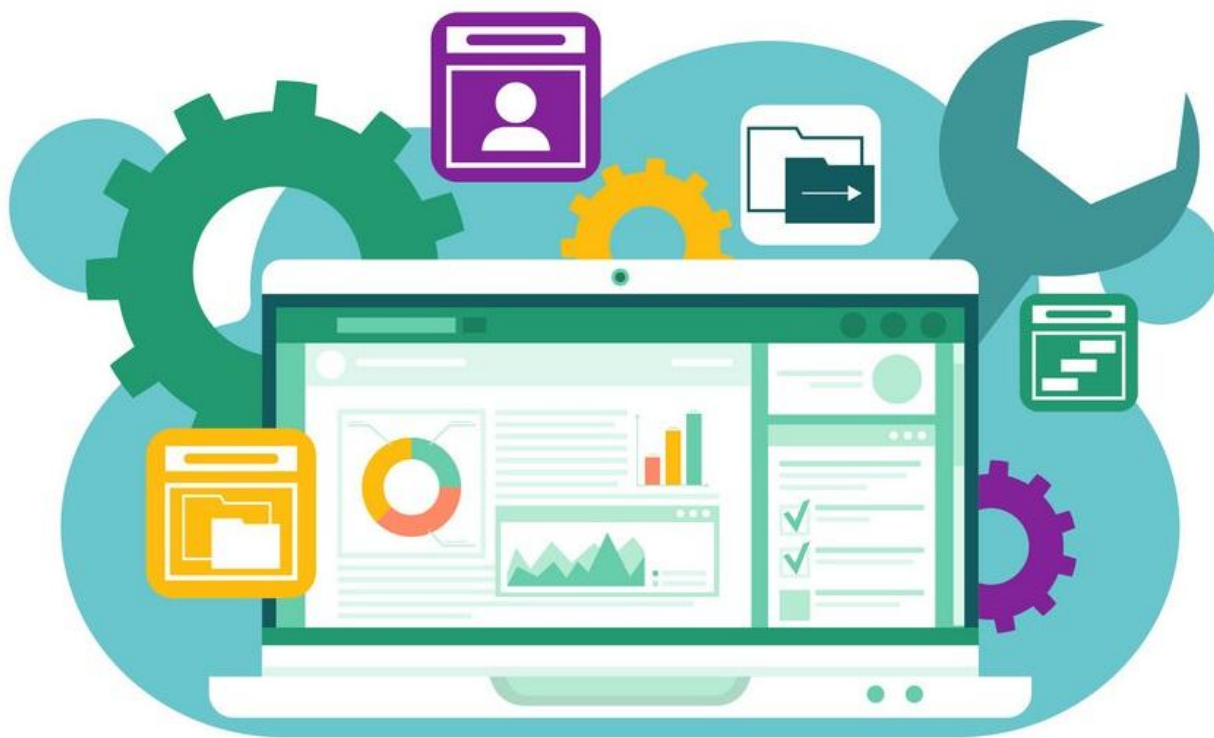


А.А. Алтаев, Э.Ц. Доржиева, И.В. Старинский, И.С. Тулохонова



КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Улан-Удэ

2024

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Восточно-Сибирский государственный университет технологий и управления»
(ВСГУТУ)

А.А. Алтаев, Э.Ц. Доржиева, И.В. Старинский, И.С. Тулохонова

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Учебное пособие

*Рекомендовано Учебно-методическим советом ВСГУТУ в качестве учебного пособия
для обучающихся по программам высшего образования
02.03.03 «Математическое обеспечение и администрирование информационных систем»;
01.03.04 «Прикладная математика»*

Улан-Удэ
Издательство ВСГУТУ
2024

УДК 519.876.5 (075.8)
ББК 32.973я73
К637

Печатается по решению редакционно-издательского совета Восточно-Сибирского государственного университета технологий и управления.

Рецензенты:

Т.Г. Дармаев, канд. физ.-мат. наук, доц. БГУ им. Д. Банзарова
Л.П. Бильгаева, канд. техн. наук, доц. ВСГУТУ

Алтаев А.А., Доржиева Э.Ц., Старинский И.В., Тулохонова И.С.

К637 Компьютерное моделирование: учеб. пособие. – Улан-Удэ: Изд-во ВСГУТУ, 2024. – 84 с.
ISBN 978-5-907942-01-1

Учебное пособие включает краткую теорию моделирования систем и указания по выполнению лабораторных заданий. Предназначено студентам направлений подготовки 01.03.04 «Прикладная математика», 02.03.03 «Математическое обеспечение и администрирование информационных систем».

ISBN 978-5-907942-01-1

ББК 32.973я73
© А.А. Алтаев с соавт., 2024
© ВСГУТУ, 2024

Содержание

Введение	5
Введение в компьютерное моделирование	5
Общая классификация моделей	5
Преимущества и ограничения моделей.....	6
Модели, используемые в практикуме	6
Дискретно-детерминированные модели	7
Пример моделирования автомата	10
Контрольная работа	11
Дискретно-стохастические модели	11
Реализация вероятностных переходов	11
Реализация функции выхода в стохастическом режиме	13
Контрольная работа	14
Непрерывно-стохастические модели	14
Потоки событий.....	14
Основные элементы СМО	16
Канал	16
Очередь	16
Вентиль	17
Транзакт	18
Источник	19
Типы событий.....	20
Расчет времени наступления основного события	21
Алгоритм изменения модельного времени.....	24
Моделирование простейшей СМО	24
Контрольная работа	26
Условные обозначения типов СМО	26
Моделирование с ЯИМ GPSS	27
Стандартные числовые атрибуты (СЧА)	28
Транзакты.....	28
Списки транзактов.....	29
Блоки.....	29
Блок GENERATE	29
Блок ADVANCE.....	30
Блоки SEIZE и RELEASE.....	30
Блоки QUEUE и DEPART	30
Блок TERMINATE. Счетчик TG1	31
Реализация модели СМО в GPSS	32
Проведение экспериментов с моделью model01.gps	33
Пример завершения моделирования по числу обработанных транзактов	33
Проведение экспериментов с моделью model02.gps	33
Памяти.....	34
Логические переключатели.....	34
Блоки, изменяющие характеристики транзактов.....	35
Блок TRANSFER	35
Блок TEST.....	36
Блок GATE.....	36
Блок LOOP.....	37
Блок SAVEVALUE	37
Примеры моделей с разветвлением маршрутов.....	37

Проведение экспериментов с моделью model03.gps и обработка результатов эксперимента	38
Проведение экспериментов с моделью model04.gps	41
Модель СМО с обслуживанием транзактов, имеющих относительные приоритеты ..	42
Проведение экспериментов с моделью model05.gps	43
Организация прерываний работу ОКУ	43
Моделирование СМО с прерываниями	43
Проведение экспериментов с моделью model06.gps	45
Таблицы	46
Проведение экспериментов с моделью model07.gps	48
Обеспечение минимального значения времени при использовании функции EXPONENTIAL	48
Ансамбли транзактов	50
Проведение экспериментов с моделью model08.gps	51
Проведение экспериментов с моделью model09.gps	53
Карты описания	53
Карты INITIAL и EQU	53
Переменные	53
Функции	54
Примеры моделей с применением функции и переменной	56
Проведение экспериментов с моделью model10.gps	57
Применение дискретно заданных функций	57
Проведение экспериментов с моделью model11.gps	60
Блоки, предназначенные для работы с нумерованными объектами	60
Блок COUNT	60
Блок SELECT	62
Моделирование СМО с нумерованными объектами	62
Проведение экспериментов с моделью model12.gps	64
Повторение эксперимента с измененными условиями	64
Проведение экспериментов с моделью model13.gps	65
Косвенная адресация	65
Проведение экспериментов с моделью model14.gps	67
Пример применения модели с нумерованными очередями	67
Проведение экспериментов с моделью model15.gps	69
Варианты заданий	69
Заключение	76
Рекомендуемая литература	77
ПРИЛОЖЕНИЯ	78
Приложение 1	78
Приложение 2	79
Приложение 3	81
Приложение 4	82

ВВЕДЕНИЕ

Дисциплина «Компьютерное моделирование», читаемая студентам направлений подготовки 01.03.04 «Прикладная математика», 02.03.03 «Математическое обеспечение и администрирование информационных систем», предусматривает выполнение лабораторных заданий. Данное пособие предназначено для методического сопровождения указанных работ и является обновленным переизданием практикума [2]. Пособие содержит краткую теорию по моделированию систем, задания по лабораторным работам с примерами их выполнения.

Пособие также может быть полезно студентам среднего профессионального образования направления 09.02.07 «Информационные системы и программирование, изучающими одноименную дисциплину».

Первая работа состоит в построении и изучении работы конечного автомата, во второй рассматривается вероятностный автомат. В третьей работе изучается поведение простейшей системы массового обслуживания. В четвертой работе строятся модели с применением языка имитационного моделирования (ЯИМ) GPSS и дана методика проведения и обработки экспериментов.

В приложении 4 дана сводка принятых в пособии сокращений.

ВВЕДЕНИЕ В КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Моделирование является одним из инструментов научного познания окружающего мира. Модели могут использоваться при изучении поведения какого-либо реального природного явления или при проектировании физических объектов, таких как самолет или, например, плотина. При создании модели предварительно у изучаемого объекта или процесса выделяются наиболее важные его характеристики. Отобранные характеристики затем используются при построении самой модели. Чем больше характеристик будет учтено, тем точнее и сложнее будет построенная модель.

Модели позволяют представить то, что трудно увидеть или понять. Модели могут помочь ученым передавать свои идеи, понимать процессы и делать прогнозы. Модели используют знакомые объекты для представления незнакомых вещей. В таблице 1 приведены примеры того, что могут представлять модели.

Таблица 1

Модели могут представлять ...	Пример
объекты, которые слишком малы, чтобы их видеть	модель атома или клетки
объекты, которые слишком велики, чтобы их увидеть	модель планеты
уже не существующие объекты	модель динозавра
объекты, которые еще не изобретены	прототип робота
события, происходящие слишком медленно, чтобы их заметить	модель горообразования
события, происходящие слишком быстро	модель предсказания землетрясения
события, которые еще не произошли	модель прогноза погоды

Общая классификация моделей

Используются три типа моделей:

- 1) физические модели;
- 2) компьютерные (математические) модели;
- 3) концептуальные модели.

Модели, которые можно увидеть и потрогать, называются физическими моделями. Примеры таких моделей: модель Солнечной системы или земного шара, модель туловища человека.

Компьютерная модель состоит из математических уравнений и данных. Простые математические модели позволяют рассчитать такие вещи, как, например, сколько километров в час проедет автомобиль. Другие математические модели настолько сложны, что для их создания требуется программное обеспечение. Компьютерные модели помогают смоделировать события, которые занимают много времени, такие как движение тектонических плит. Они также полезны при моделировании событий, которые происходят слишком быстро, чтобы их увидеть, как, например, предсказание землетрясений.

Концептуальные модели используют сравнение с известными вещами, чтобы помочь проиллюстрировать или объяснить идею. Система классификации, используемая для классификации живых существ, является примером такой концептуальной модели. В таких системах классификации ученые, например, группируют организмы по степени их сходства. Другим примером концептуальной модели является атомарная модель Бора, в которой электроны вращаются вокруг ядра атома аналогично тому, как планеты вращаются вокруг Солнца.

Преимущества и ограничения моделей

Научные модели дают следующие преимущества:

- Использование при коммуникации. Модели применяются для передачи результатов наблюдений или идей другим людям. Модели помогают людям визуализировать свои идеи или абстрактные понятия.

- Представление громадных или, напротив, миниатюрных объектов в виде натуральных моделей.

- Модели помогают получать новые научные знания. Модели могут быть использованы для иллюстрации и объяснения научных теорий. Модели могут помочь поддержать научные теории или доказать их ошибочность. Модели могут быть изменены или заменены по мере получения новой информации. По мере развития технологий ученые могут пересматривать и обновлять модели.

- Модели экономят время, деньги и жизни. Работа и тестирование с моделями может быть безопаснее, быстрее и дешевле, чем при использовании реального объекта.

- Модели используются для составления и тестирования прогнозов. Например, инженеры могут использовать модели для предсказания того, как будут работать их изобретения.

Модели наряду со своими достоинствами имеют и ограничения:

- Детализация – модели не учитывают все особенности объектов, которые они представляют. Например, карты не могут включать все детали особенностей Земли, таких как горы, долины и т. д.

- Большинство моделей допускают некоторые приближения для упрощения описания того, что происходит в природе. Такие условности могут приводить к неточным предсказаниям.

-

Модели, используемые в практикуме

В данном пособии рассматриваются математические модели. Существует несколько подходов к классификации таких моделей:

1. С учетом фактора времени – статические и динамические модели. В первом случае характеристики модели не зависят от времени, во втором – являются функциями от времени.

2. По влиянию случайных воздействий – детерминированные и стохастические модели. В детерминированных моделях роль случайного фактора сведена к нулю, в стохастических же, напротив, он учитывается.

3. По типу изменения состояния – непрерывные и дискретные модели. У моделей первого типа их состояние изменяется непрерывно во времени, для второго – переходы из одного состояния в другое считаются мгновенными.

4. По подходу к способу проведения исследования – аналитические и имитационные модели. Первый вариант предусматривает описание модели в виде дифференциальных или интегро-дифференциальных уравнений. Имитационное моделирование применяется в тех случаях, когда отсутствует возможность получения аналитических зависимостей, которые бы описывали реальную систему.

5. Оптимизационные модели, применяемые при решении экономических задач (получение максимальной прибыли при наличии ограничений на используемые ресурсы, моделирование экономического роста и т. д.).

Для изучения были выбраны дискретно-детерминированные, дискретно-стохастические и непрерывно-стохастические модели.

ДИСКРЕТНО-ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ

Такие модели называются конечными автоматами или F-схемами (finite-state machine, FSM). Название обусловлено тем, что для описания этой модели используют три конечных множества: входных сигналов, выходных сигналов и состояний автомата. Дополняют указанные множества две функции: переходов и выхода. Следовательно, конечный автомат можно описать в виде $M = [S, I, O, f_s, f_o]$, где:

- 1) S – множество состояний автомата (State, состояние);
- 2) I – множество входных символов, или входной алфавит (Input, вход);
- 3) O – множество выходных символов, выходной алфавит (Output, выход);
- 4) f_s – функция переходов, $f_s: I \times S \rightarrow S$;
- 5) f_o – функция выхода, $f_o: I \times S \rightarrow O$ или $S \rightarrow O$.

Необходимое условие: перед началом моделирования автомат должен быть всегда инициализирован в исходное состояние s_0 .

Функция f_s является функцией следующего состояния. Она позволяет перевести автомат в новое состояние, используя пару (*state*, *input*). Обозначим текущий такт в виде t_i , а следующий такт в виде t_{i+1} . Тогда, состояние автомата для следующего такта $state(t_{i+1})$ будет получено путем применения f_s к состоянию автомата $state(t_i)$ и входному символу $input(t_i)$:

$$state(t_{i+1}) = f_s(input(t_i), state(t_i))$$

Функция выхода f_o используется в двух вариантах, соответствующих двум версиям конечного автомата: Мили и Мура.

В автомате Мили функция выхода f_o примененная к текущему состоянию и входному символу в момент t_i дает выходной символ:

$$output(t_i) = f_o(input(t_i), state(t_i))$$

В автомате Мура выходной символ зависит лишь от текущего состояния автомата:

$$output(t_i) = f_o(state(t_i))$$

Наиболее популярны два способа описания конечного автомата: табличный и графический. Графический способ рассмотрим на примере модели торгового автомата, продающего банки с газированной водой со стоимостью 15 рублей. Автомат может принимать лишь монеты со стоимостью 5 и 10 рублей.

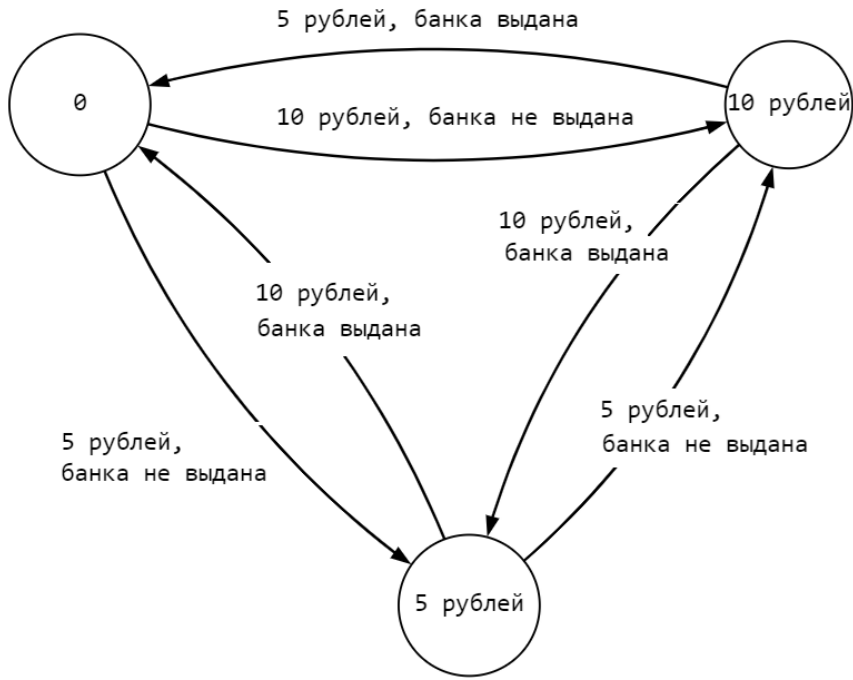


Рисунок 1 – Модель работы торгового автомата

Представим модель, приведенную на рисунке 1, к виду, принятому в теории автоматов. Для этого введем обозначения состояний автомата: S_0 – в накопителе автомата нет монет, S_1 – в накопителе 5 рублей, S_2 – в накопителе 10 рублей. Тогда получаем множество $S = \{S_0, S_1, S_2\}$. Множество входных символов будет состоять из двух символов $I = \{a, b\}$, где ‘a’ условно означает 5 рублей, и ‘b’ – 10 рублей. Множество выходных символов O представим в виде ‘0’ и ‘1’, где ‘0’ – банка с водой не выдана, ‘1’ – банка с водой выдана. Таким образом, третье множество имеет вид $O = \{0, 1\}$.

На рисунке 2 представлен преобразованный граф торгового автомата. На переходах графа показаны входные и выходные символы в виде дроби: $input(t_i)/output(t_i)$. Например, $b/0$ означает, что на вход автомата поступил символ ‘b’ (10 рублей) и автомат выдал символ ‘0’ (банка не была выдана).

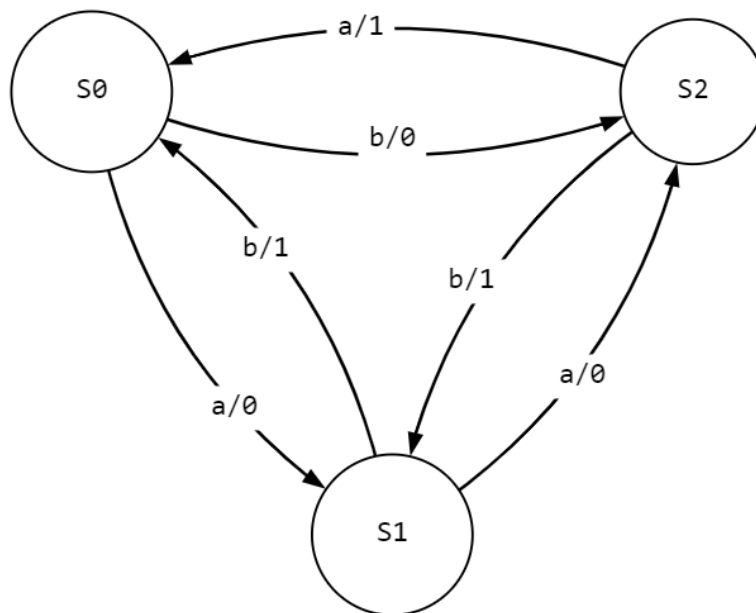


Рисунок 2 – Конечный автомат

Рассмотренный автомат относится к автоматам Мили, так как символ, получаемый на выходе, зависит от состояния, в котором находился в данный момент автомат, и от символа, поступившего на вход автомата.

