

1 Основные понятия моделирования.

1.1 Понятие моделирования.

Модели - упрощенное представление явления или объекта, которое позволяет изучать объект в более или удобной форме.

Моделирование - изучение объектов с помощью их моделей.

1. Выделить основные элементы объекта.
2. Построить модель.
3. Исследовать модель.
4. Получить результаты.

1.2 Виды моделирования.

1. *Концептуальное* - в виде некоторых понятий, определений, описаний взаимодействий.
2. *Физическое* - предполагает построение физического аналога некоторого объекта.
3. *Структурно-функциональное* - представляется в виде графиков, диаграмм, блок-схем, некоторых описаний взаимодействий.
4. *Математическое* - представляет собой формализованное описание объектов, использующее математические функции, выражения.
5. *Имитационное (simulation)* - предполагает существование программной реализации модели, которая выполняется на компьютере многократно.
6. *Компьютерное* - предполагает использование компьютерных технологий.

1.3 Математическое моделирование.

Анализ объекта заменяется анализом математической модели. Математическая модель это набор формальных соотношений между параметрами и переменными.

1.3.1 Три раздела математики:

1. *Дифференциальные и интегральные уравнения.* Такая модель - детерминированная.
2. *Случайные процессы.* Такая модель - детерминированная. Каждое последующее состояние меняется независимо от предыдущего.
3. *Исследование операций.* Используется для нахождения некоторых оптимальных параметров.

1.3.2 Два этапа:

1. Постановка задачи.
2. Исследование мат. модели.

1.4 Имитационное моделирование.

Предполагает существование пакетов программ для компьютерной имитации поведения объекта.

1.4.1 Четыре этапа:

1. Установление адекватности модели.
2. Идентификация параметров.
3. Оценка значимости параметров.
4. Структурное преобразование.

1.4.2 Процесс моделирования - фазы:

1. определение системы
2. формализация описания
3. подготовка данных
4. трансляция модели
5. оценка адекватности
6. планирование эксперимента
7. планирование прогонов
8. макетный эксперимент
9. анализ результатов

10. интерпретация

11. реализация

12. документирование

1. Определение системы - обрисовать круг задач, которые будут описывать систему, определить переменные и параметры.
2. Формализация - найти взаимосвязи между элементами и параметрами.
3. Подготовка данных - определение данных отражающих широкий спектр объекта.
4. Трансляция - представление модели в виде некоторого программного комплекса.
5. Оценка адекватности - проверка:
 - 1) Полнота учета основных факторов и ограничений.
 - 2) Соответствие исходных данных модели реальным.
 - 3) Наличие в модели всех данных, необходимых для накопления ответных величин.
 - 4) Синтаксическая корректность программного моделирования.
 - 5) Правильность преобразования данных в конечный результат.
 - 6) Осмысленность результатов при нормальных условиях.

1.5 Таблица. Классификация общих подходов моделирования.

	Подход	Примеры	Уровни	Атрибуты	Время	Топография	Вычисления	Структурные изменения
1	Модель, основанная на уровнях	ДУ, динамические системы.	Макро	Непрерывность	Аппроксимально непрерывно			Нет
2	Микроаналитические модели	Модель налогообложения	Микро / макро без обратной связи	Непрерывность, дискретность	Дискретное		Синхронные	Переменная структура модели
3	Модели дискретных событий	Модели очередей, управление технологическими процессами	Микро	Непрерывность, дискретность	Событийное	Граф	Асинхронные	Переменная структура модели
4	Мультиуровневые модели	Теория игр, соцсети	С обратной связью микро / макро	Непрерывность, дискретность	Непрерывное, дискретное		Синхронные	Переменная структура модели
5	Клеточный автомат	Модель разделения ресурсов	Микро	Дискретность	Дискретное	Решетка	Синхронная / асинхронная	Нет
6	Нейронные сети		Микро	Непрерывность, дискретность	Непрерывное, дискретное	Граф	Синхронная	Адаптация
7	Эволюционные алгоритмы		Микро	Дискретность	Дискретное		Синхронное	Адаптация

2 Принципы построения имитационных моделей.

2.1 Элементы имитационной модели.

1. состояние
2. событие
3. датчик случайных чисел
4. таймер
5. цепи событий
6. цель моделирования
7. счетчики
8. блок инициализации
9. критерий остановки
10. методы обработки результатов

1) Состояние системы должно быть определено с достаточной степенью детальности, необходимо и достаточно для вероятностного продолжения моделирования(пример: в системе массового обслуживания состояние задается текущим числом заявок в ней, фазами текущего обслуживания в ней и моментами наступающих событий каждого вида).

2) Под событием модели понимается скачкообразные изменения ее состояния; оно может быть первичным (прибытие заявки или завершение обслуживания) и вторичным (прием заявки на обслуживание, продвижение очереди), которое наступает вследствие первичного.

3) С помощью датчика случайных чисел в модели формируются ее очередные состояния, случайные числа генерируются в соответствии с заданным распределением.

4) Имитационный процесс развивается в модельном (системном) времени, счетчик модельного времени называется таймером.

