкам (точкам останова).

Билет 27.

Сети массового обслуживания: открытые, замкнутые, смешанные

Вычислительные сети могут быть в общем виде представлены в виде сети массового обслуживания (сеть МО).

Сети МО:

1. **Открытая.** m узлов причем хотя бы в один из узлов сети поступает извне входящий поток заявок и имеется сток заявок. Поступление заявок не зависит от состояния сети. Такие сети используются как правило, для исследования функционирования вычислительной сети, работающей в неоперативном(?) режиме.
2. **Замкнутая.** Сеть МО с множеством узлов без источника и стока, в которой циркулирует постоянное число заявок. Используются для моделирования таких вычислительных систем, источниками информации для которых служат абонентские терминалы, работающие в диалоговом режиме. Терминал - многоканальная Сеть МО с ожиданием и включается в состав сети. Два режима работы терминалов
 - а. **Простой.** Абоненты только отправляют задания в вычислительную систему и обдумывают ответ от системы.
 - б. **Сложный.** Представляется в виде совокупности операций некоторого процесса называемого **технологическим**. Каждая операция технологического процесса моделируется соответствующей сетью массового обслуживания.
3. **Смешанные.** СМО в которой циркулирует несколько различных типов заявок (трафик). 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. Для диалогового также сложный и простой режимы.

Сети Петри

Сеть Петри – это мат модель дискретных динамических систем (параллельных программ, ОС, компьютеров и их устройств, сетей компьютеров и т.п.), ориентированных на качественный анализ и синтез таких систем, а именно обнаружение блокировок, тупиковых ситуаций, "узких" мест автоматический синтез параллельных программ и компонентов компьютера на разных уровнях проектирования)

Формальное описание

$$PN = \{ \Theta, P, T, F, M_0 \}$$

где Θ – мн-во дискретные моменты времени

$P \neq \emptyset$ – непустое мн-во элементов сети, называемые позициями

$T \neq \emptyset$ – непустое мн-во элементов сети, называемые переходами

$P \cap T = \emptyset$ – мн-во позиций не пересекается с мн-вом переходов

F – функция инцидентности

$$F: (P \times T) \cup (T \times P) \rightarrow \{0, 1, \dots, k\}$$

где k – кратность дуги.

M_0 – начальная маркировка.

Функцию инцидентности F можно представить в виде двух отображений:

$F^P(p, t) = P \times T \rightarrow \{0, 1, \dots, k\}$ – для каждой позиции указываются переходы, связанные с ней с учётом кратности.

$F^T(t, p) = T \times P \rightarrow \{0, 1, \dots, k\}$ – для каждого перехода указываются позиции, связанные с ним с учётом кратности.

Функции инцидентности в общем случае зависят от времени и могут быть представлены в виде матриц инцидентности.

$$F^P = \begin{pmatrix} f_{11}^P & \dots & f_{1n}^P \\ \vdots & \ddots & \vdots \\ f_{k1}^P & \dots & f_{kn}^P \end{pmatrix}$$
$$F^T = \begin{pmatrix} f_{11}^T & \dots & f_{1k}^T \\ \vdots & \ddots & \vdots \\ f_{n1}^T & \dots & f_{nk}^T \end{pmatrix}$$

Из вершины позиции $p_i \in P$ ведёт дуга в вершину перехода $t_j \in T$ тогда и только тогда, когда $f_{ij}^P > 0$. В этом случае говорят, что t_j – выходной переход позиции p_i .

$P^j = \{p_k : f_{kj}^P > 0\}$ – мн-во позиций, для которых t_j – выходной переход.

Аналогично, из вершины перехода $t_j \in T$ ведёт дуга в вершину позиции $p_i \in P$ тогда и только тогда, когда $f_{ji}^T > 0$. В этом случае говорят, что p_i – выходная позиция перехода t_j .

$T^i = \{t_k : f_{ki}^T > 0\}$ – мн-во переходов, для которых p_i – выходная позиция.