Рассмотрим табличное описание данного торгового автомата (табл. 2). Если автомат находился в состоянии S_1 и на его вход поступил символ 'b', то автомат перейдет в состояние S_0 и выдаст на выход символ '1'. В данном случае в таблице выбирается столбец S_1 и строка 'b', их пересечение и дает новое состояние автомата и выходной символ.

Таблица 2

Символы входного алфавита $input(t_i)$	Текущие состояния автомата $state(t_i)$		
	S_0	S_1	S_2
	Состояния автомата на следующем такте $state(t_{i+1})$		
a	S_1	S_2	S_0
b	S_2	S_0	S_1
Символы выходного алфавита $output(t_i)$			
a	0	0	1
b	0	1	1

Перед началом моделирования автомат должен быть инициализирован в состояние S_0 .

Рассмотрим пример автомата Мура, у которого выходной символ зависит лишь от состояния, в котором автомат находится. Предположим, что у нас есть некое устройство со встроенным таймером, которое при нажатии на кнопку отправляет в электрическую цепь одиночный импульс напряжения. Если кнопка не отжимается в течение нескольких временных тактов, импульс не должен повторяться вновь и вновь. Алгоритм работы устройства таков: через равные интервалы времени идет опрос кнопки и если она отжата, то на вход автомата идут символы a, a, a и т. д. и, напротив, если кнопка находится в нажатом состоянии, на вход идут символы 'b', 'b', 'b'... В выходной алфавит включим два символа: '0' и '1'. '0' означает отсутствие импульса напряжения, '1' – скачок импульса. Построенный граф приведен на рисунке 3.

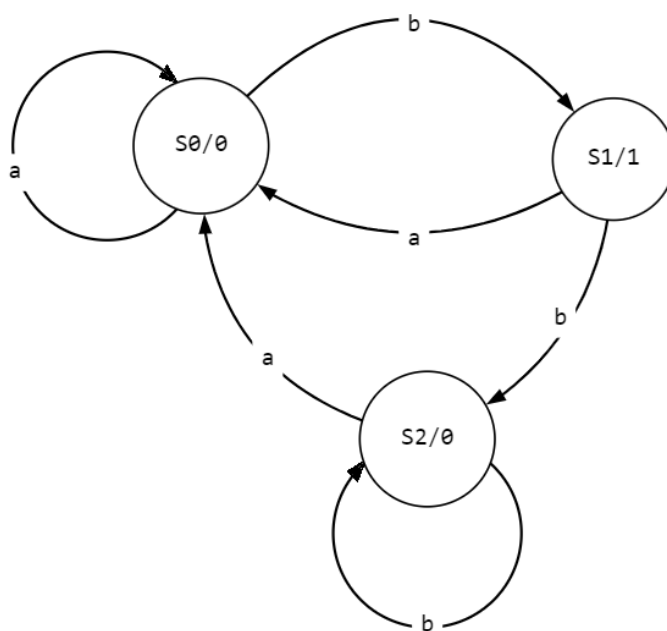


Рисунок 3 – Граф автомата Мура

Граф имеет три вершины: S_0 – соответствует отжатой кнопке, S_1 – кнопка была нажата, S_2 – кнопка продолжает удерживаться в нажатом состоянии. В вершинах помимо состояния

автомата в знаменателе дроби указаны символы выходного алфавита. То есть, если автомат находится в состояниях S_0 и S_2 автомат вырабатывает '0' (импульс отсутствует), если в состоянии S_1 на выходе получаем сигнал '1' (устройство вырабатывает импульс напряжения) (табл. 3).

Таблица 3

Символы входного Алфавита $input(t_i)$	Текущие состояния автомата и символы выходного алфавита $state(t_i)/output(t_i)$		
	$S_0/0$	$S_1/1$	$S_2/0$
	Состояния автомата на следующем такте $state(t_{i+1})$		
a	S_0	S_0	S_0
b	S_1	S_2	S_2

Отсутствие петли у состояния S_1 показывает, что автомат в нем находится лишь в течение одного временного такта, вырабатывая при этом символ 1. Как и в предыдущем примере автомат начинает свою работу с состояния S_0 .

Пример моделирования автомата

Граф состояний рассматриваемого конечного автомата показан на рисунке 4. Множества входного и выходного алфавитов представлены битами: $I = \{0, 1\}$ и $O = \{0, 1\}$, множество состояний содержит 4 буквы: $S = \{a, b, c, d\}$. Необходимо смоделировать работу автомата, если на его вход подается последовательность '1010110' (левые биты подаются первыми). Стартовое состояние – 'a'.

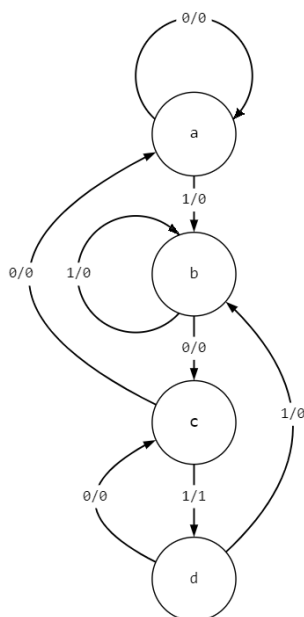


Рисунок 4. Граф двоичного автомата

Моделирование проводим в таблице 4. Первая ее строка заполняется исходя из того, что на вход подается бит 1 ($input(t_0) = 1$) и автомат находится в состоянии 'a' ($state(t_0) = 'a'$). По графу автомат из состояния 'a' переходит в состояние 'b', вырабатывая при этом бит 0. На втором такте (вторая строка таблицы) на вход поступает бит 0, сам автомат находится в состоянии 'b'. Следовательно, следующее состояние будет 'c' и на выход будет выдан бит 0 (см. рис. 4). Аналогично заполняются и другие строки таблицы.

Таким образом, выходная последовательность имеет вид '0010100'. Табличное представление рассмотренного автомата представлено в таблице 5.

Таблица 4

$input(t_i)$	$state(t_i)$	$state(t_{i+1})$	$output(t_i)$
1	a	b	0
0	b	c	0
1	c	d	1
0	d	c	0
1	c	d	1
1	d	b	0
0	b	c	0

Символы входного алфавита $input(t_i)$	Текущие состояния автомата $state(t_i)$			
	a	b	c	d
	Состояния автомата на следующем такте $state(t_{i+1})$			
0	a	c	a	c
1	b	b	d	b
	Символы выходного алфавита $output(t_i)$			
0	0	0	0	0
1	0	0	1	0

Контрольная работа

Тест реализован на языке Java и размещен на авторском сайте [5] по адресу <https://altaev-aa.narod.ru/km3/fa.html>. Программа представляет собой jar файл, для запуска которого требуется наличие на компьютере виртуальной java машины. На страничке сайта имеются все необходимые указания, достаточные для запуска программы.

Программа рандомно определяет мощность и состав множества состояний S автомата, алфавитов входного I и выходного O слов. Тестируемый, используя эти множества, должен сам построить детерминированный конечный автомат Мили, заполнив таблицы функций переходов и выхода. Затем программа генерирует случайное входное слово, по которому тестируемый должен смоделировать работу автомата. По завершении теста выводится сообщение о количестве допущенных ошибок.

ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ

Такие модели называются также вероятностными автоматами, цепями Маркова [1] или Р-схемами (probability machines). В таких моделях переходы между состояниями автомата и выработка выходного слова подчинены влиянию случайных закономерностей. Это означает, что функции $state(t_{i+1})$ и $output(t_i)$ описываются матрицами вероятностей.

Реализация вероятностных переходов

В качестве примера рассмотрим автомат, в котором переходы из одного состояния в другое происходят случайным образом, а выработка выходного символа – как и в детерминированном автомате Мура (рис. 5). Для упрощения модели принято, что $S = \{0, 1, 2, 3\}$ и $O = \{0, 1, 2, 3\}$, а множество входного алфавита состоит из одного символа $I = \{a\}$. Так как на вход автомата поступает один и тот же символ 'a', он на графе не отображен, но взамен на переходах (дугах) показаны их вероятности. Если автомат находится в состоянии 0, то существует вероятность 75 %, что он перейдет в состояние 1 и 25 % – в состояние 2, выдав на свой выход символ '0'. Из состояния 1 автомат всегда переходит в третье состояние (из этой вершины исходит лишь одна дуга). То же утверждение касается состояния 2, который будет всегда заменяться состоянием 3. Из состояния 3 равновероятны переходы в состояния 0 и 3 (т. е. автомат не меняет состояние). Из данного примера видно, что сумма вероятностей переходов на исходящих дугах должна быть всегда равна 1.

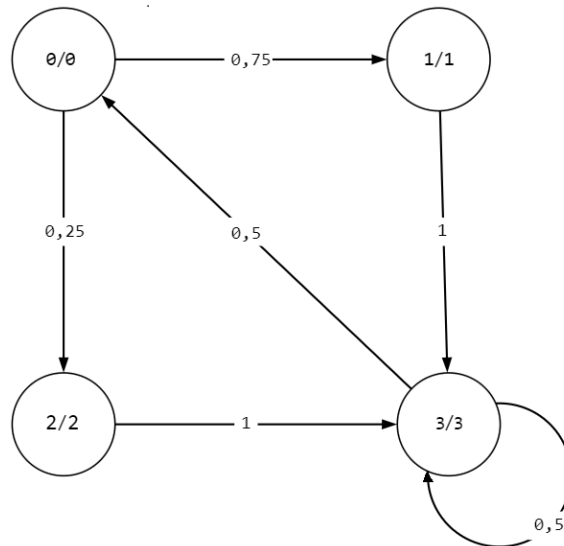


Рисунок 5 – Цепь Маркова

Матрица вероятностей переходов будет иметь вид

$$A = \begin{pmatrix} 0 & 0,75 & 0,25 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0,5 & 0 & 0 & 0,5 \end{pmatrix}$$

Элемент матрицы a_{ij} представляет собой вероятность перехода из i состояния в j состояние. Таким образом, строки матрицы соответствуют текущим состояниям, а столбцы – новым и сама матрица является квадратной. Сумма элементов всей строки должна быть равна 1

$$\sum_{j=0}^3 a_{ij} = 1$$

Инициализация начального состояния автомата также выполняется случайным образом, для чего в описание автомата должен быть введен вектор вероятностей начальных состояний. Например, такой

$$C = \begin{pmatrix} 0,8 \\ 0,1 \\ 0,1 \\ 0 \end{pmatrix}$$

Сумма элементов этого вектора должна быть равна 1.

Рассмотрим более сложный вариант автомата, добавив в его входной алфавит еще один символ: $I = \{a, b\}$. Цепь Маркова для символа 'а' оставим прежней (см. рис. 5), для второго выберем другой (рис. 6). Следовательно, для описания автомата будут использованы две матрицы

$$A_{input(t)=a} = \begin{pmatrix} 0 & 0,75 & 0,25 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0,5 & 0 & 0 & 0,5 \end{pmatrix},$$

$$A_{input(t)=b} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0,5 & 0,5 \\ 0,75 & 0 & 0 & 0,25 \\ 0 & 0,25 & 0,25 & 0,5 \end{pmatrix}$$

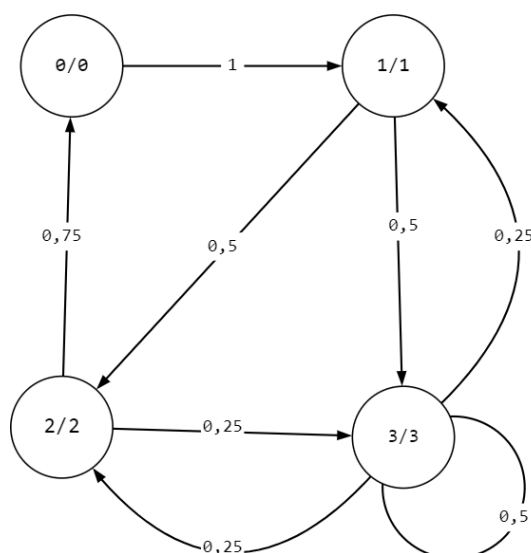


Рисунок 6 – Цепь Маркова для символа ‘b’

Рассмотрим, как выполняются переходы в вероятностном автомате. Пусть на вход автомата поступает символ ‘b’ и он находится в этот момент в состоянии ‘3’. Следовательно, для вычисления будущего состояния используем вторую матрицу и в ней выбираем последнюю строку (рис. 7).

$$A_{/input(t)=br} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0,5 & 0,5 \\ 0,75 & 0 & 0 & 0,25 \\ 0 & 0,25 & 0,25 & 0,5 \end{pmatrix}$$

↑ Входной символ ← Вероятности для состояния ‘3’

Рисунок 7 – Выбор вероятностей

Для понимания того, как выполняются переходы, отложим на числовой оси отрезки, соответствующие вероятностям 0,25, 0,25 и 0,5 (рис. 8).

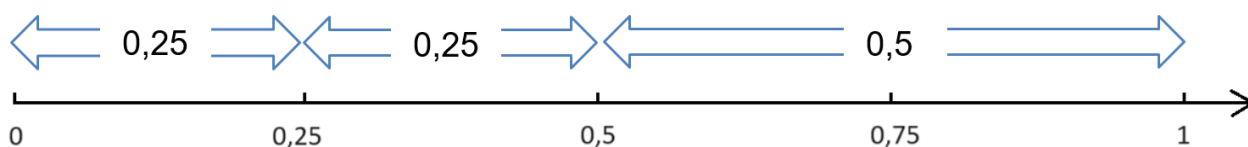


Рисунок 8 – Числовая ось с отложенными на ней отрезками

Выбор состояния выполняется с помощью случайной величины, принимающей любые значения из интервала [0; 1]. Если эта величина примет значение, попадающее в интервал:

- [0; 0,25), то автомат перейдет в состояние 1;
- [0,25; 0,5), перейдет в состояние 2;
- [0,5; 1], перейдет в состояние 3.

Реализация функции выхода в стохастическом режиме

В предыдущих двух примерах рассматривались автоматы Мура, так как их выходные символы были привязаны к состояниям. Усложним второй автомат, добавив в множество O дополнительный символ ‘4’ и введя матрицы вероятностей выхода.

$$B_{/input(t)=\text{'a'}} = \begin{pmatrix} 0 & 0,5 & 0,25 & 0,25 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0,5 \end{pmatrix}$$

$$B_{/input(t)=\text{'b'}} = \begin{pmatrix} 0,5 & 0 & 0,5 & 0 & 0 \\ 0,5 & 0 & 0 & 0,5 & 0 \\ 0,5 & 0,25 & 0 & 0 & 0,25 \\ 0 & 0 & 0,4 & 0 & 0,6 \end{pmatrix}$$

В отличие от матриц A эти матрицы прямоугольные, так как строки соответствуют состояниям, а столбцы – выходным символам. Так, например, если автомат находился в состоянии ‘0’ и на ход поступил символ ‘а’, то имеется 50%-ная вероятность, что на выходе будет символ ‘1’, по 25 %, что будет либо символ ‘2’, либо ‘3’.

Как и в случае с A , для матриц B должно выполняться условие

$$\sum_{j=0}^4 b_{ij} = 1$$

Контрольная работа

Тест размещен на авторском сайте по адресу <https://altaev-aa.narod.ru/km3/pa.html>. На странице даны все необходимые указания для запуска теста.

В окне программы тестируемый должен ввести свою фамилию, далее программа случайно определяет размеры и состав множества состояний автомата и алфавитов входного и выходного слов. Далее тестируемый должен построить сам автомат, заполнив таблицы вероятностей переходов, выхода и начального состояния. На следующем этапе программа генерирует входное слово и предлагает смоделировать работу автомата. По завершении теста выводится сообщение о количестве допущенных ошибок.

НЕПРЕРЫВНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ

К таким типам моделей, называемых также Q -схемами (queue, очередь), относятся системы массового обслуживания (СМО, queuing system). Примеры СМО: обслуживание автомобилей на станциях технического обслуживания или на автозаправках, клиентов в заведениях общепита и т.п.

Потоки событий

Одним из основных понятий Q -схем является событие, означающее любое изменение состояния модели. Это может быть прибытие автомобиля на автозаправку или, напротив, его отъезд с АЗС по завершении заправки. Для отсчета момента наступления событий используют абсолютные $\{t_i\}$ и относительные $\{\tau_i\}$ времена (рис. 9). Абсолютным будет время, прошедшее с начала наблюдения за моделью до момента наступления события, относительным считается время, отсчитываемое между двумя соседними по временной шкале событиями.

События, наблюдаемые во время моделирования, образуют поток событий. Рассмотрим классификацию потоков событий.

Различают потоки однородных и неоднородных событий. В первом случае события отличаются друг от друга лишь временем их наступления. Примером может быть фиксация лишь прибытия автомобилей на СТО. Примером потока неоднородных событий является фиксация прибытия и убытия автомобилей с СТО.

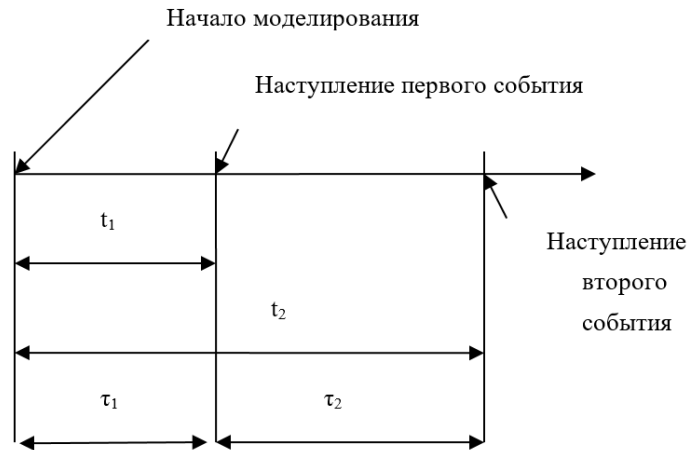


Рисунок 9 – Два способа отсчета времени наступления событий

Если времена между тремя последовательно наступающими событиями $\{\tau_{i-1}\}$ и $\{\tau_i\}$ не связаны между собой, то такая последовательность событий называется потоком с ограниченным последствием.

Рассмотрим малый промежуток времени $[t, t+dt]$. Вероятность ненаступления события на этом временном интервале обозначим через P_0 , вероятность наступления одного события – как P_1 и вероятность наступления двух и более событий через $P_{>1}$. Так как временной отрезок dt очень мал, то вероятность ненаступления события P_0 намного больше двух остальных вероятностей: $P_0 \gg P_1$ и $P_0 \gg P_{>1}$, но в целом сумма всех трех вероятностей должна быть равна единице: $P_0 + P_1 + P_{>1} = 1$. Могут быть два варианта соотношения второго и третьего слагаемых: они либо сопоставимы между собой по значению: $P_1 \approx P_{>1}$, либо последнее слагаемое пренебрежимо мало: $P_1 \gg P_{>1}$. В первом случае говорят, что в модели наблюдается неординарный поток событий, т. е. допускается, что во время моделирования одновременно могут наступать два и более событий. Во втором случае третьим слагаемым можно пренебречь: $P_0 + P_1 \approx 1$. В таком случае говорят, что в модели наблюдается ординарный поток событий, т. е. не предполагается одновременное наступление двух или более событий.

Введем понятие интенсивности наступления событий

$$\lambda = \frac{\lim_{dt \rightarrow 0} P(t, t+dt)}{dt}$$

Как видим, эта та же вероятность наступления события, но на малом промежутке времени dt . Эта величина может быть постоянной или являться функцией от времени (рис. 10). В первом случае говорят, что в модели наблюдается стационарный поток событий ($\lambda = \text{const}$), во втором – нестационарный поток ($\lambda = \lambda(t)$). График нестационарного потока, изображенный на рисунке 10, показывает, что в начале моделирования события наступали редко, но в конце времени моделирования события наступали чаще (график имеет возрастающий вид).

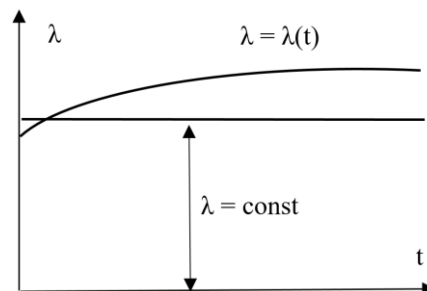


Рисунок 10 – Стационарный и нестационарный потоки событий

Основные элементы СМО

Объекты, используемые в моделях систем массового обслуживания, относятся к двум типам:

- динамические (транзакты, второе название – заявки);
- статические (каналы, очереди, логические вентили).