5) Логика модели реализуется в процессе обработки цепей событий. В цепи текущих событий находятся события, которые наступают в один момент модельного времени (пример: уход из системы обслуживания заявки, выборка на обслуживание из очереди головной заявки). В цепи будущих событий находятся события запланированные посредством датчика случайных чисел на последующие моменты системного времени (пример: завершение обслуживания, прибытие очередной заявки, поломка обслуживающего устройства, уход из очереди заявок). В цепи задержанных событий находятся события, развитие которых заблокировано.

6) Цель моделирования - определение показателей качества функционирования системы.

8) Под инициализацией понимается поведение модели до начала прогона в исходное состояние (существует некоторый период установки).

9) критерий остановки определяет момент прекращения прогона модели (счетчики числа обслуживаемых заявок, правильное управление прогоном по достижении точности одного из показателей).

10) обработка результатов модели состоит в сжатии полученной информации, вычисление статистических оценок точечных и интервальных, оценки статистической значимости различных параметров, построение гистограмм и статистических функций распределения.

2.2 Объекты GPSS.

2.2.1 Классы сущностей(entities).

Система	Сущности	Атрибуты	Активности
Авиация	Автопилот Элерон Корпус	Ошибка Угол Курс	Сигнал Тор- можение По- ворот
Завод	Детали Станки Работники	Кол-во Производи- тельность Умение	Заказ Об- работка Планирова- ние

2.2.2 Стандартные числовые атрибуты.

Позволяют отразить зависимость поведения транзактов от состояния системы.

V1, V2, ...

BV1, BV2, ...

2.2.3 Транзакты(заявки, запросы сообщения).

Движущийся транзакт называется активным. Логический транзакт представляется записью и находится в списках определяющих текущее положение и статус каждого из них. Транзакты могут находиться в одном блоке, одновременно в нескольких или ждать возможности занять блок.

2.2.4 Сегмент модели.

Каждый i-й сегмент имеет свой блок генерации транзактов.

GENERATE

..... — сегмент

TERMINATE

Все сегменты и блоки имеют один масштаб времени.

2.2.5 Параметры транзактов:

PR - приоритет транзакта (по умолчанию 0)

MARK TIME - значение абсолютного времени при входе транзакта в модель.

DELAY INDICATOR - транзакт не смог перейти к следующему блоку.

TRACE INDICATOR - выдается сообщение о продвижении транзакта.

CURRENT BLOCK - номер блока, где находится транзакт.

NEXT BLOCK - следующий блок в который должен перейти транзакт.

CHAINS - в каких цепях данный транзакт.

2.2.6 Состояния транзакта:

ACTIVE - находящийся в обработке.

SUSPENDED - ожидающий в цепи будущих событий.

PASSIVE - находящийся в состоянии покоя.

TERMINATED - подлежащий удалению из текущей модели.

2.2.7 Цепи транзактов:

Future Events Chain (цепь будущих событий)

Current Events Chain (цепь текущих событий)

Delay Chain (цепь задержек)

Pending Chain (цепь зависания на устройстве)

User Chain

ЦТС - содержат транзакты, которым предстоит пройти хотя бы один блок до увеличения модельного времени, упорядоченные по убыванию приоритета. При исчерпании ЦТС в нее переносятся ближайшие по времени транзакты из ЦБС.

ЦБС упорядочена по времени активации транзактов.

Цепь повторных попыток содержит транзакты, которые не удовлетворяют блокам GATE, TEST, TRANSFER ALL, TRANSFER BOTH.

2.2.8 Цепи устройств:

Каждое устройство имеет четыре цепи транзактов.

Pending - ожидание возможности прервать устройство в режиме прерывания.

Interrupt - цепь прерванных транзактов.

Delay - ожидающих возможного занятия устройства.

Retry - ожидающих готовности.

2.2.9 Устройства и памяти.

Устройство - объект, которым в любой момент времени может владеть только один транзакт.

SEIZE - занять некоторое устройство.

PREEMPT - занимает устройство транзакт с более высоким приоритетом.

STORAGE - память, определяется своей ёмкостью.

ENTER - занять некоторое количество ячеек.

LEAVE - освободить.

Факторы	Устройства	Память
Очередь	Раздельная	Общая
Доступность	Раздельная	Общая
Ресурс	Время	Производительность
Неоднородность	Возможна	Исключено
Идентификатор	Есть	Нет
выхода		
Возможность	Есть	Нет
прерывания		

FAVAIL - доступность устройства.

SAVAIL - доступность памяти.

FUNAVAIL - недоступность.

SUNAVAIL - недоступность.

2.2.10 Ансамбли и группы.

Независимо работы, которая выполняется параллельно, не могут обеспечиваться одним транзактом. Запускающий их транзакт должен расщепляться с последующей сборкой компонентов. Все транзакты в моделях принадлежат к множествам ансамблям. Каждый транзакт принадлежит только одному ансамблю. Транзакт введенный в модель блоком GENERATE получает номер ансамбля и становится первым и единственным членом нового ансамбля. Потомки формируемые при входе в блок SPLIT попадают в тот же ансамбль. Для соединения служит блок ASSEMBLE. После входа в него начинается подсчет членов ансамбля пока количество не достигнет параметра A он их будет считать.

** рисунок **

GATHER - то же самое, что ASSEMBLE, только движение продолжают все транзакты в порядке FIFO(First In First Out).