Каждая позиция p_i содержит некоторый целочисленный ресурс $\mu(p_i) \geq 0$, что отображает число фишек внутри позиции. Каждой фишке соответствует некоторый ресурс. Маркировка $M = \{\mu_1, \mu_2, \dots, \mu_k\}$ называется разметкой сети Петри.

Динамическое поведение моделируемой системы описывается в терминах функционирования сети Петри. Она функционирует в дискретном времени по тактам в асинхронном режиме. Смена маркировки происходит в результате срабатывания перехода сети.

Переход $t_j \in T$ может сработать при некоторой маркировке M , если $\forall p_i \in P \mu_i(\Theta) - f_{ij}^P \geq 0$ и $i \neq j$.

Для каждой входной позиции для данного перехода содержится как минимум столько фишек какова кратность дуги, ведущей к переходу.

В результате срабатывания перехода t_j маркировка $M_i(\Theta)$ сменяется на $M_i(\Theta+1)$ по следующему правилу:

$$\mu_i(\Theta+1) = \mu_i(\Theta) - f_{ij}^P(\Theta) + f_{ji}^T(\Theta)$$

Иными словами, переход t_j изымает из каждой своей входной позиции число фишек равное кратности входных дуг и посылает в каждую выходную позицию число фишек, равное кратности выходных дуг.

Если может сработать несколько, то случайно срабатывает 1 любой (важно, что случайно). Функционирование сети останавливается если при некоторой маркировке ни один из переходов сработать не может (тупиковая маркировка).

В силу недетерминированного функционирования сеть Петри может порождать различные последовательности срабатывания переходов при одной и той же начальной маркировке. Последовательности образуют слова в алфавите (переходов?), множество все возможных слов, порождённых сетью – язык сети Петри. Две сети Петри называются эквивалентными, если они порождают один и тот же язык.

В отличие от конечных автоматов (КА) в терминах, которых описывается в глобальном состоянии системы, сети Петри концентрируют внимание на локальных событиях (переходах) и на локальных условиях

(позициях) \Rightarrow в терминах сети Петри более реалистичнее, чем с помощью КА моделируется поведение распределённых асинхронных систем.

Теоретическо-графовое представление сети Петри

Сети Петри – двудольный ориентированный мультиграф. Он содержит позиции (кружочки), переходы (вертикальные планки), ориентированные дуги (стрелки), соединяющие позиции с переходами и переходы с позициями. Кратные дуги – несколько параллельных дуг.

Наличие кратных дуг в сети поэтому мультиграф, наличие двух типов вершин – двудольный, дуги имеют направление \Rightarrow ориентированный.

При выполнении сети Петри формируется две последовательности:

1. Последовательность маркировок M_0, M_1, \dots
2. Последовательность запущенных переходов t_{j_0}, t_{j_1}, \dots

Последовательности обычно представляют в виде дерева, узлами которого являются маркировки/переходы. В дереве маркировок могут встречаться повторяющиеся маркировки, поэтому дальнейшую постройку дерева ведут только для одной из них.

Основные свойства сетей Петри

$R(PN)$ – мн-во достижимых разметок сети Петри $PN = \{\Theta, P, T, F, M_0\}$.

1. Свойство ограниченность

Сеть называется ограниченной, если для любой позиции p и для любой достижимой в сети разметки M справедливо неравенство $M(p) \leq k$.

2. Свойство безопасности

Сеть называется безопасной, если для любой позиции p и для любой достижимой в сети разметки M справедливо неравенство $M(p) \leq 1$.

3. Свойство консервативности

Сеть называется консервативной, если сумма фишек во всех позициях постоянна при работе сети.

4. Свойство живости

Переход t_j называется потенциально живым, если существует достижимая из M_0 некоторая маркировка M , при которой переход t_j может сработать.

Переход t_j называется мертвым, если не существует достижимой из M_0 маркировки M , при которой переход t_j может сработать.

Переход t_j называется живым, если для любой достижимой из M_0 маркировки M переход t_j может сработать.