Транзакты могут перемещаться по модели или оставаться на месте, если их дальнейшее перемещение невозможно из-за наличия какого-то блокирующего условия. В сложных моделях из транзактов могут создаваться их клоны, или же, напротив, эти клоны могут сливаться в один общий транзакт.

Канал

Канал (второе название – прибор) на схемах (рис. 11) обозначается кружком. Стрелки обозначают маршруты движения транзактов. Канал может находиться в трех состояниях:

- свободен;
- обслуживание;
- не рабочее.

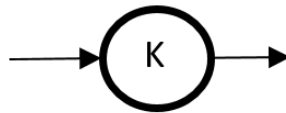


Рисунок 11 – Канал

Состояние «свободен» означает, что в данный момент в канале нет транзакта. Если в канал вошел транзакт, то канал переходит в состояние обслуживания. Если канал переходит в нерабочее состояние, то находящийся в нем транзакт удаляется, т. е. его обслуживание прерывается. Такое состояние еще называется состоянием прерывания.

Очередь

В канале может обслуживаться лишь один транзакт. Примером работы канала может быть кассир магазина и покупатель. Кассир в какой-то момент времени может обслуживать только одного покупателя. Другие транзакты, претендующие на обслуживание в данном канале, образуют перед ним очередь (накопитель заявок). Условное обозначение очереди на схемах показано на рисунке 12.

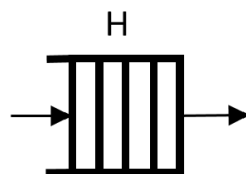


Рисунок 12 – Очередь

Стрелками показаны потоки входящих/исходящих в/из очереди транзактов. Как правило, для очереди действует правило FIFO (First In, First Out – первым вошел, первым вышел). Емкостью накопителя L^H называется максимальное число транзактов, которое может он в себя одновременно поместить. Текущей длиной ℓ очереди является число транзактов, находящихся в данный момент в накопителе. Очередь может находиться в трех состояниях:

- очередь пуста ($\ell = 0$);
- очередь не пуста ($0 < \ell < L^H$);
- очередь заполнена ($\ell = L^H$), т. е. для транзактов, пытающихся войти в очередь, нет свободных мест.

Встречаются два типа очередей:

- ограниченной длины, $L^H = A$, где A – целая положительная константа, например, $A = 10$ (всего 10 мест для транзактов);
- неограниченной длины, $L^H = \infty$, очередь может содержать любое количество транзактов.

Вентиль

К статическим объектам, помимо канала и очереди, относится также логический вентиль, или второе название – клапан Кл (рис. 13).

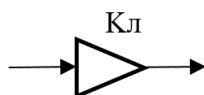
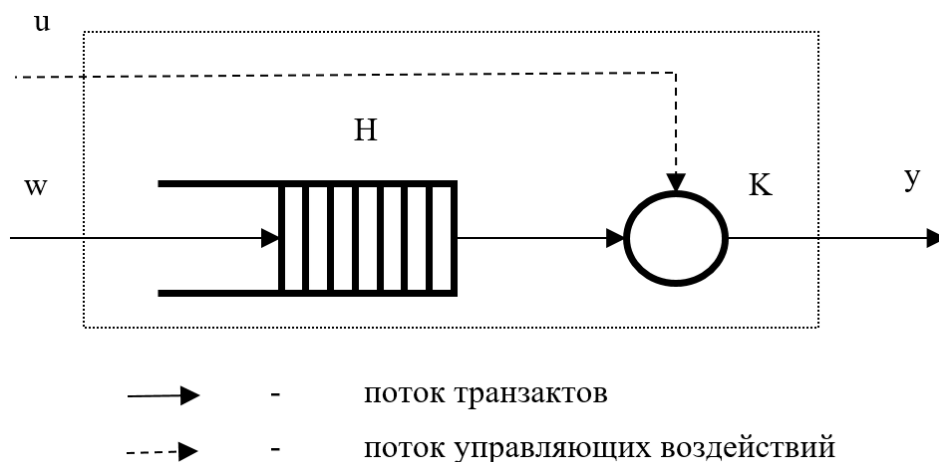


Рисунок 13 – Клапан

Вентиль может находиться в двух состояниях:

- открыт – транзакты проходят через вентиль;
- закрыт – клапан через себя транзакты не пропускает.

На рисунке 14 изображен один из вариантов СМО, состоящий из канала K и очереди (накопителя) H . Сплошными стрелками обозначены потоки входящих (w) и исходящих (y) транзактов, пунктирной стрелкой – поток управляющих воздействий u .



- ▶ - поток транзактов
- - -▶ - поток управляющих воздействий

Рисунок 14 – СМО

Самый простейший вариант СМО можно получить в случае, когда в модели нет управляющих воздействий u и очередь имеет бесконечную длину.

В более сложном варианте модели очередь имеет ограничение на свою максимальную длину $L^H = A$ (рис. 15). В модели имеются два клапана, которые работают по следующему принципу:

- если $\ell < L^H$, то клапан $K_{л1}$ открыт, а клапан $K_{л2}$ закрыт. Это означает, что очередь не заполнена полностью и приходящие в модель транзакты (поток w) могут входить в очередь H ;
- если $\ell = L^H$, то $K_{л1}$ закрыт, $K_{л2}$ открыт, так как очередь заполнена. Транзакты через клапан $K_{л2}$ уходят из модели. Такие транзакты теряют право на обслуживание в канале K и считаются «потерянными».

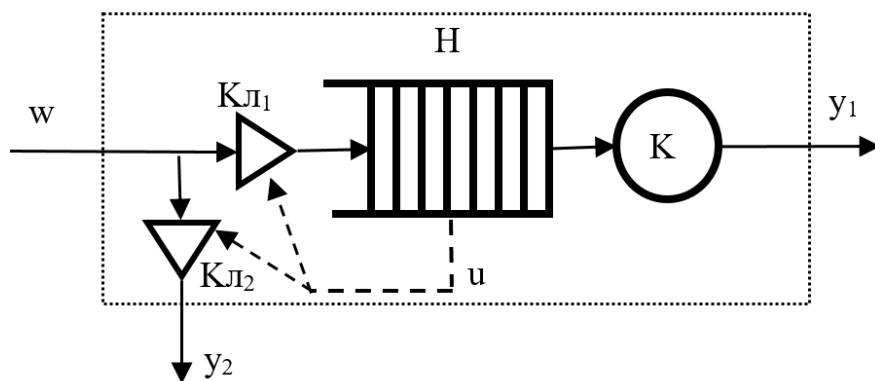


Рисунок 15 – СМО с потерями

В данной модели СМО имеется два потока исходящих из нее транзактов: обслуженных в канале (поток y_1) и «потерянных» (поток y_2). Такие модели называются СМО с потерями. Мы также видим, что в этом типе модели есть поток управляющих воздействий u , направленный от очереди к клапанам.

Транзакт

Рассмотрим особенности динамического объекта модели СМО – транзакта (заявки). У транзактов могут быть приоритеты:

- статические и динамические;
- относительные и абсолютные.

Статический приоритет транзакт получает при входе в модель и в дальнейшем он не меняется. Динамический приоритет по мере перемещения транзакта по модели может менять свое значение.

Относительные приоритеты могут учитываться при прохождении транзактов через очередь (рис. 16).

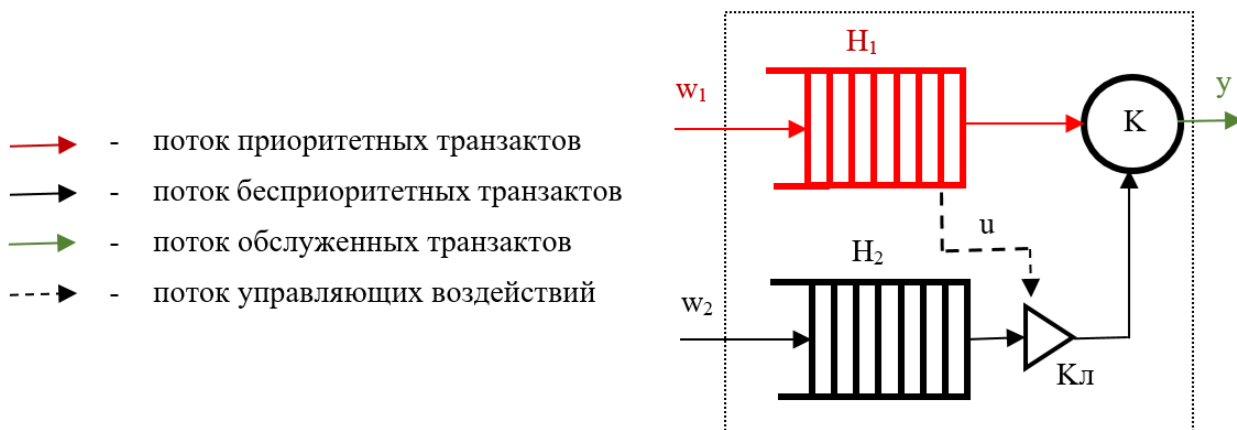


Рисунок 16 – Прохождение модели транзактами с относительными приоритетами

Приоритетные транзакты имеют преимущество перед бесприоритетными при прохождении ими модели. Такие транзакты образуют перед каналом отдельную очередь H_1 . Перед очередью из бесприоритетных транзактов H_2 имеется клапан КЛ, работающий по следующему принципу:

- клапан открыт, если в накопителе H_1 нет заявок, т. е. нет приоритетных транзактов и в этом случае бесприоритетные транзакты имеют право на обслуживание в канале;
- клапан закрыт, если имеются в очереди H_1 транзакты с повышенным приоритетом и, следовательно, бесприоритетные транзакты не имеют право на обслуживание в канале.

В данном примере СМО имеется поток управляющих воздействий u , направленный от накопителя H_1 к клапану.

В СМО, где имеется помимо потока беспriorитетных транзактов поток заявок с абсолютными приоритетами, дисциплина обслуживания таких заявок иная. Транзакты с абсолютными приоритетами в очереди не задерживаются, а сразу захватывают канал (рис. 17). Беспriorитетный транзакт, находившийся в этот момент в канале на обслуживании, покидает модель необслуженным.

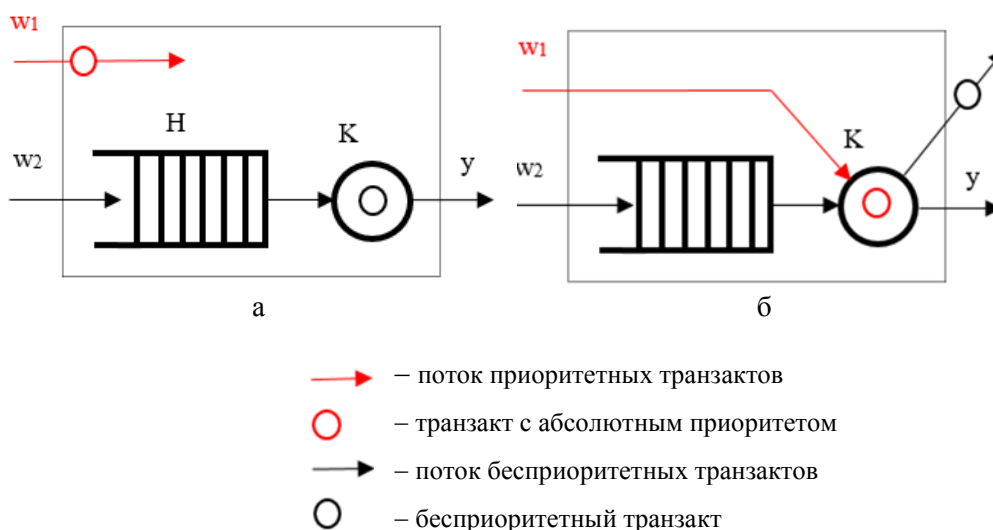


Рисунок 17 – Захват канала высокоприоритетным транзактом: а – приход в прибор транзакта с абсолютным приоритетом; б – захват канала транзактом с абсолютным приоритетом

После захвата канал переходит в состояние прерывания.

Часто моделируются устройства, занимающиеся обработкой каких-то деталей. При выходе из строя такое устройство какое-то время будет находиться в ремонте, т. е. находится в нерабочем состоянии. Его в модели можно представить в виде канала, а обрабатываемую деталь как транзакт с обычным приоритетом. Перевод устройства в нерабочее состояние можно смоделировать, подав на такой канал высокоприоритетный транзакт и вызвав тем самым у него состояние прерывания. Деталь (низкоприоритетный транзакт) будет удалена из устройства (канала) без завершения ее полной обработки. В зависимости от особенностей моделируемой реальной системы необработанная деталь после устранения поломки устройства может быть впоследствии вновь подана на устройство для повторной обработки, или же будет считаться браком и удаляться. Общее резюме по СМО с прерываниями сведено в таблицу 6.

Таблица 6

Реальная система	Модель СМО
Устройство	канал
Обрабатываемая деталь	низкоприоритетный транзакт
Устройство в данный момент не обрабатывает деталь	канал свободен
Устройство обрабатывает деталь	в канале на обслуживании находится низкоприоритетный транзакт
Устройство из-за поломки находится в нерабочем состоянии (в ремонте)	в канал помещен высокоприоритетный транзакт, и канал переведен в состояние прерывания

Источник

Источник обеспечивает прибытие транзактов в модель (рис. 18).

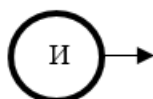


Рисунок 18 – Источник транзактов

Как видим по схеме, особенностью источника является генерация транзактов, у него отсутствует входной поток транзактов, имеется лишь исходящий поток. Созданные транзакты, покидая источник, образуют Пуассоновский поток событий, так как он однородный, ординарный (транзакты покидают источник по одному) и с ограниченным последствием. Также в большинстве случаев такой поток событий является стационарным.

Типы событий

Рассмотрим простейшую систему массового обслуживания, содержащую источник И, очередь Н бесконечной длины и канал К (рис. 19). Источник находится вне модели.

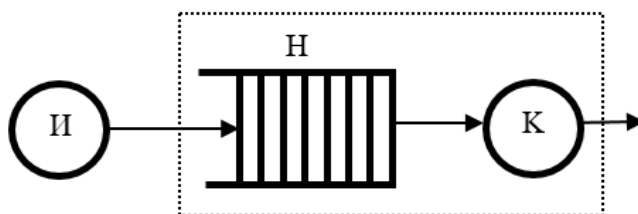


Рисунок 19 – Простейшая СМО

События, которые могут происходить в такой модели, делятся на два типа: основные и вспомогательные. Момент наступления основного события определяется случайно, это означает, что в модели должен быть реализован алгоритм вычисления времени наступления такого события. К основным событиям в рассматриваемой модели относятся приход транзакта в модель, что равнозначно выходу созданного транзакта из источника. Вторым основным событием является завершение обслуживания транзакта в канале.

Вспомогательные события не наступают сами по себе, они появляются в модели лишь как следствие наступления основного события. Примеры вспомогательных событий:

- вход транзакта в канал;
- вход транзакта в очередь;
- выход транзакта из очереди.

Первые два события связаны с приходом транзакта в модель. Когда транзакт входит в модель, то проверяется состояние канала: свободен он или нет. Если да, то прибывший транзакт занимает канал, переводя его в состояние «занято». При этом транзакт в очереди не задерживается. Если же канал занят обработкой другого транзакта, то прибывший транзакт попадает в очередь. Очередь же транзакт покидает, когда наступит второе основное событие: завершение обработки предыдущего транзакта в канале. Связь между основными и вспомогательными событиями приведена в таблице 7.

Таблица 7

Основное событие	Проверка условия	Вспомогательные события
Приход транзакта в модель	Свободен канал?	Да. Прибывший транзакт занимает канал. Канал переводится в состояние «занято» и начинается обслуживание поступившего транзакта
		Нет. Транзакт помещается в очередь
Завершение обслуживания транзакта в канале.	Обработанный транзакт покидает канал и модель	
	Имеются в очереди транзакты?	Да. Транзакт, находившийся в очереди, покидает ее и поступает в канал на обслуживание Нет. Канал переходит в состояние «свободно»

Расчет времени наступления основного события

Время наступления основного события является относительной величиной τ (см. рис. 9), так как оно отсчитывается с момента наступления предыдущего события. Рассмотрим, как происходит расчет времени для приходов транзактов в модель. Когда транзакт покидает источник (см. рис. 18) и входит в модель, то в источник помещается следующий транзакт и вычисляется время его прихода в модель (рис. 20). Первый транзакт помещается в источник с началом моделирования.

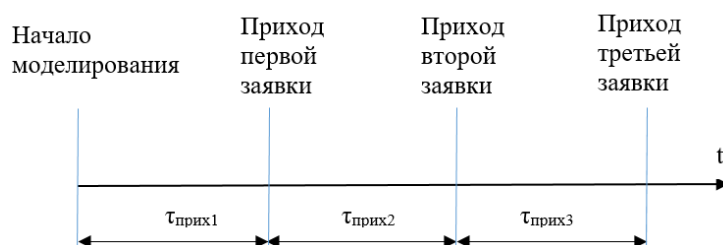


Рисунок 20 – Отсчет времени прихода транзакта в модель

Таким образом, в источнике всегда присутствует транзакт и для него всегда назначено время пребывания в этом источнике.

Что касается второго основного события, то время обслуживания транзакта в канале отсчитывается с момента входа транзакта в канал (рис. 21).

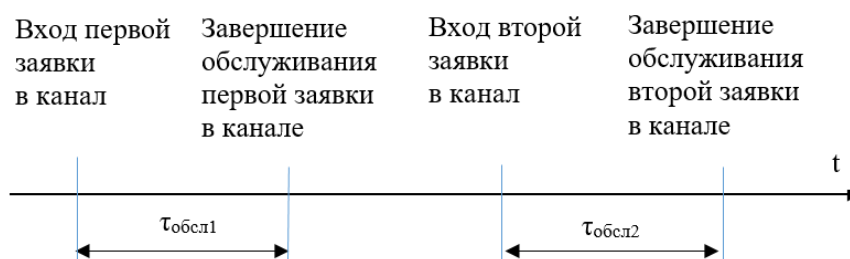


Рисунок 21 – Отсчет времени обслуживания транзакта в канале

Для расчета времени наступления основного события необходимо знать средние значения этого времени, т. е. в нашем случае значения $\bar{\tau}_{\text{прих}}$ и $\bar{\tau}_{\text{обсл}}$. Для вычисления конкретных значений времени наступления события используются различные законы распределения случайной величины (далее – СВ): равномерное, показательное, нормальное и т. д.

Рассмотрим, как вычисляется значение времени на примере равномерного закона распределения СВ. В этом законе для вычисления времени наступления события помимо его среднего значения $\bar{\tau}$ используется еще полуразмах от среднего B . Графическое представление равномерного закона дано на рисунке 22.

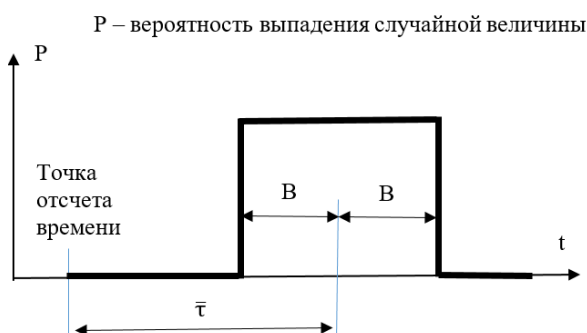


Рисунок 22 – Иллюстрация равномерного закона распределения СВ

Значения времени будут лежать в интервале $[\bar{\tau} - B; \bar{\tau} + B]$ и будут равновероятными между собой. Например, считая, что времена принимают целые значения, примем среднее значение $\bar{\tau} = 8$ и полуразмах от среднего $B = 3$. Тогда значения времени образуют множество чисел $\{5, 6, 7, 8, 9, 10, 11\}$. У времен, находящихся вне интервала $[5; 11]$, будет нулевая вероятность.

Должно выполняться условие $\bar{\tau} > B$, так как если обе величины будут равны, то время может принимать нулевые значения и поток событий перестанет быть ординарным.

Java код программы генерации 1000 случайных чисел с применением равномерного закона распределения случайной величины при среднем значении $\bar{\tau} = 8$ и полуразмахе от среднего $B = 3$ приведен в приложении 1, а результаты обработки одного из прогонов приведены в excel файле <https://disk.yandex.ru/i/fFUQizQzCwFsgw> (рабочий лист «Uniform»). Среднее значение составило 8,093, а частоты выпадения случайного числа приведены в таблице 8.