MATCH A - синхронизировать (A - указывает на сопряженный блок). При входе в один из блоков MATCH и обнаружении в сопряженном члена того же ансамбля, оба транзакта продолжают движение. При отсутствии пары активный транзакт задерживается в цепи парности. Принадлежность транзакта к ансамблю определяется только генетической. Однако имеется возможность их произвольно группировать. Каждый транзакт может принадлежать любому числу групп.

JOIN - присоединить транзакт к группе.

REMOVE - удаляет транзакт из группы.

EXAMINE - определяет принадлежность к группе.

2.2.11 Приоритет транзактов и прерывание.

Механизм приоритета.

Транзакт может иметь целочисленный приоритет, по умолчанию нулевой. При генерации в блоке GENERATE A,B,C,D,E -> задает приоритет. Если параметр не задан, то он нулевой.

PRIORITY - смена приоритета. Запланированный на один момент модельного времени более приоритетный транзакт обрабатывается раньше.

Прерывание обслуживания.

Возможно только для устройств.

PREEMPT A (A - имя устройства) позволяет прервать обработку транзакта на устройстве A, в случае если приоритет у активного транзакта выше, чем у обрабатываемой заявки.

RETURN A - снимает прерывание с устройства A. Прерывающая заявка сама прерываться не может.

2.2.12 Логические ключи.

Используются для управления движением транзакта.

LSj - логический ключ.

Значение 1, если в положении "SET"

0, если в состоянии "RESET"

2.2.13 Переменные и сохраняемые значения.

GPSS сопоставляет с переменными выражения над константами стандартными числовыми атрибутами, функциями, содержащими арифметические и логические операторы.

VARIABLE - целое

BVARIABLE - логическое

FVARIABLE - действительное

Сохраняемые значения.

Транзакты могут ссылаться друг на друга. Их общение реализуется через сохраняемые значения.

SAVEVALUE - скалярная величина

MSAVEVALUE - матрица

2.2.14 Накопление статистики.

Очереди.

Ожидание возникает в модели при блокировке, но измерение длительности ожидания начинается при входе в блок QUEUE(регистратор очереди).

DEPART - забрать транзакт из очереди.

Транзакт может находиться в нескольких очередях QUEUE-QUEUE-SEIZE-DEPART-ADVANCE-RELEASE-DEPART.

Стандартные числовые атрибуты могут использовать не только для статистики но и для управления транзактом.
Гистограмма распределения времени ожидания в очереди должна быть описана таблицей QTABLE.

Таблицы общего назначения.

Статистику можно собрать не только времени ожидания. Интервал перехода от одного блока до другого называется транзитным временем.

MARK - позволяет запомнить системное время, начало отсчета.

TABULATE - окончание момента отсчета. Если блок MARK не стоит, то время с блока GENERATE.

TABLE - описание таблиц.

2.2.15 Трассировка.

TRACE - включает индикатор трассировки. В этом случае при входе транзакта в очередной блок в журнал выдается трассировочное сообщение.

UNTRACE - выключает трассировку.

2.2.16 Генераторы случайных чисел.

Генератор генерирует мультипликативно с частотой $2^{31} - 2$ (исключая ноль)

RNj - выдает значение [0,999]

RN3@24+40 получим [40,63]

3 Описание языка GPSS.

Модель в GPSS определяется последовательностью модельных операторов. Модельным оператором может быть GPSS оператор или процедура на языке PLUS. GPSS оператор - либо команда либо блок.

Каждый GPSS оператор записывается в одной строке до 250 символов.

Оператор составлен из областей, которые разделены пробелом или разделителем

<номер строки> <метка> <действие> <операнды> <;комментарий>

если комментарий с новой строки, то *.

Пример:

```
;GPSS World Sample File
GENERATE 300,100 ;новый клиент
QUEUE BarberQueue ;начало ожидания
SEIZE Barber ;захват или ожидание парикмахера
DEPART BarberQueue ;окончание ожидания
ADVANCE 400,200 ;стрижка
RELEASE Barber ;освобождаем парикмахера
TERMINATE ;клиент уходит
```

3.1 GPSS команды:

BVARIABLE - описание логической переменной.

CLEAR - уничтожает статистику и удаляет все транзакты.

CONDUCT - выполняет процедуру на языке PLUS.

CONTINUE - возобновляет моделирование.

EQU - назначает переменной значение.

EXIT - заканчивает сеанс моделирования.

FUNCTION - определяет функцию.

FVARIABLE - определяет переменную действительного типа.

HALT - останавливает имитацию и удаляет все команды в очереди.

INCLUDE - включить файл в модель.

INITIAL - инициализирует или модифицирует значения логических ключей, сохраняемых значений и матрицу.

INTEGRATE - установить изменение переменной.

MATRIX - определяет матрицу.

QTABLE - определяет таблицу для времени ожидания в очереди.

REPORT - устанавливает имя файла в отчете или запрашивает отчет.

RESET - обнуляет статистику имитации.

RMULT - устанавливает стартовые значения первых семи генераторов случайных чисел.

SHOW - вычисляет и выводит значение выражения.

START - устанавливает значение счетчика завершения и начинает имитацию.

STEP - выполняет конечное число блоков.