Таблица 8

Случайное число	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Частота	0	0	0	0	0	134	127	139	160	138	156	146	0	0

График плотности распределения вероятности выпадения СВ приведен на рисунке 23.

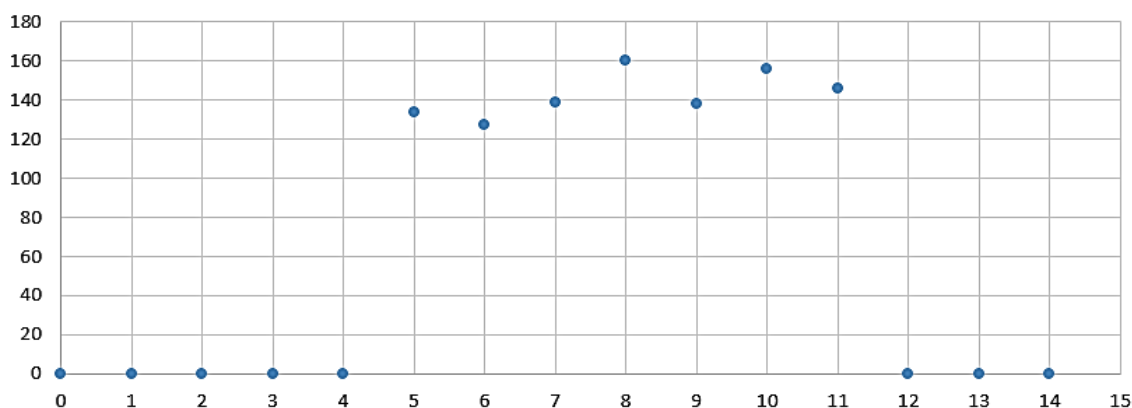


Рисунок 23 – Плотность распределения случайной величины

При каждом новом прогоне программы получаемые данные будут изменяться, и, следовательно, будет меняться и сам график.

Рассмотрим применение другого закона распределения СВ – нормального. Нормальное распределение применяется тогда, когда еще до построения самой модели путем наблюдения за исходной системой проводится сбор данных, что позволяет набрать нужную статистику для получения средних значений требуемых характеристик. Например, в течение часа записываются интервалы времени между подъезжающими к светофору автомобилями: $\{\tau_1, \tau_2, \tau_3, \dots, \tau_n\}$, где n – число автомобилей. Далее вычисляется среднее значение интервала времени прибытия автомобилей

$$\bar{\tau} = \frac{1}{n} \sum_{i=1}^n \tau_i$$

и среднеквадратичное отклонение

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\tau_i - \bar{\tau})^2}$$

Второй параметр σ показывает насколько сильно значения τ_i отличаются от их среднего значения: чем сильнее выражен разброс между τ_i , тем больше величина σ . На рисунке 24 приведен график вероятности P выпадения случайного числа согласно нормального закона. У чисел, близких к $\bar{\tau}$, вероятность выше.

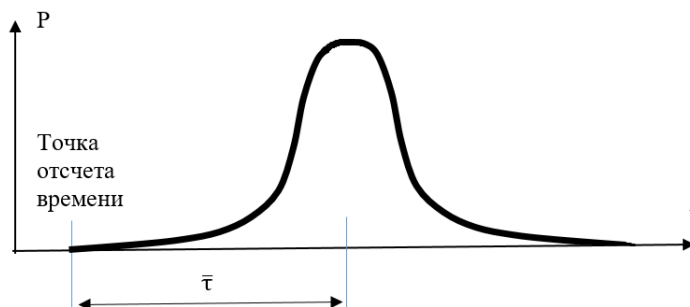


Рисунок 24 – Иллюстрация нормального закона распределения СВ

Java-код программы генерации случайных чисел с применением нормального закона распределения для среднего значения $\bar{\tau} = 8$ и среднеквадратичного отклонения σ приведен в приложении 2. Результаты двух из прогонов (при $\sigma = 1$ и $\sigma = 2$) данной программы для 10000 случайных чисел приведены в таблице 9 (см. также excel файл <https://disk.yandex.ru/i/fFUQizQzCwFsgw>, рабочий лист «Normal»).

Примечание. При новых прогонах программы результаты будут другими.

Таблица 9

Случайное число	0	1	2	3	4	5	6	7	8
Частота при $\sigma = 1$	0	0	0	0	5	46	618	2359	3810
Частота при $\sigma = 2$	0	9	17	111	270	648	1198	1678	1915

Продолжение таблицы 9

Случайное число	9	10	11	12	13	14	15	16	17
Частота при $\sigma = 1$	2480	613	57	12	0	0	0	0	0
Частота при $\sigma = 2$	1811	1217	649	350	94	27	5	1	0

Графики частот выпадения СВ приведены на рисунке 25. Эти графики показывают, что чем меньше среднеквадратичное отклонение σ , тем ближе к среднему значению располагаются случайные числа.

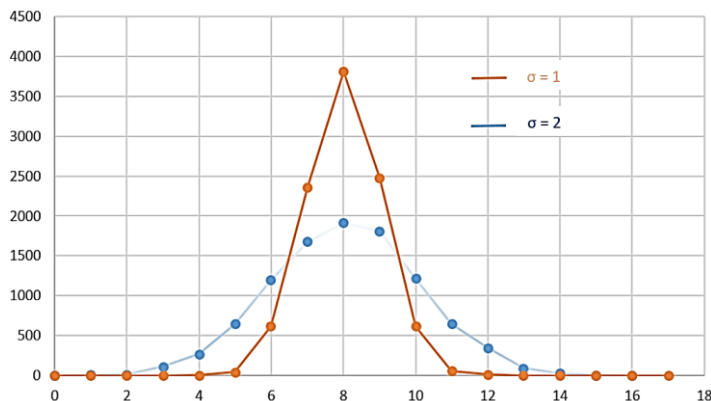


Рисунок 25 – Графики распределения частот СВ по нормальному закону в зависимости от среднеквадратичного отклонения σ

Алгоритм изменения модельного времени

Модельное время t является абсолютным временем и меняется с переменным шагом Δt в интервале $[0; T]$, где T – время всего моделирования. Приращение Δt определяется как минимальное из трех времен: $\min\{\tau_{\text{прих}}, \tau_{\text{обсл}}, T-t\}$, где $T-t$ – время, оставшееся до конца моделирования.

Моделирование простейшей СМО

После ознакомления с алгоритмами вычисления времен наступления основных событий рассмотрим пример моделирования простейшей СМО (см. рис. 19). Расчеты проведем в excel таблице. Выберем простейший закон распределения СВ – равномерный. Случайные числа выбираем из разрешенного диапазона в произвольном порядке.

Пусть даны следующие исходные данные: $\tau_{\text{прих}} = 7 \pm 2$, $\tau_{\text{обсл}} = 9 \pm 1$, $T = 40$. Считая, что время будет представлено в целых числах, время прихода транзакта будет выбираться из множества $\{5, 6, 7, 8, 9\}$, время обслуживания будет принимать значения $\{8, 9, 10\}$. Моделируя работу генератора случайных чисел, в произвольном порядке выбираем числа из первого множества. Пусть мы получим следующую последовательность для $\tau_{\text{прих}} = 5, 9, 9, 6, 6, 6$. Сумма этих чисел равна 41 (она не должна быть меньше, чем T). Первое число 5 – это время прихода в модель первого транзакта, следующее число 9 – второго, далее 9 – третьего и т. д. Аналогично, используя второе множество, получаем вторую числовую последовательность: $\tau_{\text{обсл}} = 9, 8, 8, 8, 10, 10$. В этом ряду: 9 – время обслуживания первого транзакта, 8 – второго, 8 – третьего и т. д. Введем полученные данные в таблицу (рис. 26). Для визуализации перемещений транзактов по модели ячейки таблицы, соответствующие разным транзактам, раскрасим в разные цвета.

Подготовим таблицу, в которой будем проводить моделирование (рис. 27). В начале моделирования модельное время $t = 0$ (ячейка A4), первый транзакт помещен в источник И (см. рис. 19), его время выхода из источника и соответственно прибытия в модель равно 5 (ячейка B4 закрашена в цвет первого транзакта). Следующий столбец отображает текущую длину очереди l и так как в начале моделирования в очереди пока нет транзактов, то в ячейку C4 вписываем 0. В канале также нет транзакта и в ячейке D4 ставим прочерк. Время до завершения моделирования равно 40 (ячейка E4).

Н	I	J	K	L	M
	$\tau_{\text{прих}} = 7 \pm 2$		$\tau_{\text{обсл}} = 9 \pm 1$		
	5		9		1-й транзакт
	9		8		2-й транзакт
	9		8		3-й транзакт
	6		8		4-й транзакт
	6		10		5-й транзакт
	6		10		6-й транзакт
сумма=	41				

Рисунок 26 – Времена наступления основных событий

	A	B	C	D	E	F
1	Моделирование СМО					
2						
3	t	t_прих	L	t_обсл	T-t	delta_t
4	0	5	0	-	40	5

Рисунок 27 – Начало моделирования

Находим время, которое должно пройти до наступления ближайшего основного события: $\Delta t = \min\{\tau_{\text{прих}}, T-t\} = \min\{5, 40\} = 5$ и его вписываем в ячейку F4.

На следующем шаге переходим к моменту свершения основного события (прихода первой заявки), для чего увеличиваем модельное время $t = t + \Delta t = 0 + 5 = 5$. Заполняем строку таблицы, соответствующую измененному состоянию модели (рис. 28).

	A	B	C	D	E	F
3	t	t_прих	L	t_обсл	T-t	delta_t
4	0	5	0	-	40	5
5	5	9	0	9	35	9

Рисунок 28 – Вход первого транзакта в модель и начало его обслуживания в канале

В A5 вписываем модельное время $t = 5$, в источник И помещаем второй транзакт, время прихода которого равно 9 (ячейка B5). Первый транзакт, пройдя без задержки через очередь (в ячейке C5 вписываем 0), попадает в канал на обслуживание (в ячейке D5 указываем время обслуживания 9). До конца моделирования остается 35 единиц времени (E5). Для заполнения последнего столбца находим минимальное значение среди трех времен: $\Delta t = \min\{\tau_{\text{прих}}, \tau_{\text{обсл}}, T-t\} = \min\{9, 9, 35\} = 9$.

Далее вновь меняем модельное время $t = t + \Delta t = 5 + 9 = 14$ и начинаем заполнять следующую строку таблицы (рис. 29). На данном шаге совпали моменты наступления двух основных событий: прихода второго транзакта и завершения обслуживания первого транзакта в канале. Следует вначале освободить канал и поместить в него следующий транзакт. Таким образом, в источнике теперь находится третий транзакт со временем прибытия в модель 9, очередь по-прежнему пуста, в канале на обработке находится второй транзакт. Первый транзакт удален из модели. Приращение времени принято равным 8.

	A	B	C	D	E	F	G
3	t	t_прих	L	t_обсл	T-t	delta_t	Комментарии
6	14	9	0	8	26	8	1-й транзакт ушел и модели обслуженным

Рисунок 29 – Уход первого транзакта из модели и начало обслуживания в канале второго

Рассмотрим состояние модели, когда модельное время было увеличено на 8 единиц и стало равным 22 (рис. 30). Так как на предыдущем шаге Δt было принято равным $\tau_{\text{обсл}}$, то при $t = 22$ канал освобождается от обслуженного второго транзакта и переходит в состояние «свободно», так как третий транзакт в модель еще не вошел. Третьему транзакту войти в модель предстоит через одну единицу времени. Пояснение: на предыдущем шаге у третьего транзакта $\tau_{\text{прих}}$ было равным 9, при переходе на следующий шаг время изменилось на $\Delta t = 8$ и, следовательно, пересчитываем время $\tau_{\text{прих}} = 9 - 8 = 1$.

	A	B	C	D	E	F	G
3	t	t_прих	L	t_обсл	T-t	delta_t	Комментарии
7	22	1	0	-	18	1	2-й транзакт ушел из модели обслуженным

Рисунок 30 – Уход из модели второго транзакта

Другие изменения модели показаны на рисунке 31.

	A	B	C	D	E	F	G
3	t	t_прих	L	t_обсл	T-t	delta_t	Комментарии
8	23	6	0	8	17	6	
9	29	6	1	2	11	2	
10	31	4	0	8	9	4	3-й транзакт ушел из модели обслуженным
11	35	6	1	4	5	4	
12	39	2	0	10	1	1	4-й транзакт ушел из модели обслуженным
13	40	1	0	9	0	0	

Рисунок 31 – Иллюстрация смены состояний модели СМО

В момент времени $t = 29$ наблюдается увеличение длины очереди на 1. Это связано с тем, что в момент прихода четвертого транзакта канал был занят. Такую же ситуацию застает пятый транзакт при $t = 35$ и задерживается в очереди.

На предпоследнем шаге приращение времени равно 1, что на последнем шаге приводит к завершению моделирования. Итоги моделирования такие:

- в модель всего вошло 5 транзактов;
- всего было обслужено 4 транзакта;
- в очереди кратковременную задержку имели 2 транзакта;
- 5-й транзакт не успел к концу моделирования завершить свою обработку в канале;
- 6-й транзакт не успел войти в модель (остался в источнике).

Примеры имитации двух других моделей приведены в файле: <https://disk.yandex.ru/i/2bQjcsxLfFLBAw>

Контрольная работа

Тест размещен на авторском сайте [4, страница: <https://altaev-aa.narod.ru/km3/qs.html>]. Имеются все необходимые для запуска программы указания.

Условные обозначения типов СМО

В 1953 г. британский статистик Д.Д. Кендалл предложил условное описание типов СМО в виде $A/S/c/K$ [6], где A – используемый закон распределения СВ для вычисления времени прибытия транзактов в модель; S – закон распределения СВ для вычисления времени обслуживания; c – количество каналов обслуживания; K – вместимость очереди. Указанные параметры означают:

- A описывает дисциплину прибытия транзактов. Вместо A следует указать либо символ G (General, общее распределение), либо M (Markovian, Марковское, экспоненциальное, распределение);
- S указывает на используемый закон распределения СВ для вычисления времени обслуживания в канале. Принимает такие же значения, что и A ;
- c – количество используемых каналов;
- K – емкость очереди. Если очередь неограниченной длины, то K либо не указывается, либо записывается как ∞ .

СМО, моделирование которого было рассмотрено выше, в нотации Кендалла можно записать как $G/G/1$, так как использованный в данном примере равномерный закон относится к общему типу, был один канал и очередь имела неограниченную длину.

МОДЕЛИРОВАНИЕ С ЯИМ GPSS

Язык имитационного моделирования (ЯИМ) GPSS (General Purpose System Simulation, система моделирования общего назначения) был разработан в 1968 г. В пособии используется GPSS World Student (<https://altaev-aa.narod.ru/km3/GPSS.zip>). По этой версии языка выпущено несколько достаточно подробных пособий, таких как [7, 8, 9]. В данном пособии не ставится целью обзор всех возможностей GPSS, а рассматриваются лишь самые необходимые из них, но позволяющие обеспечить построение модели для проведения с ними имитационных экспериментов.

Когда разрабатывалась первая версия GPSS (1968), данные в память компьютера вводились с помощью перфокарт и, соответственно, инструкции GPSS назывались картами [3]. Всего было предусмотрено три типа карт:

- карты описания;
- блоки;
- карты управления.

Карты описания располагаются вначале программы и с их помощью назначаются требуемые характеристики таблицам, многоканальным устройствам и описываются переменные, функции и т. д. Блоки позволяют построить саму модель и именно по ним перемещаются транзакты. Карты управления позволяют запустить моделирование, а также указать необходимые параметры режимов моделирования. В современной версии GPSS (GPSS World) карты управления не используются, а их функции возложены на соответствующие пункты меню GUI программы.

Карта содержит следующие элементы:

- метка;
- название карты;
- поля;
- комментарии.

Указанные элементы должны отделяться друг от друга либо пробелом, либо знаком табуляции.

Метка выполняет две функции:

- при использовании в карте описания – это имя таблицы, многоканального устройства, переменной или функции;
- при применении в блоке – это метка перехода из другого места модели на этот блок.

Если метка для данной карты не предусмотрена, то достаточно вместо метки ввести пробел, либо символ табуляции.

После названия карты через запятую перечисляются поля, определяющие функциональность данной карты.

Общий синтаксис карты приведен на рисунке 32.

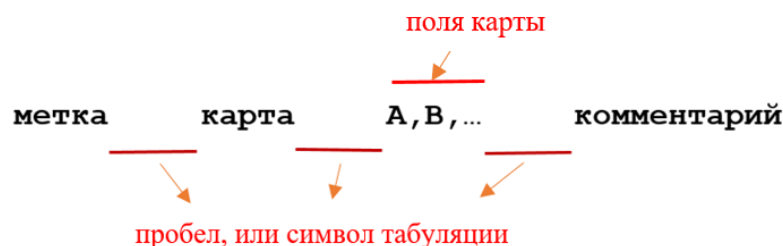


Рисунок 32 – Синтаксис карты (команды) GPSS

В полях допускается использование выражений, но их описание должно быть дано внутри круглых скобок.

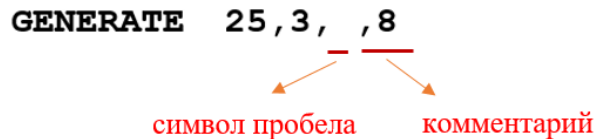
На рисунке 33 приведен пример блока GENERATE.

GENERATE 25,3,,8

Рисунок 33 – Пример карты

В данном примере отсутствуют метка и комментарий. В поле А введено число 26, в поле В – 3, значение поля С не указано, в поле D записано 8. Внутри полей пробелы недопустимы. На рисунке 34 дан пример карты, содержащей ошибку. Пробел завершает область полей и начинает текст комментария.

GENERATE 25,3, ,8



символ пробела комментарий

Рисунок 34 – Наличие пробела в полях карты

В следующем примере (рис. 35) в блоке GENERATE использовано лишь одно поле А и в нем само выражение в виде вызова функции заключено в скобки.

GENERATE (EXPONENTIAL(1,0,2.6))

Рисунок 35 – Применение выражения в поле карты

Стандартные числовые атрибуты (СЧА)

Основная цель имитационного моделирования – это сбор необходимой статистики по исследуемой реальной системе. GPSS для этой цели использует стандартные числовые атрибуты (далее – СЧА), к которым можно обратиться двумя способами:

- <имя СЧА><номер объекта> – если объект нумерованный;
- <имя СЧА>\${имя объекта} – если объект именованный.

Примеры (прил. 3):

- Q5 – здесь Q – имя СЧА, позволяющее обратиться к очереди (Queue), 5 – номер очереди;
- Q\$nakopitel – СЧА очереди с именем nakopitel.

Есть особая разновидность СЧА – логическая, принимающая значения 0 (false) и 1 (true).

Имеется группа системных СЧА, относящихся ко всей модели в целом (прил. 3):

- RN_j – служит для вызова встроенного в GPSS генератора случайных чисел, $j = \overline{1, 24}$;
- C1 – модельное время;
- TG1 – счетчик завершения моделирования.
-

Транзакты

Транзакты могут отличаться друг от друга:

- своими приоритетами, их значения определяются по СЧА PR;
- параметрами. СЧА P позволяет узнать значение нужного параметра.

Время, прошедшее с момента входа транзакта в модель, называется транзитным и определяется по СЧА M1.

Списки транзактов

Во время моделирования транзакты хранятся в списках:

- список текущих событий, когда транзакту предоставлена в данный момент возможность перемещения по модели. Все транзакты по очереди извлекаются из указанного списка и будут перемещаться по модели до тех пор, пока не попадут в какой-нибудь другой список. После того как весь список будет обнулен, модельное время будет изменено до времени наступления будущего события и список текущих событий вновь пополнится;
 - список будущих событий – это транзакты, находящиеся, например, в источнике или канале. Таким образом, этот список на самом деле является списком основных событий;
 - список задержки. Здесь находятся транзакты, оказавшиеся, например, в очередях.
- Имеются еще списки прерываний, пользователя и т. д.

Блоки

Как выше упоминалось, с помощью блоков строится модель. Перемещение транзактов по блокам позволяет имитировать поведение самой системы. Вход транзакта в блок приводит к наступлению таких событий, как, например, вход в очередь, выход из очереди, вход в канал, выход из канала и т. д. Блоки относятся к двум типам:

- неблокирующие – не препятствующие входу транзакта в него (их большинство);
- блокирующие – отказывающие транзакту во входе при наличии запрещающего условия.