STOP - устанавливает условие остановки, указывает номер транзакта и номер блока.

STORAGE - определяет память.

TABLE - определяет таблицу.

VARIABLE - определяет переменную.

3.2 GPSS блоки:

ADOPT - изменение содержимого ансамбля.

ADVANCE - поместить транзакт в цепь будущих событий.

ALTER - проверка и модификация транзакта в группе.

ASSEMBLE - ожидание и уничтожение связанных транзактов.

ASSIGN - изменение параметра транзакта.

BUFFER - поместить транзакт в конец цепи текущих событий.

CLOSE - закрыть поток данных.

COUNT - поместить количество сущностей в параметр транзакта.

DEPART - уменьшение количества элементов в очереди.

DISPLACE - изменение следующего блока для транзакта.

ENTER - занять или ожидать занятия в памяти.

EXAMINE - проверить принадлежность к группе.

EXECUTE - выполнить действие определяемое другим блоком.

FAVAIL - установить устройство "доступно".

FUNAVAIL - установить устройство "не доступно".

GATE - проверка сущностей и изменение направления потока транзактов.

GATHER - ожидать связанных транзактов.

GENERATE - создание транзактов и установка цепи будущих событий.

INDEX - изменение параметров транзакта.

INTEGRATION - включить или выключить интегрирование пользовательских элементов.

JOIN - установить члена в численную или транзактную группу.

LEAVE - освободить память.

LINK - переместить транзакт в цепь пользователя.

LOGIC - установить логический ключ.

LOOP - уменьшает параметр и переходит на блок, если результат не ноль.

MARK - помещает системное время в параметр транзакта.

MATCH - ожидать соответствующего транзакта для достижения соединения.

MSAVEVALUE - установить значение в элемент матрицы.

OPEN - инициализировать поток данных.

PLUS - вычислить PLUS выражение и сохранить результаты параметра.

PREEMPT - заменить владельца устройства.

PRIORITY - изменить приоритет транзакта.

QUEUE - увеличить количество элементов в очереди.

READ - взять очередную строку из потока данных.

RELEASE - освободить устройство.

REMOVE - изъять транзакт из группы числовой или транзактной.

RETURN - освободить устройство транзакта замещающего какой-то транзакт.

SAVEVALUE - установит значение сохраняемой величины.

SAVAIL - установить память "доступно".

SCAN - протестировать группу транзактов и поместить значение в параметры.

SEEK - изменить указатель строки в потоке данных.

SEIZE - установить или ожидать установления владельца устройства.

SELECT - поместить номер выделенного объекта в параметр транзакта.

SPLIT - создать связанный транзакт.

SUNAVAIL - установить память "не доступно".

TABULATE - обновить таблицу.

TERMINATE - уничтожить транзакт и уменьшить счетчик окончания.

TEST - проверить арифметическое условие и изменить поток транзактов.

TRACE - пометить активный транзакт индикатором слежения.

TRANSFER - переместиться на специальный блок.

UNLINK - удалить транзакт из цепи пользователя.

UNTRACE - выключить индикатор слежения для активного транзакта.

WRITE - послать значение в поток данных.

3.3 Стандартные числовые атрибуты(SNA):

Имитация переменных состояний. SNA могут быть числовыми или строковыми, могут использоваться как операнды в выражениях.

ε - объект(\$имя, число, *адрес).

ρ - параметр транзакта.

A1 - множество ансамбля активного транзакта(int).

AC1 - значение системного времени(меняется с последнего выполнения команды CLEAR)(real).

BV ε -результат оценки логической переменной.(BV1,BV5,BV\$LogKey).

C1 - значение относительного системного времени(меняется с последнего выполнения команды RESET)(real).

CA_{ε} - среднее по времени количество транзактов в цепи пользователя(real).
 CC_{ε} - общее количество транзактов в цепи пользователя(int).
 CH_{ε} - текущее количество транзактов в цепи пользователя(int).
 CM_{ε} - максимальное количество транзактов в цепи пользователя(int).
 CT_{ε} - среднее время транзактов в цепи пользователя(real).
 F_{ε} - устройство занято(1-если занято, 0-нет)(int).
 FC_{ε} - количество транзактов пытающихся занять устройство или захватить(int).
 FE_{ε} - устройство захвачено блоком PREEMPT(1-да, 0-нет).
 FN_{ε} - результат вычисления функции(real).
 FR_{ε} - занятость устройства(доля времени $x/1000$ от 0 до 1000)(real).
 FT_{ε} - среднее время за которое захватывается устройство(real).
 FV_{ε} - устройство доступно.
 GN_{ε} - количество членов в числовой группе(int).
 GT_{ε} - количество членов транзакционной группы(int).
 LS_{ε} - логический ключ установлен.
 MB_{ε} - транзакт ждут в блоке MATCH.
 MP_{ρ} - транзактное время минус текущее время минус ρ .
 $MX_{\varepsilon}(m,n)$ - значение элемента матрицы ε в строке m и столбце n , в матрице размерности больше 2 остальные индексы равны 1.
 $M1$ - транзитное время равное абсолютному системному времени минус MARK-время транзакта.
 P_{ρ} - значение параметра активного транзакта(int, real, string).
 PR - приоритет транзакта(int).
 Q_{ε} - количество элементов в очереди.
 QA_{ε} - среднее по времени количество элементов в очереди(real).
 QC_{ε} - количество всех входящих в очередь элементов(int).
 QM_{ε} - максимальное количество элементов в очереди(real).
 QT_{ε} - среднее время ожидания в очереди(real).
 QX_{ε} - среднее время, исключая нулевое время ожидания(real).
 QZ_{ε} - количество элементов с времени ожидания 0(int).
 R_{ε} - доступный объем памяти(int).
 RN_{ε} - случайное число от 0 до 999, где ε -номер случайного генератора(int).
 S_{ε} - используемый объем памяти(int).
 SA_{ε} - среднее по времени объем памяти используемый(real).
 SC_{ε} - общее количество памяти.
 SE_{ε} - память пуста(1-да, 0-нет).
 SF_{ε} - память полна(1-да, 0-нет).
 SR_{ε} - занятость памяти (доля времени $x/1000$ от 0 до 1000)(real).
 SM_{ε} - максимальный объем используемой памяти.
 ST_{ε} - среднее время занятости памяти на одну единицу памяти.
 SV_{ε} - память доступна(1-да, 0-нет).
 TB_{ε} - не взвешенное среднее элементов таблицы.
 TC_{ε} - количество элементов в таблице(int).

$TD\varepsilon$ - среднее квадратичное отклонение элементов в таблице(real).
 $TG1$ - количество оставшихся TERMINATE блоков до завершения(int).
 $V\varepsilon$ - результат вычисления действительной переменной(real).
 $W\varepsilon$ - количество транзактов в блоке ε (int).
 $X\varepsilon$ - значение сохраняемой величины ε (int, real, string).
 $XN1$ - количество активных транзактов.
 $Z1$ - объем оперативной памяти системы.

Приоритет	Операторы, используемые в выражениях
8	\wedge - степень: $\rightarrow A^B$
7	$\#$ (или $*$) - умножение: $A * B \rightarrow A \cdot B$
7	$/$ деление: A/B
7	\backslash - целочисленное деление: $A \backslash B \rightarrow \lfloor A/B \rfloor$
6	$@$ - остаток от деления: $A @ B \rightarrow A \bmod B$
5	- вычитание
5	+ сложение
4	\geq (GE) (1-да, 0-нет)
4	\leq (LE) (1-да, 0-нет)
4	$> G$
4	$< L$
3	$= E$
3	$\neq NE$
2	$\&$ AND
1	$ $ OR

4 Язык PLUS.

4.1 Оисание процедур.

Описание процедуры-функции идет до или после блока.

```
PROCEDURE <имя>(<список элементов>)  
BEGIN  
END
```

4.1.1 Локальные переменные.

```
TEMPORARY <список скалярных переменных>  
TEMPORARY MATRIX <имя>(<список верхних границ>)(максимальная  
размерность - 6)  
RETURN <выражение>
```

4.1.2 Вызов процедур.

```
<имя процедуры>(<список аргументов>)
```

4.2 Операторы языка PLUS.

```
/*.....*/ - комментарии.  
<имя> = <выражение>; - присвоение.  
<метка>:  
GOTO <метка>;  
IF (<выражение>) THEN <оператор>;  
ELSE <оператор>;  
WHILE (<выражение>) DO <оператор>;  
INCLUDE "(специализация файла)"
```

4.3 Библиотечные функции.

t_n - номер транзакта.

```
Query XN Exist ( $t_n$ ) - 1, если транзакт существует.  
Query XN Parametr ( $t_n$ , par) - значение параметра.  
Query XN Assembly ( $t_n$ ) - номер ансамбля.  
Query XN Priority ( $t_n$ ) - приоритет транзакта.  
Query XN M1 ( $t_n$ ) - отметка времени транзакта.
```

4.3.1 Строковые процедуры.

Length (Str) - длина строки.

Trim (Str) - обрезаает начальные и хвостовые пробелы.

DataType (<данные>) - выдает строку или аргумент.

Value (Str) - числовой эквивалент строки.

String (<данные>) - преобразует данные в эквивалентную строку.

String Compare (Str1,Str2) - сравнение двух строк по ASCII кодам(Str1<Str2 - (-1), Str2=Str2 - (0), Str1>Str2 - (1)).
 Lower Case (Str) - в нижний регистр.
 Upper Case (Str) - в верхний регистр.
 Left/Right (Str, MaxCount) - возвращает подстроку длины MaxCount слева/справа.
 Find (SubStr, Str) - возвращает позицию SubStr в Str.
 SubString (Str, Offset, MaxCount) - возвращает подстроку длины MaxCount с позиции Offset из Str.
 Word (Str, WNumber) - возвращает слово с номером WNumber.
 Align/Plase (InsStr, Str, Offset) - замена подстроки InsStr с правой/левой позиции Offset в Str.
 Catenate (Str1, Str2) - склеивание двух строк.
 PolyCatenate (Str1, Str2, ... ,StrN) - склеивание N строк.
 Copies (Str, Count) - склеивает Count копий строки Str.