В дальнейшем, если не будет указано к какому типу относится рассматриваемый блок, то это будет означать, что он является неблокирующим.

Для прогнозирования времени наступления основных событий предусмотрены два блока: GENERATE и ADVANCE. В данных блоках для расчета времени следует использовать функции, перечень которых приведен в [7, п. «1.5.5. Библиотечные генераторы случайных чисел»]. Рассмотрим лишь три из них:

- DUNIFORM(Stream,Min,Max) – генерирует случайные целые числа из диапазона [Min, Max], Stream – номер встроенного в GPSS одного из 24 генераторов СВ.

Примечание. Нумерация генераторов начинается с 1, т. е. Stream должен быть выбран из интервала [1; 24].

- EXPONENTIAL(Stream,Locate,Scale) – генерирует вещественные числа с применением экспоненциального закона распределения СВ, Stream как и в предыдущем случае – номер генератора, Locate – величина сдвига (добавляется ко всем сгенерированным числам), Scale – среднее значение времени. Пример использования данной функции приведен на рисунке 35.

- NORMAL(Stream,Mean,StdDev) – генерирует вещественные числа с применением нормального закона распределения СВ, Stream – номер генератора, Mean – среднее значение, StdDev – стандартное отклонение.

При перечислении названий функций и их аргументов пробелы намеренно удалены, так как прямое предназначение пробелов – это отделение элементов карты друг от друга: метки от имени карты, имени от полей и полей от комментария.

Блок GENERATE

Выполняет роль источника транзактов (см. рис. 18). Имеет поля:

- A – указывается либо выражение в скобках для вызова одной из функций генерации СВ, либо числовая константа, равная среднему времени прихода транзактов. Числовая константа может быть заменена СЧА, значение которого и станет средним временем прихода;

- В – если в поле А указана константа, то указывается полуразмах от среднего. Таким образом, будет реализована генерация чисел на основе равномерного закона распределения СВ. Если в поле А записано выражение, то поле В не заполняется;
- С – время прихода в модель первого транзакта. Если поле не заполнено, то время прихода первого транзакта задается рандомно на общих правилах;
- D – задает общее количество транзактов. Если поле не заполнено, то генерируется бесконечное число транзактов;
- E – задает значение приоритета создаваемым транзактам. Если E не заполнено, то транзакты получают нулевой приоритет.

Примеры:

- GENERATE 6 – задает детерминированный процесс прихода транзактов (транзакты поступают в модель с разницей в 6 единиц времени);
- GENERATE 6, 2 – времена прихода транзактов выбираются случайным образом из интервала [4, 8];
- Блок, приведенный на рисунке 35, генерирует транзакты с применением экспоненциального закона распределения СВ со средним временем 2,6;
- GENERATE , , , 1 – как только моделирование начнется в модель войдет единственный транзакт.

Блок ADVANCE

Данный блок задерживает внутри себя транзакт на некоторое время. Для расчета времени задержки используются два поля: А и В, правила заполнения которых такие же, как и у блока GENERATE.

Блоки SEIZE и RELEASE

Блок ADVANCE сам по себе не умеет имитировать канал (см. рис. 11), так как не может совершать события «перевести канал в состояние занято» и «перевести канал в состояние свободно». За вызов этих событий отвечают блоки SEIZE и RELEASE. Оба блока имеют лишь одно поле – поле А, в котором следует указывать либо номер канала, либо его имя. Транзакт, заходя в блок SEIZE, переводит канал, имя или номер которого указано в поле А, в состояние «занято». Блок RELEASE при входе в него транзакта, напротив, переводит канал в состояние «свободно». SEIZE является блокирующим блоком, так как транзакт не сможет войти, если канал уже находится в состоянии «занято», т. е. в канале уже есть другой транзакт.

Три последовательно расположенных блока создают канал 1:

```
SEIZE      1
ADVANCE    4, 2
RELEASE    1
```

Транзакт, входя в первый блок, переводит канал в состояние «занято» и перемещается во второй блок, в котором задерживается на некоторое случайное время (от 2 до 6). В течение всего времени обслуживания другие транзакты не могут войти в блок SEIZE. Как только время обслуживания завершится, транзакт покидает ADVANCE, проходит через блок RELEASE, возвращая каналу 1 состояние «свободно», и уходит в другие блоки модели.

Атрибуты, присущие каналу, содержат в своем имени букву F (Facility, устройство) (см. прил. 3).

Блоки QUEUE и DEPART

Данные блоки позволяют собирать статистику по очереди. Оба имеют по два поля:

- А – имя или номер очереди;
- В – приращение длины очереди. Если поле В не указано, то по умолчанию приращение равно 1.

Пример модели с очередью и каналом:

```

QUEUE      9
SEIZE      pribor
DEPART     9
ADVANCE    (NORMAL (1, 8, 2))
RELEASE    pribor

```

В данном примере время обслуживания транзакта вычисляется по нормальному закону распределения СВ, среднее время принято равным 8, стандартное отклонение равно 2, для получения случайных чисел используется 1-й генератор. В начале моделирования в канале и в очереди нет транзактов. Первый транзакт, проходя через очередь 9, увеличивает ее длину на 1. Далее он проходит через SEIZE, занимая pribor. Затем он, пройдя через DEPART, освобождает очередь 9 (длина очереди вновь становится равной 0) и входит в блок ADVANCE, где и будет задержан. Другие, идущие следом транзакты, поступают в QUEUE и остаются в нем, так как SEIZE блокирует их дальнейшее перемещение.

Атрибуты очереди содержат в своем имени букву Q (Queue, очередь) (см. прил. 3). Примеры некоторых из них:

- Q – текущая длина очереди. Если Q9 равно 5, то это означает, что на данный момент в очереди 9 находится 5 транзактов;
- QZ – (zero, ноль), число транзактов, пробывших в очереди с нулевым временем, т. е. вошедших в очередь и сразу ее покинувших.

Блок TERMINATE. Счетчик TG1

Тогда как блок GENERATE создает транзакты, блок TERMINATE ответственен за их удаление из модели. В поле A данного блока записывается декремент счетчика завершения TG1. Указанный счетчик отвечает за завершение моделирования. Начальное значение TG1 задается с помощью управляющей карты START (как было указано выше, функции управляющих карт перенесены в меню программы GPSS World). Для запуска модели необходимо выполнить команды:

- Command/Create Simulation;
- Command/START.

Далее в появившемся окне (рис. 36) следует изменить поле карты START.

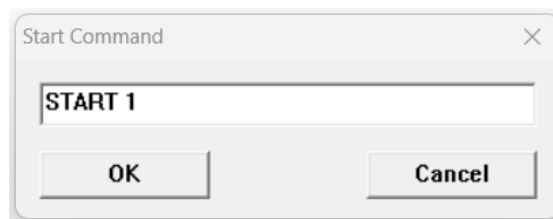


Рисунок 36 – Окно ввода начального значения счетчика TG1

Каждый транзакт, входящий в блок TERMINATE, будет убавлять значение счетчика на величину декремента, указанного в поле A этого блока. Моделирование завершится тогда, когда TG1 станет нулевым или примет отрицательное значение. Пусть в модели есть единственный блок

```

TERMINATE 1
а в карте указали
START      10

```

В таком случае моделирование завершится, когда в TERMINATE войдет ровно 10 транзактов. Другой пример:

```

TERMINATE 2
START      7

```

В данном случае для завершения моделирования понадобится 4 транзакта.

Реализация модели СМО в GPSS

Модель создается командой File/New, Model. Блоки удобнее вводить в модель через окно Edit/Insert GPSS Blocks

Для моделирования в GPSS выберем задание, рассмотренное в теме «Моделирование простейшей СМО», с исходными данными в виде: $\tau_{\text{прих}} = 7 \pm 2$, $\tau_{\text{обсл}} = 9 \pm 1$, $T = 40$. Модель данной задачи будет иметь вид (<https://altaev-aa.narod.ru/km3/book/model01.gps>):

```
GENERATE 7,2
QUEUE    nakopitel
SEIZE    canal
DEPART   nakopitel
ADVANCE  9,1
RELEASE  canal
TERMINATE
GENERATE 40
TERMINATE 1
```

Модель состоит из двух сегментов, каждый из которых начинается с блока GENERATE и завершается блоком TERMINATE. В первом сегменте реализована сама СМО, второй сегмент создан лишь для контроля за временем моделирования. Если в карте START указать TG1 равным 1, то получим отчет в виде:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	5	0	0
	2	QUEUE	5	1	0
	3	SEIZE	4	0	0
	4	DEPART	4	0	0
	5	ADVANCE	4	1	0
	6	RELEASE	3	0	0
	7	TERMINATE	3	0	0
	8	GENERATE	1	0	0
	9	TERMINATE	1	0	0

Колонка «ENTRY COUNT» содержит число входов в блоки. Так, например, в первый блок (GENERATE) вошло 5 транзактов, во второй также 5 и т. д. Колонка «CURRENT COUNT» показывает число транзактов, оставшихся в блоках, на момент завершения моделирования. Из отчета видно, что при завершении моделирования в очереди оставался один транзакт и еще один транзакт находился на обслуживании в канале. Рассмотрим два последних блока. В самом начале моделирования в блок GENERATE был помещен один транзакт и для него было определено время пребывания в этом блоке, равное 40. По истечении данного времени транзакт покидает GENERATE, перемещается в TERMINATE, убавляя при этом счетчик TG1 до нуля, и моделирование завершается. В первом сегменте блок TERMINATE не имел заполненного поля A и транзакты, входившие в данный блок, не оказывали никакого влияния на счетчик завершения.

Из всех 9 блоков, составляющих данную модель, лишь один блок, SEIZE, является блокирующим. Следующее замечание: блок GENERATE не может быть транзитным блоком, т. е. с вышележащего блока на него не должны переходить транзакты. Он является источником транзактов и не должен принимать «чужие» транзакты. Это означает, что над ним должен находиться либо блок TERMINATE, удаляющий транзакты, либо блок TRANSFER, передающий транзакты на другие места модели.

Само условие завершения моделирования в зависимости от решаемой задачи может быть двух типов:

- по истечении модельного времени (как в рассмотренном выше примере);
- по числу обработанных или потерянных транзактов.

Проведение экспериментов с моделью model01.gps

Задание:

- Поменять код модели model01.gps, введя в нее свою фамилию. Фамилию можно ввести в метку или в имя какого-либо объекта.
- Изучить зависимость характеристик очереди и канала от времени обслуживания транзактов. Время обслуживания транзактов в экспериментах принимать равным 7, 8, 9, 10 и 11, полуразмах от среднего принять равным 1.
- Результаты экспериментов занести в таблицу отчета.
- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- Отчет должен завершаться выводом о результатах эксперимента.

Исходные данные:

- шаблон отчета по модели model01: <https://disk.yandex.ru/i/kK4MGkA2HnS20g>;
- видео: <https://disk.yandex.ru/i/u2D7a15FQpbHtA>.

Пример завершения моделирования по числу обработанных транзактов

Рассмотрим второй вариант. Пусть необходимо обработать 100 транзактов. Время моделирования T не задано в данном случае. Модель будет иметь следующий вид (<https://altaev-aa.narod.ru/km3/book/model02.gps>):

```
GENERATE 7,2
QUEUE    nakopitel
SEIZE    canal
DEPART   nakopitel
ADVANCE  9,1
RELEASE  canal
TERMINATE 1
```

В управляющей карте следует указать количество обслуженных транзактов:

```
START 100
```

Отчет имеет вид:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	134	0	0
	2	QUEUE	134	33	0
	3	SEIZE	101	1	0
	4	DEPART	100	0	0
	5	ADVANCE	100	0	0
	6	RELEASE	100	0	0
	7	TERMINATE	100	0	0

На момент завершения моделирования в канале находился на обслуживании один транзакт, в очереди оставалось 33 транзакта, всего было обслужено 100 транзактов.

Проведение экспериментов с моделью model02.gps

Задание:

- Поменять код модели model02.gps, введя в нее свою фамилию.
- Изучить зависимость характеристик очереди (средней длины очереди – AVE.CONT. и среднего времени пребывания одного транзакта в очереди – AVE.TIME) от числа обработанных транзактов.

- Провести эксперименты, задавая значения числа обработанных транзактов 80, 90, 100, 110 и 120, и полученные результаты занести в таблицу отчета.
 - В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
 - В отчете должен быть вывод о результатах эксперимента.
- Шаблон отчета: <https://disk.yandex.ru/i/i2sZMkCYqUXOkQ>

Памяти

Памяти являются многоканальными устройствами (МКУ), характерным свойством которой является ее емкость. Так, например, на у МКУ, изображенной рисунке 37, емкость равна 3. Канал, который мы рассматривали до этого (см. рис. 11), называется одноканальным устройством (ОКУ).

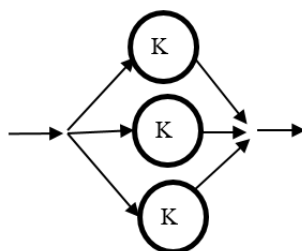


Рисунок 37 – Память

Для применения памяти в модели используется карта описания:

метка STORAGE A

и два блока

ENTER A, B

LEAVE A, B

У карты в области метки записывается имя памяти (использование номера в метках не допускается), в поле A – ее емкость. У блоков ENTER и LEAVE в поле A указывается имя памяти, в поле B – число мест, занимаемых или освобождаемых в памяти входящим в них транзактом. Если B не указано, то по умолчанию значение поля берется равным 1. Пример:

Strg STORAGE 5

...

ENTER Strg

ADVANCE 10, 2

LEAVE Strg

В данном примере одновременно в блоке ADVANCE может находиться 5 транзактов. Часто память используется для моделирования очереди ограниченной длины. Пример такой модели будет рассмотрен ниже.

СЧА памяти содержат в своем имени букву S (от слова Storage): S (число занятых мест в памяти), R (число свободных мест), SE (логическое СЧА, E – Empty, память пуста) и т. д. (см. прил. 3). Если, например, в рассмотренном выше примере величина S\$Strg будет равна 3, то это означает, что на данный момент в памяти находится три транзакта. При этом атрибут R\$Strg будет равен 2, а SE\$Strg будет иметь значение 0 (false, т. е. память не пуста).

Блок ENTER является блокирующим, так как, если память будет полностью заполненной, то транзакт не сможет войти в данный блок.

Логические переключатели

Логические переключатели (второе название – ключи) могут принимать два значения:

- SET – установлен;
- RESET – сброшен.

Блок, отвечающий за изменение состояния ключа, имеет формат:

```
LOGIC X A
```

где X – логический оператор, а в поле A указывается имя или номер логического переключателя, значение которого меняется. Оператор X имеет три варианта:

- S – переключатель переводится в состояние SET;
- R – переключатель переводится в RESET;
- I – значение переключателя меняется на противоположное (Inverse, обратный).

Пример:

```
LOGIC R logicKey
```

Транзакт, войдя в данный блок, сбрасывает переключатель logicKey.

Переключатель имеет единственный логический СЧА – LS, который принимает значение 1 (true), если установлено состояние SET. Если транзакт войдет в блок LOGIC, пример которого приведен выше, то LS\$logicKey примет значение 0.

Блоки, изменяющие характеристики транзактов

К характеристикам транзактов, как говорилось выше, относятся их приоритеты и параметры. Блок, изменяющий приоритет, имеет формат

```
PRIORITY A
```

В поле A указывается значение присваиваемого приоритета (значение не должно быть выше 127). По умолчанию входящие в модель транзакты имеют нулевой приоритет.

Параметры транзактов изменяются с помощью блока ASSIGN, который имеет три режима работы:

- ASSIGN A, B

В поле A указывается имя или номер изменяемого параметра, в поле B – значение, записываемое в параметр;

- ASSIGN A+, B

В этом режиме к значению параметра A добавляется величина B;

- ASSIGN A-, B

В этом режиме значение параметра A убавляется на величину B.

Примеры:

```
ASSIGN count, 0
```

В параметр count заносится 0.

```
ASSIGN count+, 1
```

Параметр count увеличивается на 1.

Блок TRANSFER

Как правило транзакты по блокам перемещаются последовательно с вышележащих блоков к нижележащим. Но такой порядок их движения можно поменять, используя блоки ветвления. Первый из таких блоков – TRANSFER [7, с. 57] имеет следующий формат:

```
TRANSFER A, B, C, D
```

В поле A указывается условие перехода, в остальных – метки переходов. Рассмотрим наиболее популярные варианты данного блока. Первый из них позволяет без проверки какого-либо условия перенаправлять поток транзактов в другое место модели. Пример:

```
TRANSFER , anyLabel
```

В данном случае поле A не заполнено, транзакты, вошедшие в блок, перенаправляются на другой блок, имеющий метку anyLabel.

Второй вариант предусматривает работу блока в вероятностном режиме. В таком случае в поле A указывается вероятность перехода транзакта на метку, указанную в поле C. Например, блок

TRANSFER .4, firstLabel, secondLabel

40 % транзактов, вошедших в него, будет перенаправлять на блок с меткой secondLabel, остальные 60 % – с меткой firstLabel.

Поле В может отсутствовать. В этом примере:

TRANSFER .5, , anyLabel

50 % транзактов перенаправляются на блок с меткой anyLabel, остальные пропускаются на нижележащий блок.

Описание других вариантов блока TRANSFER дано в [7].

Блок TEST

Данный блок работает в двух режимах:

- задержки транзактов;
- ветвления маршрутов транзактов.

Имеет формат:

TEST X A, B, C

В поле С указывается метка перехода. Если поле не заполнено, то TEST работает в первом режиме (в режиме задержки), в противном случае в режиме ветвления. Оператор X указывает на операцию сравнения, которую необходимо выполнить над величинами, указанными в полях А и В. В полях А и В могут быть записаны числовые константы, СЧА и выражения.

Варианты записи оператора X следующие:

- l – (less, меньше, <), значение А меньше чем В;
- e – (equals, равно), $A = B$;
- le – $A \leq B$;
- g – (greater, больше), $A > B$;
- ge – $A \geq B$;
- ne – $A \neq B$.

Если условие выполнится, то транзакт проходит на нижележащий блок, в противном случае он либо останется в блоке TEST (если он работает в режиме задержки, т. е. поле С не указано), либо уйдет на блок, чья метка указана в поле С.

Пример:

TEST l Q\$nakopitel, 5, anyLabel

В данном примере транзакт пройдет на нижележащий блок, если текущая длина очереди nakopitel будет меньше 5, иначе транзакт уйдет по метке anyLabel.

Блок GATE

Данный блок позволяет проверить состояние объекта и принять решение о перемещении транзакта. Как и TEST может работать в двух режимах: задержки и разветвления. Объекты, которые проверяет GATE, могут быть трех типов: одноканальное устройство (МКУ), многоканальное устройство (МКУ) и логический переключатель. Формат блока имеет вид:

GATE X A, B

В поле А указывается имя или номер проверяемого устройства, в В – метка перехода. Если поле В не заполнено, т. е. метка не указана, то блок будет работать в режиме задержки, а при наличии метки – в режиме пересылки. Задержка или пересылка будет выполнена, если проверяемое условие не будет выполнено.

Варианты записи логического оператора X следующие:

1. если проверяется ОКУ

- u – канал используется (use);
- nu – не используется;
- i – находится в состоянии прерывания (invade, захватывать);

- ni – не находится в состоянии прерывания;
- 2. если проверяется МКУ
 - se – память пуста (storage empty);
 - sne – память не пуста;
 - sf – память заполнена (storage full);
 - snf – память не заполнена;
- 3. если проверяется переключатель
 - lr – переключатель сброшен (logic reset);
 - ls – переключатель установлен (logic set).

Примеры:

```
GATE nu canal, anyLabel
```

Если ОКУ canal не используется, транзакт перейдет в нижележащий блок, иначе переместится по метке anyLabel.

```
GATE snf buffer
```

Если память buffer не заполнена, то транзакт перейдет в нижележащий блок, иначе транзакт останется в блоке GATE.

Блок LOOP

Организует повторное прохождение транзактом определенного участка модели. Имеет два поля. В поле А задается величина проходов (как правило, указывается параметр Р транзакта), в поле В метка, с которой начинается сам цикл.

Блок SAVEVALUE

В отличие от параметра Р, который может хранить величины, присущие только одному транзакту, у самой модели есть общие переменные. Такие переменные можно изменять с помощью блока SAVEVALUE (сохраняемая величина). Режимы ее работы схожи с режимами блока ASSIGN, т. е. предусмотрены режимы замены, добавления и убавления. СЧА таких переменных обозначается через Х (см. прил. 3).

Примеры:

```
SAVEVALUE 1, 3
```

Транзакт, войдя в данный блок, изменит значение атрибута Х1 на 3.

```
SAVEVALUE anyValue-, 1
```

Атрибут Х\$anyValue будет уменьшен на 1.

Примеры моделей с разветвлением маршрутов

Рассмотрим реализацию СМО с потерями (см. рис. 15). Установим емкость накопителя L^H , равную 5 транзактам, интенсивность прихода транзактов $\lambda_{\text{прих}} = 6$ транзактов/час, интенсивность обслуживания $\lambda_{\text{обсл}} = 4$ транзакта/час и примем общее время моделирования, равное 8 часам. Считая, что время измеряется в минутах, найдем средние времена прихода и обслуживания: $\bar{t}_{\text{прих}} = 60/6 = 10$ минут, $\bar{t}_{\text{обсл}} = 60/4 = 15$ минут. Для генерации времени наступления основных событий будем применять экспоненциальный закон распределения СВ.

В первой версии модели для проверки длины очереди применим блок TEST, во второй версии – блок GATE.

Код первой версии (<https://altaev-aa.narod.ru/km3/book/model03.gps>) имеет вид:

```
GENERATE (EXPONENTIAL(1, 0, 10))
TEST L Q$nakopitel, 5, labelLosses
QUEUE nakopitel
SEIZE canal
DEPART nakopitel
```

```

ADVANCE (EXPONENTIAL(1,0,15))
RELEASE canal
SAVEVALUE served+,1 ; обслужено заявок
TERMINATE
labelLosses SAVEVALUE losses+,1 ; потеряно заявок
TERMINATE
GENERATE (8#60) ; в GPSS символ # - операция умножения
TERMINATE 1

```

Пояснение: в блоке TEST проверяется условие $Q\$nakopitel < 5$.

Прогон модели с картой

START 1

выводит следующий отчет:

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE		44		0	0	
	2	TEST		44		0	0	
	3	QUEUE		34		5	0	
	4	SEIZE		29		0	0	
	5	DEPART		29		0	0	
	6	ADVANCE		29		1	0	
	7	RELEASE		28		0	0	
	8	SAVEVALUE		28		0	0	
	9	TERMINATE		28		0	0	
LABELLOSSES	10	SAVEVALUE		10		0	0	
	11	TERMINATE		10		0	0	
	12	GENERATE		1		0	0	
	13	TERMINATE		1		0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
CANAL	29	0.983	16.277	1	39	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)
RETRY							
NAKOPITEL	5	5	34	4	2.730	38.545	43.684

SAVEVALUE	RETRY	VALUE
SERVED	0	28.000
LOSSES	0	10.000

Как видно из отчета, за 8 часов было обслужено 28 транзактов, не попавших в очередь вследствие чего потерянных 10 транзактов.

Проведение экспериментов с моделью model03.gps и обработка результатов эксперимента

Задание:

- Поменять код модели model03.gps, введя в нее свою фамилию.
- Изучить зависимость числа обработанных транзактов и потерянных транзактов от максимальной длины очереди.
- Провести эксперименты, задавая значения максимальной длины очереди 3, 4, 5, 6 и 7, и полученные результаты занести в таблицу отчета.
- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: https://disk.yandex.ru/i/N1PxbkvzAB_opA

Вторая модель (<https://altaev-aa.narod.ru/km3/book/model04.gps>) использует в качестве накопителя память (МКУ) с емкостью 5 транзактов:

```
nakopitel STORAGE 5
GENERATE (EXPONENTIAL(1,0,10))
GATE SNF nakopitel,labelLosses ; SNF - память не заполнена
ENTER nakopitel
SEIZE canal
LEAVE nakopitel
ADVANCE (EXPONENTIAL(1,0,12))
RELEASE canal
SAVEVALUE served+,1
TERMINATE
labelLosses SAVEVALUE losses+,1
TERMINATE
GENERATE (8#60)
TERMINATE 1
```

Прогон этой версии модели выводит тот же результат, что и первая версия. Дополним работу с этой моделью обработкой результатов экспериментов. Например, мы желаем установить, существует ли связь между числом потерянных транзактов и емкостью накопителя. При этом потерянные транзакты в статистике не будут учтены.

Проведем эксперименты, задавая объем памяти `nakopitel`, равным 3, 4, 5, 6 и 7. Результаты эксперимента будем вносить в Excel таблицу (рис. 38).

	A
	объем памяти nakopitel
1	
2	3
3	4
4	5
5	6
6	7

Рисунок 38 – Таблица для фиксирования результатов эксперимента

Задавая в модели последовательно значения емкости памяти (карта `STORAGE`) в таблицу вносим соответствующие количества потерянных транзактов (рис. 39).

	A	B
	объем памяти nakopitel	количество потерянных транзактов
1		
2	3	15
3	4	12
4	5	10
5	6	7
6	7	6

Рисунок 39 – Результаты проведенных экспериментов

Связь между задаваемыми и наблюдаемыми величинами является функцией. В нашем примере аргументом функции является емкость памяти, а самой функцией – число потерянных транзактов. Сама функция может быть выражена:

- в табличной форме;
- в виде графика;
- в виде аналитической зависимости;

В рисунке 39 зависимость между числом потерянных транзактов и емкостью накопителя отражена в виде таблицы. Отобразим эту зависимость на графике. Для этого следует выделить таблицу вместе с ее заголовками и выполнить команду Вставка/Вставить точечную (X,Y)/Точечная. В результате на рабочем листе Excel появится диаграмма (рис. 40).

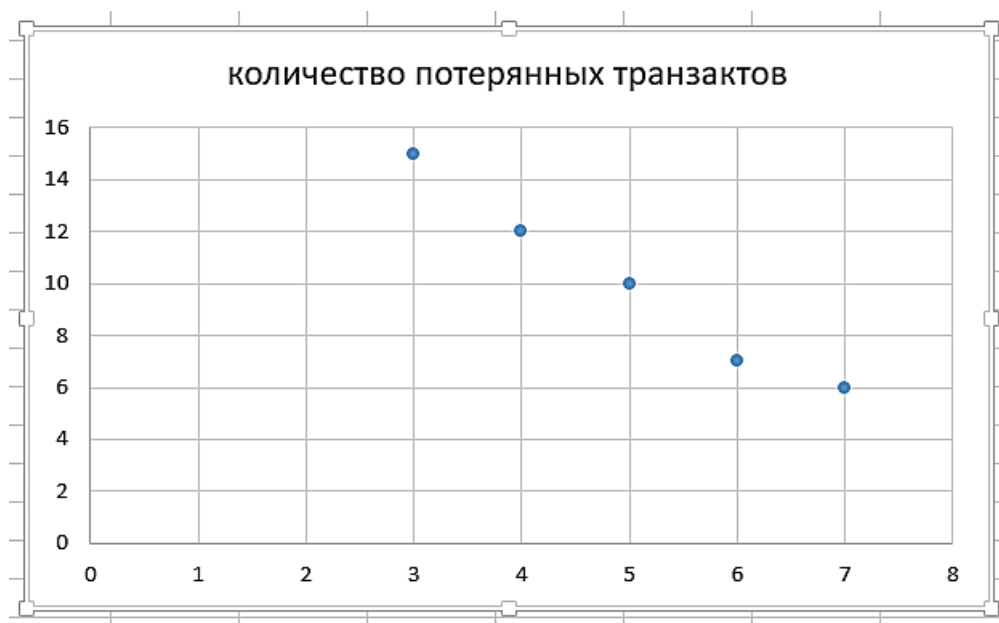


Рисунок 40 – Графическое представление зависимости числа потерянных транзактов от емкости накопителя

Далее выясняем, какая аналитическая зависимость связывает обе величины между собой. Для чего следует щелкнуть правой кнопкой мыши над одной из точек диаграммы и в появившемся контекстном меню выбрать команду «Добавить линию тренда ...» Сбоку появляется панель, в которой можно выбрать подходящее уравнение (рис. 41). Такие уравнения называются уравнениями регрессии. Excel позволяет подобрать уравнения лишь нескольких типов: экспоненциальное, линейное, логарифмическое, полиномиальное, степенное и линейную фильтрацию. Следует опытным путем определить какое из них наилучшим образом описывает зависимость между числом потерянных транзактов и емкостью накопителя. Критерием отбора оптимального уравнения является величина достоверности аппроксимации R^2 . Чем ближе данный параметр к единице, тем лучше выбранное уравнение подходит для описания зависимости между интересующими нас величинами (числа потерянных транзактов от емкости накопителя). Следует вывести на рабочий лист Excel графики всех перечисленных выше функций и сравнить между собой значения их величин R^2 (<https://altaeva-a.narod.ru/km3/book/ehksperiment.xlsx>). Результаты отображены в таблице 10. Линейная фильтрация не рассматривалась. Результаты показывают, что для описания результатов эксперимента более всего подходит полиномиальное уравнение второй степени ($R^2 = 0,9915$).

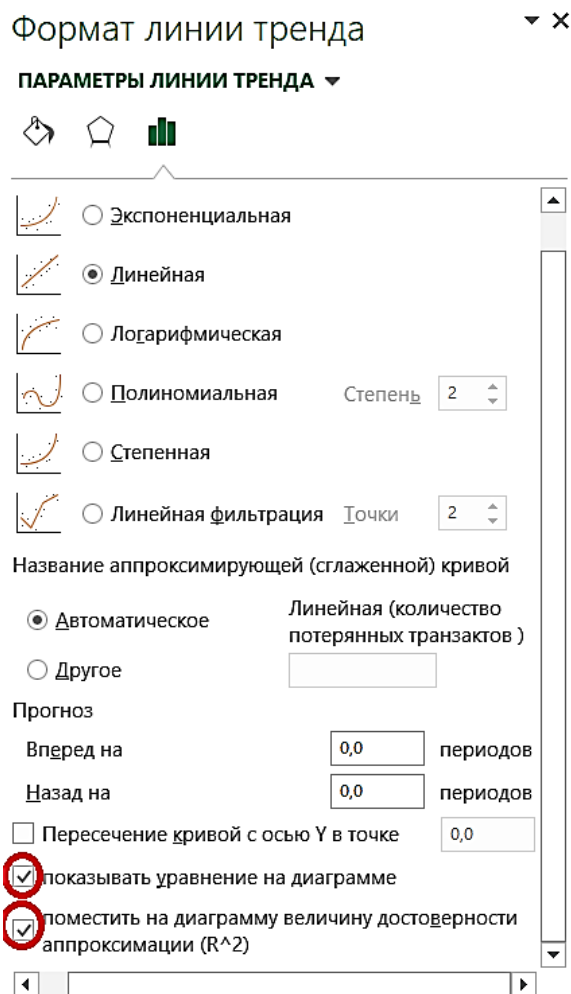


Рисунок 41 – Выбор уравнения регрессии

Таблица 10

Тип уравнения	Уравнение регрессии	R ²
Экспоненциальное	$y = 30,952e^{-0,237x}$	0,9865
Линейное	$y = -2,3x + 21,5$	0,9796
Логарифмическое	$y = -10,92\ln(x) + 27,112$	0,9899
Полиномиальное	$y = 0,2143x^2 - 4,4429x + 26,429$	0,9915
Степенное	$y = 53,719x^{-1,109}$	0,9662

Проведение экспериментов с моделью model04.gps

Задание:

- Поменять код модели model04.gps, введя в нее свою фамилию.
- Изучить зависимость числа обработанных транзактов от емкости памяти.
- Провести эксперименты, задавая значения максимальной емкости памяти 3, 4, 5, 6 и 7, и полученные результаты занести в таблицу отчета.
- Провести обработку результатов проведенных экспериментов и подобрать подходящее уравнение регрессии. Предупреждение: возможен вариант, когда не наблюдается никакой зависимости между изменяемой и наблюдаемой величинами.
- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: https://disk.yandex.ru/i/NdXeBDpBC_pNvg

Модель СМО с обслуживанием транзактов, имеющих относительные приоритеты

Транзакты, имеющие относительные приоритеты, имеют преимущество над бесприоритетными лишь при прохождении ими очереди (см. рис. 16). Изменим предыдущее задание, добавив к нему условие, что 25 % входящих транзактов имеет повышенный приоритет.

Задание немного изменено: интенсивность обработки принята в 5 транзактов в час ($\bar{\tau}_{\text{осбл}} = 12$), длина очереди принята бесконечной. Код модели (<https://altaev-aa.narod.ru/km3/book/model05.gps>) имеет вид:

```

GENERATE (EXPONENTIAL(1,0,10))
TRANSFER .75,,detour ; отделение четверти транзактов и
PRIORITY 1 ; повышение им приоритета
detour QUEUE nakopitel
SEIZE canal
DEPART nakopitel
ADVANCE (EXPONENTIAL(1,0,12))
RELEASE canal
TEST E PR,1,noPrior ; отделение приоритетных транзактов
SAVEVALUE servedPrior+,1 ; обслужено приоритетных транзактов
TRANSFER ,term
noPrior SAVEVALUE servedNoPrior+,1 ; бесприоритетные транзакты
term TERMINATE
GENERATE (8#60)
TERMINATE 1
    
```

В модели для определения приоритета активного в данный момент транзакта используется СЧА PR. Отчет имеет вид:

LABEL	LOC	BLOCK TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY	
	1	GENERATE	38		0	0		
	2	TRANSFER	38		0	0		
	3	PRIORITY	7		0	0		
DETOUR	4	QUEUE	38		6	0		
	5	SEIZE	32		0	0		
	6	DEPART	32		0	0		
	7	ADVANCE	32		1	0		
	8	RELEASE	31		0	0		
	9	TEST	31		0	0		
	10	SAVEVALUE	7		0	0		
	11	TRANSFER	7		0	0		
NOPRIOR	12	SAVEVALUE	24		0	0		
TERM	13	TERMINATE	31		0	0		
	14	GENERATE	1		0	0		
	15	TERMINATE	1		0	0		
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
CANAL	32	0.884	13.264	1	32	0	0	0
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	
RETRY								
NAKOPITEL	7	6	38	5	2.557	32.300	37.194	
SAVEVALUE	RETRY	VALUE						
SERVEDNOPRIOR	0	24.000						
SERVEDPRIOR	0	7.000						

Всего в модель вошло 38 транзактов, из них повышенный приоритет получило 7 транзактов, что в процентном соотношении составило $7 \times 100 / 38 = 18,4\%$, что далеко от заявленных

25 %. Объясняется это обстоятельство малым временем моделирования. Всего был обработан 31 транзакт, из них 7 приоритетных (22,8 %), т. е. все приоритетные транзакты были обслужены каналом.

Проведение экспериментов с моделью model05.gps

Задание:

- Поменять код модели model05.gps так, чтобы было три типа входящих транзактов: с нулевым приоритетом – 50 %, с приоритетами, равными 1 и 2 – по 25 %.
 - Найти статистику по всем трем типам транзактов (числа транзактов и их процентного соотношения как по вошедшим в модель транзактам, так и по обслуженным).
 - Эксперименты проводить, задавая время моделирования равным 16 часам.
 - В отчет поместить скрин отчета по результатам эксперимента. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
 - Отчет должен завершаться выводом о результатах проведенного эксперимента.
- Шаблон отчета: <https://disk.yandex.ru/i/CqO51q2HmRekQA>

Организация прерываний работы ОКУ

В GPSS нет такого понятия, как абсолютный приоритет транзакта (см. рис. 17), но есть специальные блоки, вызывающие прерывание или, наоборот, снятие прерывания с одноканального устройства. Ими являются блоки PREEMPT и RETURN, соответственно, вызывающие или снимающие прерывание работы ОКУ.

У блока PREEMPT в поле А указывается имя или номер прерываемого ОКУ, а поле В отвечает за режим работы данного блока. Всего предусмотрено два режима:

- если в В присутствует операнд PR, то блок работает в приоритетном режиме. Прерывание может выполнить транзакт с более высоким приоритетом, чем у транзакта, находящегося в канале на обслуживании. Если же приоритет недостаточен, то транзакт будет помещен в очередь на прерывание;
 - если поле В не заполнено, то действует режим прерывания.
- Поля С, D и E определяют порядок действий с прерванным транзактом:
- в С указывается метка блока, куда будет перенаправлен транзакт;
 - в D записывается имя или номер параметра транзакта, в котором сохраняется значение времени, оставшегося до конца обслуживания. Это время используется для завершения обслуживания, когда транзакт вновь вернется в ОКУ;
 - если в E стоит операнд RE, то транзакт теряет возможность дообслужиться в ОКУ.

Когда транзакт, вызвавший прерывание, войдет в блок RETURN, прерывание ОКУ завершится. В поле А блока RETURN должно быть указано имя или номер ОКУ.

Моделирование СМО с прерываниями

Модель, у которой возможны прерывания обслуживания каналом транзакта (см. рис. 17), рассмотрим с такими исходными данными: интенсивность прихода беспriorитетных транзактов составляет $\lambda_{\text{прих1}} = 4$ транзакта в час, обслуживания – $\lambda_{\text{обсл1}} = 6$ транзактов в час. У приоритетных транзактов, соответственно, эти данные составляют $\lambda_{\text{прих2}} = 1$ транзакт в сутки, $\lambda_{\text{обсл2}} = 1$ транзакт за 2 часа. Время моделирования примем 30 суток. Для расчета времени прихода выберем экспоненциальный закон распределения случайной величины, для обслуживания – нормальный с стандартным отклонением равным 1. В качестве единицы измерения времени примем одну минуту. Транзакт, чье обслуживание было прервано, должен удаляться из модели необслуженным. Код модели с прерыванием представлен ниже (<https://altaev-aa.narod.ru/km3/book/model06.gps>):

```

GENERATE (EXPONENTIAL(1,0,(60/4))) ; приход бесприоритетных транзактов
  QUEUE nakopitel
SEIZE canal
DEPART nakopitel
ADVANCE (NORMAL(1,10,1)) ; обслуживание бесприоритетных транзактов
RELEASE canal
SAVEVALUE servedNoPrior+,1 ; обслужено бесприоритетных транзактов
TERMINATE

```

```

GENERATE (EXPONENTIAL(1,0,(24#60))) ; приход приоритетных транзактов
PREEMPT canal,,interrupt,,RE
ADVANCE (NORMAL(1,(2#60),1)) ; обслуживание приоритетных транзактов
RETURN canal
SAVEVALUE servedPrior+,1 ; обслужено приоритетных транзактов
TERMINATE

```

```

interrupt SAVEVALUE interrupted+,1 ; число транзактов с прерванным
обслуживанием
TERMINATE

```

```

GENERATE (30#24#60)
SAVEVALUE averageQueueLength,QA$NAKOPITEL
TERMINATE 1

```

Отчет о результатах моделирования имеет вид:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	2829	0	0
	2	QUEUE	2829	4	0
	3	SEIZE	2825	0	0
	4	DEPART	2825	0	0
	5	ADVANCE	2825	1	0
	6	RELEASE	2795	0	0
	7	SAVEVALUE	2795	0	0
	8	TERMINATE	2795	0	0
	9	GENERATE	38	0	0
	10	PREEMPT	38	0	0
	11	ADVANCE	38	0	0
	12	RETURN	35	0	0
	13	SAVEVALUE	35	0	0
	14	TERMINATE	35	0	0
INTERRUPT	15	SAVEVALUE	32	0	0
	16	TERMINATE	32	0	0
	17	GENERATE	1	0	0
	18	TERMINATE	1	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
CANAL	2863	0.755	11.389	1	2865	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE.(-0)
RETRY							
NAKOPITEL	26	4	2829	672	2.599	39.685	52.049

SAVEVALUE	RETRY	VALUE
SERVEDNOPRIOR	0	2795.000
INTERRUPTED	0	32.000
SERVEDPRIOR	0	35.000
AVERAGEQUEUELENGTH	0	2.599

Для анализа отдельно выпишем количество транзактов вошедших в те или иные блоки:

- SEIZE – 2825 (бесприоритетные транзакты, вошедшие в канал на обслуживание);
- RELEASE – 2795 (бесприоритетные транзакты, покинувшие канал обслуживаемыми);
- PREEMPT – 38 (приоритетные транзакты, прерывавшие обслуживание в канале);
- RETURN – 35 (приоритетные транзакты, завершившие прерывание канала).