4.3.2 Математические функции:

ABS(A) - абсолютное значение.
 ATN(A) - арктангенс.
 COS(A) - косинус.
 EXP(A) - экспонента.
 INT(A) - целая часть.
 LOG(A) - натуральный логарифм.
 SIN(A) - синус.
 SQR(A) - квадратный корень.
 TAN(A) - тангенс.

4.4 Распределение случайной величины.

4.4.1 Таблица. Распределения.

Название распределения	GPSS	Комментарии
1. Бета - распределение	BETA(A,B,C,D,E)	A - генератор СЧ, [B,C] - интервал генерируемых значений, D,E-1e,2e определенное значение
2. Биномиальное	BINOMINAL(A,B,C)	B - количество независимых испытаний Бернулли > 0 , C - вероятность успешного испытания[0,1]
3. Дискретно - равномерное	DUNIFORM(A,B,C)	[B,C] - интервал генерируемых значений
4. Экспоненциальное	EXPONENTIAL	B - величина сдвига, C - величина для сжатия распределения, B+C - среднее значение
5. Экстремальные значения типа A	EXTVALA(A,B,C)	— —
6. Экстремальные значения типа B	EXTVALB(A,B,C)	— —
7. Геометрическое	GEOMETRIC(A,B)	B - вероятность успешных испытаний [0,1]
8. Гамма	GAMMA(A,B,C,D)	B - сдвиг, C - сжатие, D - параметр формы
9. Обратное распределение Гаусса	INVGAUSS(A,B,C,D)	— —
10. Обратное распределение Вейбулла	INVWEIBULL(A,B,C,D)	— —
11. Распределение Лапласа	LAPLACE(A,B,C,D)	— —
12. ЛогЛапласово распределение	LOGLAPLACE(A,B,C,D)	— —
13. Отрицательное биномиальное	NEGBINOMIAL(A,B,C,D)	B - количество успешных испытаний, после возвращается число безуспешных испытаний, C - вероятность успеха
14. Логистическое	LOGISTIC(A,B,C)	B - сдвиг, C - сжатие
15. Логлогистическое	LOGLOGISTIC(A,B,C,D)	D - форма
16. Логнормальное	LOGNORMAL(A,B,C,D)	— —
17. Парето	PARETO(A,B,C)	— —
18. Пирсона типа 5	PEARSON5(A,B,C,D)	— —
19. Нормальное	NORMAL(A,B,C)	B - среднее значение, C - стандартное отклонение
20. Пуассона	POISSON(A,B)	B - среднее значение
21. Пирсона типа 6	PEARSON6(A,B,C,D,E)	[B,C] - интервал, D,E - параметры
22. Треугольное	TRIANGULAR(A,B,C,D)	[B,C] - интервал, D - мода, D \in [B,C]
23. Равномерное	UNIFORM(A,B,C)	[B,C] - интервал
24. Вейбулла	WEIBULL(A,B,C,D)	B - сдвиг, C - сжатие, D - форма

4.4.2 Формирование случайных чисел для дискретных неравномерных распределений.

Таблица распределения.

Значение СВ	Относительная частота	Сумма частот	Диапазон	Интервал
2	0,15	0,15	0,0..0,15	1
5	0,2	0,35	0,15..0,35	2
8	0,25	0,6	0,35..0,6	3
9	0,22	0,82	0,6..0,82	4
12	0,18	1,0	0,82..1,0	5

$[0,1] \rightarrow 0,523664 \rightarrow N3 \rightarrow \text{Случайное Число} = 8$

Rasp FUNCTION RN6, D5(D - дискретное, 5 - число отрезков)
.15,2/.35,5/.6,8/.82,9/1,12

TRANSFER FN, Rasp ;обращение к распределению;
GENERATE FN\$Rasp ;в блоке;

*****график*****

4.4.3 Формирование случайных чисел для непрерывных распределений.

RaspN FUNCTION RN6, C5 (C - непрерывное, 5 - число отрезков)
.1,14.2/.35,30.4/.6,50/.85,64.7/1,80

$$Y(x) = Y(x_i) + \frac{Y(x_{i+1}) - Y(x_i)}{x_{i+1} - x_i} \cdot (x - x_i) \quad x \in [x_i; x_{i+1}]$$

*****график*****

4.4.4 Пример описания экспоненциального распределения.

Пуассоновский поток.

$$P_k(T) = \frac{(\lambda T)^k}{k!} \cdot e^{-\lambda T}, k = 0, 1, \dots$$

$P_k(T)$ - вероятность, что k аявок придет за время T.

λ - средняя интенсивность поступления заявок.

$$T_p = T_{cp} \cdot [-LN \cdot (1 - RN_n)]$$

T_p - интервал между заявками.

T_{cp} - средний интервал.

RN_n - равномерно распределенное случайное число в $[0,1]$

4.4.5 Пример на GPSS: Аппроксимация функции T_p на 23 отрезках.

```

XpDisFUNCTIONRN200,C24
0,0/.1,.104/.2,.222/...../.995,5.3/.998,6.2/.999,7/.9998,8
[-LN(1 - RNn)] [0, + ∞]
RNn = 0.999999 13.815
> 0.999999 - 0.0199%
*****рис*****
GENERATE 360, FN$ XpDis

```

$f(t) = \mu \cdot e^{-\mu t}$ - плотность распределения.
 $f_i = \frac{i!}{\mu^i} i = 1, 2, \dots$
 $\bar{F} = e^{-\mu \cdot t}$
 $\tilde{f}(t|\tau) = \frac{f(t+\tau)}{\bar{F}(\tau)} = \frac{\mu \cdot e^{-(t+\tau)}}{e^{-\mu \cdot \tau}} = \mu \cdot e^{-\mu t}$

4.5 Распределение фазового типа.