Вычислим количество случаев прерывания обслуживания неприоритетных транзактов. Учитывая, что один транзакт остался в блоке ADVANCE, т. е. не покинул его, всего было прервано обслуживание 29 неприоритетных транзактов ($29 = 2825 - 2795 - 1$).

Разница между числом транзактов, вошедших в PREEMPT и в RETURN, составляет 3 транзакта.

Блок SAVEVALUE с меткой interrupt подсчитывает количество прерываний и из отчета видно, что это число равно 32 (это число соответствует СЧА X\$INTERRUPTED). Разница между 32 потерянными транзактами и 29 неприоритетными транзактами составляет 3. Это число приоритетных транзактов, чье обслуживание было прервано такими же приоритетными транзактами.

В данной модели был выбран режим прерывания и в трех случаях, когда в канале на обслуживании находился приоритетный транзакт, другой пришедший приоритетный транзакт вызывал прерывание его обслуживания, т. е. происходило прерывание прерывания.

Коэффициент использования канала (столбец «UTIL») составляет 0,755, что говорит о том, что примерно в 25 % всего времени моделирования канал находился в состоянии «свободно». То есть в четверти случаев, когда приоритетный транзакт входил в модель, он заставлял канал в состоянии «свободно».

Проведение экспериментов с моделью model06.gps

Пояснения по предстоящей серии экспериментов:

В GPSS генераторы RN на самом деле создают последовательности псевдослучайных чисел. Это означает, что при каждом новом запуске модели будет генерироваться одна и та же последовательность случайных чисел. При этом отчеты будут иметь одинаковые результаты. Для сбора более достоверной статистики желательно модель прогонять при разных последовательностях псевдослучайных чисел. Так, например, в функциях NORMAL(Stream,Mean,StdDev) и EXPONENTIAL(Stream,Locate,Scale) их параметр Stream служит для обращения к генератору RN. При вызове функции NORMAL(1,(2#60),1) используется RN1, если вызов записан в виде NORMAL(2,(2#60),1), то используется RN2 и т. д.

Задание:

- Провести эксперименты с моделью model06.gps, меняя время обслуживания приоритетных транзактов от 1 часа до 3 с шагом 0,5 и наблюдая за изменением средней длины очереди. Эксперименты проводить, меняя генератор RN у функции, отвечающей за вычисление времени обслуживания приоритетных транзактов.

- Результаты занести в excel файл:

<https://altaev-aa.narod.ru/km3/book/model06.xlsx>

При внесении данных в таблицу учитывайте, что в Excel целая часть от дробной отделяется запятой. В строке 7 автоматически будут вычисляться средние значения длины очереди. После очередного внесения данных рекомендуется сохранять таблицу во избежание потери данных при случайном закрытии окна таблицы.

- Построить точечную диаграмму зависимости средней длины очереди от времени обслуживания приоритетных транзактов.

Указания по созданию диаграммы:

- Выделите ячейки таблицы со средним временем обслуживания приоритетных транзактов (с числами 1; 1,5; 2; 2,5; 3). Далее, нажав и не отпуская клавишу Ctrl, выделите ячейки, в которых отображаются средние значения длины очереди (ячейки B7:F7). Отпустите Ctrl и в меню Вставка выберите кнопку «Вставить точечную (X, Y) или пузырьковую диаграмму».

- Подобрать подходящее уравнение регрессии.
 - В отчет поместить скрин одного из отчетов прогона модели. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
 - В отчет также поместить скрин таблицы с результатами эксперимента и диаграмму с линией тренда.
 - Отчет должен завершаться выводом об результатах проведенного эксперимента.
- Шаблон отчета: <https://disk.yandex.ru/d/kH2apSsDzomzKg>

Таблицы

Таблицы предназначены для сбора статистики по интересующему параметру модели. Статистика основана на наблюдении за частотой попадания указанного параметра модели в заданные интервалы значений.

Для применения таблиц в модели используется карта описания TABLE и блок TABULATE. Меткой карты является имя или номер таблицы и такие же данные вписываются в поле А блока. Это позволяет установить связь между картой и блоком. Предназначение полей карты следующее:

- в поле А записывается параметр, значение которого табулируется (параметр может быть СЧА, выражением или именем);
- в В – верхняя граница первого частотного интервала (нижняя граница интервала принята равной $-\infty$);
- в С – ширина остальных интервалов;
- в D – количество интервалов

У таблиц всего два основных СЧА:

- ТВ – среднее значение табулируемой величины;
- TD – среднеквадратичное отклонение.

В качестве примера использования таблиц рассмотрим модель, которая позволит построить гистограмму распределения времен прихода транзактов в модель. Для расчета указанного времени будет применен закон нормального распределения СВ. Гистограммы распределения времен прихода транзактов были получены ранее в теме «Расчет времени наступления основного события» и отображены на рисунке 25 для двух значений среднеквадратичного отклонения σ .

Код модели имеет вид (<https://altaev-aa.narod.ru/km3/book/model07.gps>):

```

INITIAL X$timeArrival,0
Time TABLE X$timeArrival,.5,1,20
GENERATE (NORMAL(1,8,2))
SAVEVALUE timeArrival+,C1
TABULATE Time
SAVEVALUE timeArrival,0
SAVEVALUE timeArrival-,C1
SAVEVALUE averageTimeArrival,TB$Time
TERMINATE 1

```

В данной модели для расчета времени, прошедшего между приходами предыдущего и текущего транзактов, используется СЧА X\$timeArrival. Атрибут C1 позволяет получить текущее модельное время. Предыдущий транзакт заносит значение C1 в X\$timeArrival со знаком минус (блок SAVEVALUE timeArrival-,C1), текущий добавляет C1 со знаком плюс (блок SAVEVALUE timeArrival+,C1). Разница времен C1 и дает значение времени, прошедшего между прибытиями двух транзактов, т. е. это и есть относительное время, которое следует табулировать.

В модели среднее время прихода транзактов принято равным 8, среднеквадратичное отклонение $\sigma = 2$, число транзактов = 10000 (карта START 10000). Отчет о результатах прогона модели имеет вид:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	10000		0	0
	2	SAVEVALUE	10000		0	0
	3	TABULATE	10000		0	0
	4	SAVEVALUE	10000		0	0
	5	SAVEVALUE	10000		0	0
	6	SAVEVALUE	10000		0	0
	7	TERMINATE	10000		0	0
TABLE	MEAN	STD.DEV.	RANGE		RETRY	FREQUENCY
CUM. %	8.008	1.965			0	
TIME			0.500 -	1.500		4
0.04			1.500 -	2.500		18
0.22			2.500 -	3.500		83
1.05			3.500 -	4.500		286
3.91			4.500 -	5.500		640
10.31			5.500 -	6.500		1159
21.90			6.500 -	7.500		1777
39.67			7.500 -	8.500		2022
59.89			8.500 -	9.500		1761
77.50			9.500 -	10.500		1236
89.86			10.500 -	11.500		643
96.29			11.500 -	12.500		261
98.90			12.500 -	13.500		79
99.69			13.500 -	14.500		25
99.94			14.500 -	15.500		5
99.99			15.500 -	16.500		1
100.00						
SAVEVALUE	RETRY	VALUE				
TIMEARRIVAL	0	-80075				
AVERAGETIMEARRIVAL	0	8.008				

Среднее время прихода транзактов составило 8,008 (СЧА TB\$AVERAGETIMEARRIVAL и TB\$Time), что близко к $\bar{\tau}_{\text{прих}} = 8$, заявленному в функции NORMAL. В отчете также присутствует значение среднеквадратичного отклонения, вычисленного для затабулированной величины (TD\$Time), равное 1,965. Максимальное значение, полученное для интервала 7,5–8,5, близко к значению, имеющему в таблице 9 для числа 8 (2022 и 1915).

GPSS World позволяет на основе таблицы получить гистограмму частот, для чего следует в меню выбрать команды: Window/Simulation Window/Table Window и выбрать таблицу Time. Построенная GPSS гистограмма приведена на рисунке 42.

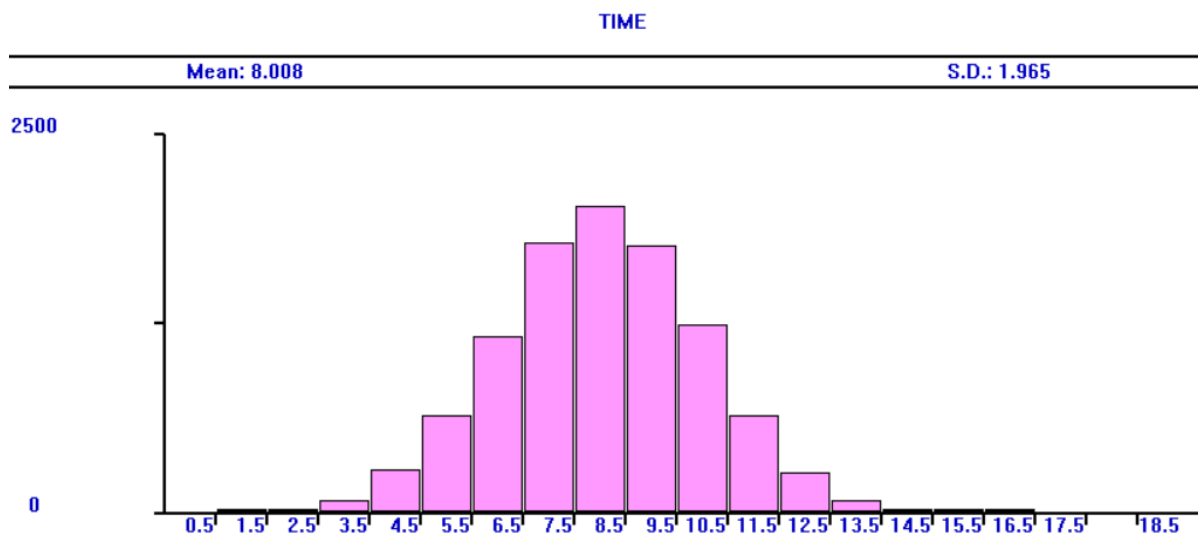


Рисунок 42 – Гистограмма частот распределения времени прихода транзактов

Проведение экспериментов с моделью model07.gps

Задание:

- Изучить зависимость среднего времени прихода транзактов от числа пришедших в модель транзактов.
- Провести эксперименты, задавая значения числа проходящих через модель транзактов как 10, 100, 1000, 10000 и 100000. Указанные значения и полученные в ходе эксперимента значения среднего времени прихода транзактов поместить в таблицу отчета.
- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- В выводе дать оценку того как меняется разность между полученными значениями среднего времени прихода транзактов и заданным в модели значением времени прихода в зависимости от числа проходящих через модель транзактов.

Шаблон отчета: <https://disk.yandex.ru/d/kH2apSsDzomzKg>

Обеспечение минимального значения времени при использовании функции EXPONENTIAL

В некоторых случаях необходимо указывать значение минимального времени между прибытиями транзактов в модель. У функции EXPONENTIAL предусмотрен для этого случая второй аргумент, в котором указывается минимальное время задержки между прибытиями транзактов. Рассмотрим пример, когда в модель должны прибывать транзакты со средним временем прихода 0,09, но минимальное время при этом должно составлять 0,03. Если используется функция, то в нем среднее время следует указывать не 0,09, а 0,06 с учетом минимальной задержки (<https://altaev-aa.narod.ru/km3/book/model07a.gps>):

```
INITIAL X$timeArrival,0
Time1 TABLE X$timeArrival,.02,.02,20

GENERATE (EXPONENTIAL(1,.03,(.09-.03)))
SAVEVALUE timeArrival+,C1
TABULATE Time1
SAVEVALUE timeArrival,0
SAVEVALUE timeArrival-,C1
SAVEVALUE averageTimeArrival,TB$Time1
TERMINATE 1
```

Прогон такой модели с картой START 1000000 выводит отчет:

CUM. %	TABLE	MEAN	STD. DEV.	RANGE	RETRY	FREQUENCY
	TIME1	0.090	0.060		0	
15.34			0.020	-	0.040	153419
39.29			0.040	-	0.060	239462
56.56			0.060	-	0.080	172688
68.82			0.080	-	0.100	122588
77.65			0.100	-	0.120	88358
84.00			0.120	-	0.140	63462
88.51			0.140	-	0.160	45077
91.74			0.160	-	0.180	32306
94.09			0.180	-	0.200	23503
95.77			0.200	-	0.220	16844
96.97			0.220	-	0.240	11953
97.82			0.240	-	0.260	8497
98.43			0.260	-	0.280	6168
98.88			0.280	-	0.300	4473
99.19			0.300	-	0.320	3111
99.42			0.320	-	0.340	2308
99.58			0.340	-	0.360	1624
99.70			0.360	-	0.380	1199
100.00			0.380	-		2960
	SAVEVALUE		RETRY	VALUE		
	TIMEARRIVAL		0	-90075.6		
	AVERAGETIMEARRIVAL		0	0.090		

Среднее время прихода равно 0,03 (СЧА X\$AVERAGETIMEARRIVAL), гистограмма частот времени прибытия имеет смещена вправо (рис. 43):

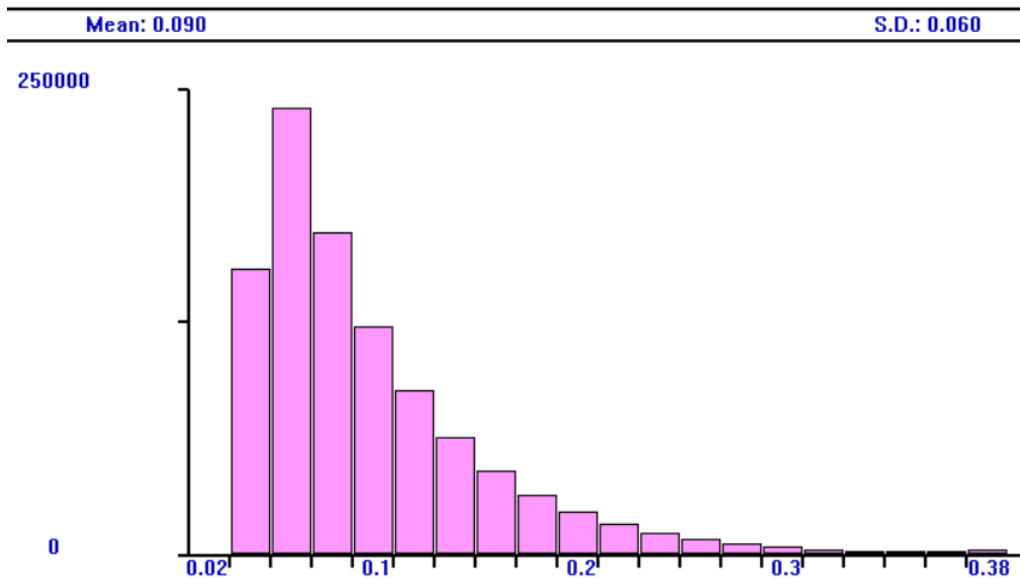


Рисунок 43 – Гистограмма частот времени прибытия транзактов

Примечание. Для этой модели эксперименты не предусмотрены.

Ансамбли транзактов

В ряде случаев в моделях необходимо создавать копии транзактов, либо, напротив, несколько транзактов объединять в один, а также синхронизировать движение копий по модели. В GPSS для реализации подобных моделей имеются следующие блоки:

- **SPLIT** – при входе транзакта в данный блок создаются его копии. Число создаваемых копий указывается в поле А. Поле В предназначено для указания метки блока, на который будут перенаправлены созданные копии, сам транзакт перемещается на следующий блок. Если поле В не заполнено, то копии также как и оригинал перемещаются на следующий блок. Оригинал и его копии образуют общий ансамбль. В поле С (он необязателен) указывается параметр транзакта, в который вписываются порядковые номера – у оригинала значение данного параметра будет равно 1, у первой копии – 2 и т. д.

- **ASSEMBLE** – члены одного ансамбля, собравшиеся в данном блоке будут объединены в один транзакт и он покидает блок. В поле А указывается число транзактов, необходимых для слияния.

- **GATHER** – по принципу действия похож на **ASSEMBLE**, т. е. собирает требуемое число транзактов одного ансамбля. Отличие в том, что после сбора все транзакты одновременно покидают блок.

- **MATCH** – данный блок имеет свою метку. В поле А указывается метка другого блока **MATCH**. Второй блок, аналогично, имеет свою метку и хранит метку первого блока. Таким образом, оба блока сопряжены между собой:

```
labelA MATCH labelB
```

...

```
labelB MATCH labelA
```

При входе транзакта в блок проверяется наличие транзакта – члена его ансамбля, в сопряженном блоке. Если в сопряженном блоке есть транзакт его ансамбля, то оба транзакта одновременно покидают свои блоки. Если же в сопряженном блоке нет соответствующего ему транзакта, то транзакт будет задержан в своем блоке.

Рассмотрим модель, в которую поступают партии из трех транзактов со средним временем прихода, равном 10 единицам. Пришедшие транзакты должны быть обработаны, каждая

в своем канале. Время обработки составляет в среднем 9 единиц. По завершении обработки все три транзакта следует объединить в один. Для расчета времени прихода и обработки использовать экспоненциальный закон распределения. Моделирование провести для 1000 партий. Модель будет иметь вид (<https://altaev-aa.narod.ru/km3/book/model08.gps>):

```

GENERATE (EXPONENTIAL(1,0,10))
SPLIT 2,,ordinalNumber
QUEUE P$ordinalNumber
SEIZE P$ordinalNumber
DEPART P$ordinalNumber
ADVANCE (EXPONENTIAL(1,0,9))
RELEASE P$ordinalNumber
ASSEMBLE 3
TERMINATE 1

```

Блок SPLIT создает для вошедшего в модель транзакта две его копии, в параметр ordinalNumber вносятся порядковые номера транзактов: 1, 2 и 3. Всего в модели будут три независимые ветви, каждая ветвь будет иметь свою очередь и канал. Отчет от прогона модели с картой START 1000 будет иметь вид:

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE		1006		0	0	
	2	SPLIT		1006		0	0	
	3	QUEUE		3018		11	0	
	4	SEIZE		3007		0	0	
	5	DEPART		3007		0	0	
	6	ADVANCE		3007		3	0	
	7	RELEASE		3004		0	0	
	8	ASSEMBLE		3004		4	0	
	9	TERMINATE		1000		0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
1	1001	0.865	9.114	1	2999	0	0	0
5								
2	1001	0.867	9.134	1	3003	0	0	0
5								
3	1005	0.856	8.976	1	3016	0	0	0
1								

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)
RETRY							
1	18	5	1006	144	3.995	41.870	48.864
0							
2	20	5	1006	139	3.881	40.672	47.192
0							
3	22	1	1006	156	3.796	39.781	47.082
0							

Из отчета видно, что по завершении моделирования в блоке ASSEMBLE осталось 4 транзакта. Они не были объединены в один транзакт, так как относились к разным ансамблям. Средние времена обслуживания в каналах 1, 2 и 3 немного отличаются друг от друга (столбцы «FACILITY» и «AVE. TIME»). Также незначительно отличаются остальные характеристики. Статистика по очередям также разная.

Проведение экспериментов с моделью model08.gps

Задание:

- Поменять код модели model08.gps, введя в нее свою фамилию.
- Изучить зависимость времени обработки одной партии от количества транзактов в одной партии.

- Провести эксперименты, задавая значения количества транзактов в интервале от 2 до 6, и полученные результаты занести в таблицу отчета.

Примечание. Временные характеристики модели не изменяются.

- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.

- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: https://disk.yandex.ru/i/73LeYdlp7iv5_w

Примечание. Учитывайте, что если меняется число создаваемых копий, то, соответственно, должно меняться число собираемых в ASSEMBLE транзактов.