Порождается системой подлежащих прохождению фаз обслуживания с показательно распределенной длительностью.

4.5.1 Распределение Эрланга

Распределение r -го порядка.

$$f(t) = \frac{\mu \cdot (\mu t)^{r-1}}{(r-1)!} e^{-\mu t}$$

$$\bar{F}(t) = \sum_{i=0}^{r-1} \frac{(\mu t)^i}{i!} e^{-\mu t}$$

$$f_k = \frac{r(r-1) \dots (r+k-1)}{\mu^k}, k = 1, 2, \dots$$

$$D = \frac{r}{\mu^2}$$

1. Найти $\tilde{r} = [\frac{f_i^2}{D} + 0, 5]$
2. Вычислить $\mu = \frac{\tilde{r}}{f_i}$

4.5.2 Гиперэкспоненциальное распределение

y_i - вероятность выбора i -й фазы,

μ_i - значение параметра распределения i -й фазы.

4.5.3 Распределение Кокса

Каждая i-я фаза обслуживания подчиненная распределению Кокса требует продвижения в вероятностью y_i и завершения с вероятностью $\bar{y}_i = 1 - y_i$.

$f_k = (1 - y)\alpha^k + y(a + b)^k$, a, b – моменты для первой и второй фазы обслуживания.

4.5.4 Гиперэрланговское распределение

4.5.5 Распределение Ньютона

Длительность любой реализации процесса здесь соответствует случайному времени блуждания заявки по изображенной сети с показательным распределением с параметром $\{\nu_i\}, i = \overline{1, m}$, с задержкой в узлах и одним поглощающим состоянием.

4.6 Преобразование распределений

Смеси распределений.

$$f(x) = \sum_{i=1}^{\infty} P\{Y = i\} f_i(x) - \text{дискретное}$$

$$f(x) = \int f_y(x)g(y)dy - \text{непрерывное}$$

Отношение.

$$f_1(x) = \lambda \cdot e^{-\lambda \cdot x}, f_2(y) = \mu \cdot e^{-\mu \cdot y} \quad Z = \frac{X}{Y}, X = Y \cdot Z$$

$$F(z) := PZ < z = \int_0^{\infty} (1 - e^{-\lambda yz}) \cdot \mu \cdot e^{-\mu \cdot y} dy$$

$$f(z) = \int_0^{\infty} \lambda y \mu e^{-\mu y} dy = \lambda y \int_0^{\infty} y e^{-y(\lambda + \mu)} dy = \frac{\lambda \mu}{(\mu + \lambda z)^2}$$

Свертка.

Приводит к суммированию средних и дисперсий.

$x_1 \dots x_N$ - имеют функцию распределения $F(x)$

Максимум. $F^n(x)$

Минимум. $1 - [1 - F(x)]^n$

Дополнительная функция распределения.

$$\bar{F}(x) = 1 - F(x)$$

$$\text{Эрланга. } \bar{F}(t) = \sum_{i=0}^{r-1} \frac{(\mu t)^i}{i!} \cdot e^{-\mu t}$$

$$\text{Гиперэкспоненциальное. } \bar{F}(t) = \sum_{i=0}^n y_i \cdot e^{-\mu_i t}$$

$$\text{Гиперэрланговое. } \bar{F}(t) = \sum_{i=0}^{r-1} \frac{(\mu t)^i}{i!} \cdot e^{-\mu t} + y \cdot \frac{(\mu t)^r}{r!} \cdot e^{-\mu t}$$

5 Примеры моделей систем.

5.1 Многоканальные системы

```
Chans STORAGE 3
    GENERATE (EXPONENTIAL(1,0,1))
    QUEUE Wait
    ENTER Chans,1
    DEPART Wait
    ADVANCE 27
    LEAVE Chans,1
    TERMINATE

    GENERATE 5000,,1
    TERMINATE 1
```

5.2 Приоритетная система обслуживания

```
Work VARIABLE (GAMMA(2,0,P2(3,3)))
    GENERATE (EXPONENTIAL(1,0,5.0)),,,3
    ASSIGN 1,1 ;тип транзакта
    ASSIGN 2,0.3 ;время обслуживания
```

```
Common QUEUE Sque
    PREEMPT Chan, PR, Rept, 100
    SAVEVALUE Time, V$Work
    ASSIGN 5, X$Time
```

```
Wait ADVANCE P5
    TRANSFER ,Out
```

```
Rept ASSIGN 5, V$Work
    TRANSFER ,Wait
```

```
Out DEPART Sque
    RETURN Chan
    TERMINATE
```

```
    GENERATE (EXPONENTIAL(1,0,3.33)),,,2
    ASSIGN 1,2
    ASSIGN 2,0.6
    TRANSFER ,Common
```

```
    GENERATE (EXPONENTIAL(1,0,2.50)),,,1
    ASSIGN 1,3
```

```

ASSIGN 2,0.9
TRANSFER ,Common

```

```

GENERATE 5000,,1
TERMINATE 1

```

5.3 Квантование времени

Заявка обрабатывается не больше, чем квант времени q . Обслуженная убирается из потока, недообслуженная в конец очереди на дообслуживание.

```

Intro SEIZE Fcty
      TEST G F5, X$Quantum, Last
      ASSIGN 5, (P5 - X$Quantum)
      ADVANCE X$Quantum
      RELEASE Fcty
      TRANSFER ,Intro

```

```

Last ADVANCE P5
      RELEASE Fcty
      .....

```

5.4 Круговое обслуживание

*****рисунок*****

- 1) Обслуживание очереди по одной заявке из каждой непустой рчереди.
- 2) До исчерпания заявок в текущей очереди, включая те, которые пришли в момент обслуживания.
- 3) Обслуживание всех заявок, накопившихся к началу обслуживания.

Пример модели 2:

```

INITIAL $Empty, 1

```

```

GENERATE (EXPONENTIAL(1,0,1))
ASSIGN 1, AC1
ASSIGN 1, 1
ASSIGN 3, 0.25
TEST E X$Empty, 1, Inq1
SAVEVALUE Pointer, P2
SAVEVALUE Empty, 0

```

```

Inq1 QUEUE *2
      TEST E X$Pointer, P2
      DEPART *2
      TRANSFER ,Server

```

*** ;повтор предыдущего блока с параметром 2:

```

GENERATE (EXPONENTIAL(2,0,1))

```

```

      _____
      ||
      ASSIGN 2, 2
      _____
      ||
      TEST E X$Empty, 1, Inq2
      _____
      ||
Inq2  _____
*** ;повтор предыдущего блока с параметром 3.

```

```

Server SEIZE Fcty
      ADVANCE P3
      RELEASE Fcty
      TEST E Q*2, 0, 0, Zzz
      ASSIGN 4,3
      ASSIGN 5,1
      ASSIGN 6,1
      SAVEVALUE Qmax,0

Check TEST GE Q*5, X$Qmax, Next
      SAVEVALUE Qmax, Q*5
      ASSIGN 6,P5

Next ASSIGN 5+, 1
      LOOP 4, Check

Out TEST E P6, 0, Zpt
      SAVEVALUE Empty, 1
      TRANSFER ,Zzz

Zpt SAVEVALUE Pointer, P6

Zzz TERMINATE 1

```

5.5 Учет недельного и суточного циклов.

8 часов в день, 7 часов в пятницу.

```

*** ;рабочий цикл
      GENERATE 1,0.3
      QUEUE QQQ
      SEIZE FFF
      DEPART QQQ
      ADVANCE 0.8,0.5
      RELEASE FFF
      TERMINATE 1

*** ;ежедневные перерывы
      GENERATE 24,0.8
      FUNAVAIL FFF
      ADVANCE 16

```



```

TEST E LS2,0,Out
FAVAIL FFF
Out TERMINATE
*** ;выходные дни
GENERATE 168,0,103
LOGIC $2
FUNAVAIL FFF
ADVANCE 65
FAVAIL FFF
LOGIC R 2
TERMINATE

```

6 Эксперименты над моделями.

В процессе имитационного эксперимента над моделью с учетом случайных исходов каждого прогона необходимо:

- 1) Классифицировать факторы на существенные и не существенные.
 - 2) Разделить и оценить количественное влияние факторов и их комбинаций на целевую функцию.
 - 3) Найти наивыгоднейшую комбинацию факторов.
- 1) Дисперсионный анализ.
 - 2) Регрессионный анализ и факторный.
 - 3) Методы статистической оптимизации.

Понятия

Эксперимент начинается с выбора одной или нескольких **целевых функций**, которые количественно оценивают полезность системы.

Пример:

Типичная задача систем массового обслуживания: обеспечение минимального времени реакции системы при ограниченных затратах или минимизация затрат при среднем времени реакции не более заданного.

Более сложно: обеспечить обслуживание максимального процента заявок за время не более заданного или минимизации времени, за которое обслуживается заданный процент заявок.

Измерение значения целевой функции называется **наблюдением**, а в всё их множество - **результатом**.

Основная задача: - изучить влияние факторов на результат.

Каждое повторение моделирование - **прогон**.

Назначим значение(уровни) факторов, которые будут определять условия каждого прогона. Если влияние некоторого фактора зависит от значения другого и не сводится к сумме влияний, то говорят о **взаимодействии этих факторов**. Комбинация уровней определяет **условие прогона**. Когда условия прогона неизменны, мы называем эти прогоны **репликами**.

Результаты в совокупности заполняют матрицу результатов. Протяженность матрицы по измерению должна быть равна числу уровней этого

фактора. Каждая размерность матрицы результатов сопоставляется одному из факторов. Если используется несколько целевых функций, то столько же и в матрице.

$$M = \begin{matrix} & \begin{matrix} \Phi_1 \\ \Phi_2 \\ \dots \\ \Phi_n \end{matrix} & \begin{pmatrix} m_{11} & m_{12} & \dots & \dots & m_{1n} \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & m_{nk} \end{pmatrix} \end{matrix}$$