Иногда необходимо, чтобы все транзакты принадлежали одному ансамблю. Это требование выполняется за счет ввода в модель одного единственного транзакта, который помещается в бесконечный цикл с периодическим заходом в блок SPLIT, что позволяет создавать копии и тем самым порождать поток входящих транзактов. Ниже приведена измененная версия предыдущей модели (<https://altaev-aa.narod.ru/km3/book/model09.gps>):

```

GENERATE    , , , 1
labelLoop  ADVANCE    (EXPONENTIAL(1, 0, 10))
SPLIT      1, labelLoop
SPLIT      2, , ordinalNumber
QUEUE      P$ordinalNumber
SEIZE      P$ordinalNumber
DEPART     P$ordinalNumber
ADVANCE    (EXPONENTIAL(1, 0, 9))
RELEASE    P$ordinalNumber
ASSEMBLE   3
TERMINATE  1

```

Отчет по такой модели имеет вид:

LABEL	LOC	BLOCK	TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE		1		0	0
LABELLOOP	2	ADVANCE		1007		1	0
	3	SPLIT		1006		0	0
	4	SPLIT		1006		0	0
	5	QUEUE		3018		15	0
	6	SEIZE		3003		0	0
	7	DEPART		3003		0	0
	8	ADVANCE		3003		3	0
	9	RELEASE		3000		0	0
	10	ASSEMBLE		3000		0	0
	11	TERMINATE		1000		0	0

DELAY	FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
3	1	1003	0.916	9.255	1	3005	0	0	0
1	2	1005	0.884	8.917	1	3015	0	0	0
11	3	995	0.928	9.449	1	2986	0	0	0

RETRY	QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)
0	1	29	3	1006	74	8.239	82.995	89.585
0	2	19	1	1006	98	4.936	49.725	55.091
0	3	27	11	1006	74	7.963	80.221	86.591

Блок GENERATE создал единственный транзакт. В цикле имеется блок ADVANCE, который позволяет обеспечить необходимую задержку длительностью 10 единиц времени. Также в цикле находится блок SPLIT, обеспечивающий создание копии транзакта. Остальная часть модели не была изменена. Статистика, приведенная в новом отчете, несколько иная. Обратите внимание на блок ASSEMBLE, в котором на момент завершения моделирования не осталось ни одного транзакта. Причина в том, что в модели на этот раз находились транзакты лишь одного ансамбля.

Проведение экспериментов с моделью model09.gps

Задание:

- Изучить зависимость времени обработки одной партии от времени прихода партии в модель.

Примечание. Учесть, что время обработки включает не только время, затраченное на обслуживание в канале, но и время, которое транзакты пребывали в очереди. Использовать таблицу, в которую заносить СЧА M1.

- Провести эксперименты, задавая значения времени прихода в интервале от 8 до 12 с шагом 1, и полученные результаты занести в таблицу отчета.
- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- Повторить серию экспериментов с условием, что будет обработано 10000 партий.
- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: <https://disk.yandex.ru/i/NvjMSspfqvT3Mw>

Карты описания

Карты INITIAL и EQU

Карта INITIAL позволяет присвоить начальное значение некоторому атрибуту. Например, карта

```
INITIAL X$lossesType1, 0
```

задает значение 0 атрибуту X\$lossesType1.

Карта EQU позволяет сопоставить имя объекта (записывается в виде метки карты) его номеру (записывается в поле A карты). Пример:

```
lossesType1 EQU 1
```

В таком случае к значению сохраняемой величины lossesType1 можно обращаться по СЧА X\$lossesType1 или по СЧА X1.

Переменные

Для создания переменной используется карта описания VARIABLE, у которой метка является его именем, а в поле A записывается выражение, по которому вычисляется значение переменной. Для составления выражения предусмотрены следующие операторы (табл. 11).

Таблица 31

Оператор	Операция	Оператор	Операция
^	Возведение в степень	@	Целый остаток
#	Умножение	-	Вычитание
/	Деление	+	Сложение
\	Целочисленное значение		

Пример:

```
anyVariable VARIABLE Q$nakopitel-5
```

Примечание. В картах описания нет обязательного требования помещать выражение внутри наружных скобок.

Если в каком-то блоке модели необходимо вычислить значение указанной выше переменной, то используют ее СЧА V: V\$anyVariable.

В GPSS также имеется булевская переменная (BVARIABLE), предусматривающая вычисление лишь двух значений: 0 (false) и 1 (true). Для составления ее выражения применяются операторы, представленные в таблице 12.

Таблица 42

Оператор	Операция	Оператор	Операция	Оператор	Операция
>	Больше	<=	Меньше или равно	&	Логическое «И»
'G'		'LE'		'AND'	
>=	Больше или равно	=	Равно		Логическое «ИЛИ»
'GE'		'E'		'OR'	
<	Меньше	!=	Не равно		
'L'		'NE'			

Пример описания булевской переменной:

```
anyBvariable BVARIABLE (Q$nakopitel-5)'AND'(p$ordinalNumber=1)
```

СЧА переменной имеет вид: BV\$anyBvariable.

Функции

У GPSS имеются встроенные математические функции, которые можно использовать при построении математических выражений:

- ABS – абсолютное значение аргумента;
- EXP – экспоненциальная функция;
- INT – целое значение аргумента путем отбрасывания его дробной части;
- LOG – натуральный логарифм;
- SQR – извлечение квадратного корня.

Помимо встроенных функций пользователь может в модели использовать свои. Карта описания функции FUNCTION содержит два поля:

- А предназначено для указания аргумента функции (как правило это СЧА);
- в поле В с помощью букв С и D указывают тип функции и целого числа – число точек, описывающих функцию. Буква С означает непрерывно (Continuous) заданную функцию, D – дискретно (Discrete) заданную.

СЧА пользовательских функций обозначается через FN.

Пример непрерывной функции, реализующей нормальный закон распределения СВ:

```
NORM FUNCTION RN1,C25
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2/.06681,-1.5
.11507,-1.2/.15866,-1/.21186,-.8/.27425,-.6/.34458,-.4/.42074,-.2
.5,0/.57926,.2/.65542,.4/.72575,.6/.78814,.8/.84134,1/.88493,1.2
.93319,1.5/.97725,2/.99379,2.5/.99865,3/.99997,4/1,5
```

Имя функции задано в области метки, СЧА функции имеет вид FN\$NORM, ее аргументом является встроенный в GPSS генератор случайных чисел RN1. Описывается функция координатами 25 точек. Так, например, первая точка имеет координаты (0, -5), вторая – (.00003, -4), последняя – (1, 5). График функции дан на рисунке 44. Из графика видно, что вероятность обращается в нуль, если значение FN\$NORM стремится к -5, или к 5.

График нормальной функции распределения

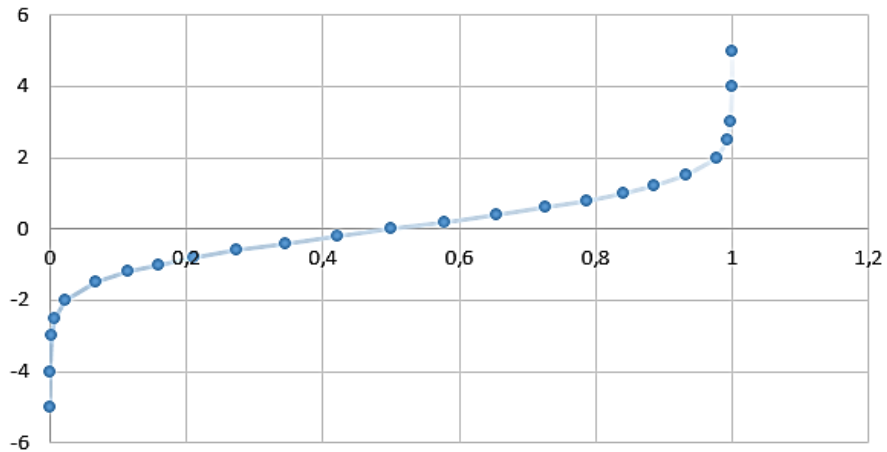


Рисунок 44 – График функции NORM

Так как RN1 может выдавать любые вещественные числа из интервала [0, 1], то для расчета значения FN\$NORM применяется формула линейного интерполирования:

$$fn = y_{i-1} + (rn - x_{i-1}) \frac{y_i - y_{i-1}}{x_i - x_{i-1}},$$

где $x_{i-1} \leq rn \leq x_i$, rn – значение, сгенерированное RN1; x и y – вектора координат точек; $i = \overline{0, n}$; n – число точек (для функции NORM, $i = \overline{0, 24}$).

Графическая иллюстрация линейного интерполирования приведена рисунке 45.

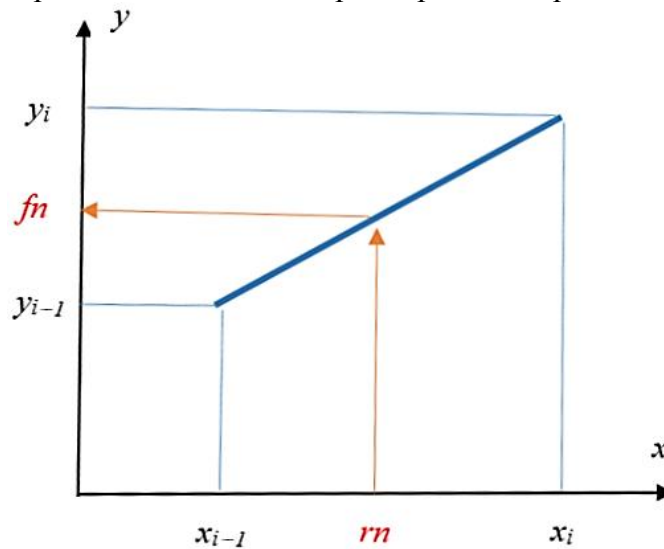


Рисунок 45 – Линейное интерполирование

Пример дискретно заданной функции

```
anyDiscrete FUNCTION RN1, D3
.4, 5/.7, 4/1, 8
```

График данной функции приведен на рисунке 46.

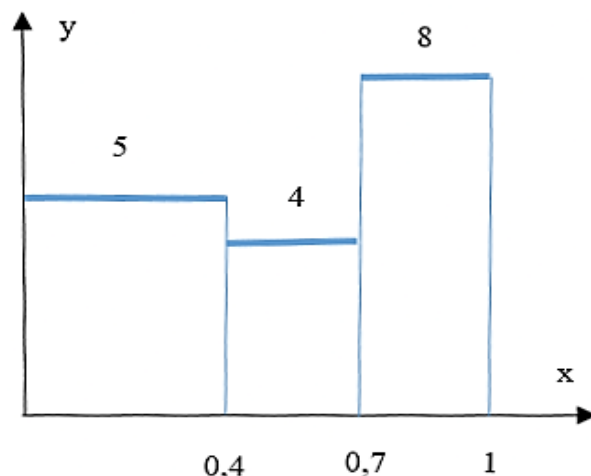


Рисунок 46 – График дискретно заданной функции

Примеры моделей с применением функции и переменной

Рассмотрим модель, использованную в теме «Таблицы», заменив в ней функцию $NORMAL(1,8,2)$ на функцию $NORM$. Для вычисления времени прибытия транзакта применим формулу:

$$\tau_{\text{прих}} = \bar{\tau}_{\text{прих}} + FN\$NORM \times \sigma,$$

где σ – среднееквадратичное отклонение (его примем равным 2). Для вычисления времени прихода транзактов применим карту `timeArr` для описания переменной.

Модель имеет вид (<https://altaev-aa.narod.ru/km3/book/model10.gps>):

```

NORM FUNCTION RN2,C25
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2/.06681,-1.5
.11507,-1.2/.15899,-1/.21186,-.8/.27425,-.6/.34458,-.4
.42074,-.2/.5,0/.57926,.2/.65542,.4/.72575,.6/.78814,.8
.8413,1/.88493,1.2/.93319,1.5/.97725,2/.9979,2.5
.99865,3/.99997,4/1,5
INITIAL X$timeArrival,0
INITIAL X$averageTimeArrival,80
INITIAL X$standardDeviation,2
timeArr VARIABLE X$averageTimeArrival+FN$NORM#X$standardDe-
viation
Time TABLE X$timeArrival,10,2,1000
GENERATE V$timeArr
SAVEVALUE timeArrival+,C1
TABULATE Time
SAVEVALUE timeArrival,0
SAVEVALUE timeArrival-,C1
SAVEVALUE averageTimeArr,TB$Time
TERMINATE 1

```

Координаты точек, использованные для описания функции $NORM$, получены из [10]. Для более точной аппроксимации рекомендуется среднее время выбирать не менее чем 60. В модели принято $X\$averageTimeArrival = 80$.

Таблица частот распределения времени прихода, полученная при применении карты `START` со значением 100000, имеет вид:

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY
CUM. %					
TIME	79.992	2.016		0	
0.01		70.000	- 72.000		5
0.13		72.000	- 74.000		127
2.21		74.000	- 76.000		2074
16.04		76.000	- 78.000		13833
50.09		78.000	- 80.000		34055
84.17		80.000	- 82.000		34078
97.80		82.000	- 84.000		13623
99.87		84.000	- 86.000		2073
100.00		86.000	- 88.000		130
100.00		88.000	- 90.000		2
SAVEVALUE	RETRY	VALUE			
TIMEARRIVAL	0	-7.99917e+006			
AVERAGETIMEARRIVAL	0	80.000			
STANDARDDEVIATION	0	2.000			
AVERAGETIMEARR	0	79.992			

Среднее время прихода транзактов $X_{\text{averageTimeArr}} = 79,992$, что немного отличается от расчетного значения 80.

Проведение экспериментов с моделью model10.gps

Задание:

- Приняв среднеквадратичное отклонение равным 1, подсчитать число транзактов, чье время прихода попало в интервал от 79 до 81. Найти процент таких транзактов по отношению к общему количеству транзактов (100000). Все расчеты реализовать в самой модели.

Рекомендации: использовать блок TEST для отделения потока транзактов с требуемыми характеристиками и поместить на их маршруте блок SAVEVALUE для их подсчета. Для вычисления процента использовать карту VARIABLE.

- Повторить эксперимент со значениями среднеквадратичного отклонения, соответственно 2 и 3. Полученные результаты занести в таблицу отчета.

- В отчет поместить скрин результатов одного из экспериментов с моделью. На скрине должен быть выделен участок модели, на котором присутствует фамилия.

- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: <https://disk.yandex.ru/i/6CTEUt0WF0w29Q>

Применение дискретно заданных функций

Второй пример модели предназначен для демонстрации применения дискретно заданной функции. Пусть в модель информационной системы прибывают сообщения двух типов (служебные и данные) с интервалом 8 миллисекунд (распределение СВ подчинено экспоненциальному закону). Служебные имеют относительный приоритет над данными, вероятность их прибытия оставляет 10 %. Длина служебных сообщений составляет 10 символов, у данных

равновероятно может быть равна 200, 400 и 700 символов. Скорость обработки информации составляет 50 символов в миллисекунду. При ожидании в очереди на обработку более 50 миллисекунд сообщения удалять из системы без обработки. Вывести статистику по необработанным сообщениям.

Код модели имеет вид (<https://altaev-aa.narod.ru/km3/book/model11.gps>):

```

INITIAL X$lossesType1,0
INITIAL X$lossesType2,0
INITIAL X$percentageLoss,0
INITIAL X$processingSpeed,50 ; скорость обработки (символ/мс)
typeMessage FUNCTION X$randomNumber,D2
.1,1/1,2
messageLength FUNCTION X$randomNumber,D4
.1,10/.4,200/.7,400/1,700
time TABLE M1,50,1,100
lengthTable TABLE P$msgLength,15,100,10
percentageOfLosses VARIABLE (X$lossesType1+X$lossesType2)/(X$lossesType1+X$lossesType2+X$processed)#100
GENERATE (EXPONENTIAL(2,0,8))
SAVEVALUE randomNumber,(RN1/1000)
ASSIGN typeMsg,FN$typeMessage
TEST E P$typeMsg,1,lblNoPriority
PRIORITY 1
lblNoPriority ASSIGN msgLength,FN$messageLength
TABULATE lengthTable
QUEUE nakopitel
SEIZE canal
TEST L M1,50,labelLosses
DEPART nakopitel
ADVANCE (P$msgLength/X$processingSpeed)
RELEASE canal
SAVEVALUE processed+,1
SAVEVALUE percentageLoss,V$percentageOfLosses
TERMINATE 1
labelLosses RELEASE canal
TEST E P$typeMsg,1,lblLossesType2
SAVEVALUE lossesType1+,1
TERMINATE
lblLossesType2 SAVEVALUE lossesType2+,1
TABULATE time
TERMINATE

```

Моделирование проводится с условием, что будет обработано 10000 транзактов (START 10000). Пояснения по модели: в функциях typeMessage (тип сообщения – 1 или 2) и messageLength (длина сообщения – 10, 200, 400 и 700) должно быть использовано одно и то же значение случайной величины. Для хранения этой величины используется атрибут X\$randomNumber. RN1 генерирует случайное целое число (прил. 2) и, следовательно, его значение делится на 1000. Переменная percentageOfLosses подсчитывает процент потерь среди числа сообщений, удаленных из модели.

Результаты моделирования имеют вид:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	10731	0	0
	2	SAVEVALUE	10731	0	0
	3	ASSIGN	10731	0	0
	4	TEST	10731	0	0
	5	PRIORITY	1115	0	0
LBLNOPRIORITY	6	ASSIGN	10731	0	0
	7	TABULATE	10731	0	0
	8	QUEUE	10731	0	0
	9	SEIZE	10731	1	0
	10	TEST	10730	0	0

	11	DEPART	10000	0	0
	12	ADVANCE	10000	0	0
	13	RELEASE	10000	0	0
	14	SAVEVALUE	10000	0	0
	15	SAVEVALUE	10000	0	0
	16	TERMINATE	10000	0	0
LABELLOSSES	17	RELEASE	730	0	0
	18	TEST	730	0	0
	19	SAVEVALUE	0	0	0
	20	TERMINATE	0	0	0
LBLLOSSESTYPE2	21	SAVEVALUE	730	0	0
	22	TABULATE	730	0	0
	23	TERMINATE	730	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
CANAL	10731	0.894	7.229	1	10731	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)
RETRY							
NAKOPITEL	737	731	10731	1192	339.305	2742.928	3085.686

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY
CUM. %					
TIME	53.927	3.134		0	
			50.000 -	51.000	136
18.63			51.000 -	52.000	108
33.42			52.000 -	53.000	109
48.36			53.000 -	54.000	82
59.59			54.000 -	55.000	69
69.04			55.000 -	56.000	54
76.44			56.000 -	57.000	46
82.74			57.000 -	58.000	35
87.53			58.000 -	59.000	27
91.23			59.000 -	60.000	23
94.38			60.000 -	61.000	15
96.44			61.000 -	62.000	8
97.53			62.000 -	63.000	13
99.32			63.000 -	64.000	5
100.00					
LENGTHTABLE	390.965	234.428		0	
			- -	15.000	1115
10.39			15.000 -	115.000	0
10.39					

40.17	115.000	-	215.000	3196
40.17	215.000	-	315.000	0
69.65	315.000	-	415.000	3163
69.65	415.000	-	515.000	0
69.65	515.000	-	615.000	0
100.00	615.000	-	715.000	3257
SAVEVALUE	RETRY		VALUE	
LOSSESTYPE1	0		0	
LOSSESTYPE2	0		730.000	
PERCENTAGELOSS	0		6.803	
PROCESSINGSPEED	0		50.000	
RANDOMNUMBER	0		0.876	
PROCESSED	0		10000.000	

Нет ни одного потерянного служебного сообщения, так как X\$lossesType1 отсутствует в отчете (он равен нулю).

Таблица time выводит статистику по временам задержки в очереди потерянных сообщений.

Выведена также статистика по размерам пришедших в модель сообщений (таблица lengthTable). Она составляет 11 % для сообщений размером в 10 символов и, примерно, по 30 % для остальных трех типов данных.

Процент потерь составил 6,8 (СЧА X\$percentageOfLosses).

Проведение экспериментов с моделью model11.gps

Задание:

- Поменять имена таблиц time и lengthTable, добавив к ним свою фамилию.
- Получить гистограммы таблиц time и lengthTable.
- В отчет поместить скрин результата эксперимента с моделью. На скрине должны быть выделены участки модели, на котором присутствует фамилия.
- В выводе дать пояснения по полученным гистограммам.

Шаблон отчета: <https://disk.yandex.ru/i/GAkv-C5Dz9W5Zw>

Блоки, предназначенные для работы с нумерованными объектами

В модели может быть множество однотипных объектов, например, одноканальных устройств и очередей к ним, расположенных не последовательно друг за другом, а размещенных параллельно. В таком случае транзакту достаточно обработки лишь в одном из каналов. Моделирование таких типов моделей удобнее реализовывать, если объекты будут пронумерованы. Для работы с такими нумерованными объектами в GPSS имеются два блока: COUNT и SELECT.

Блок COUNT

Данный блок позволяет провести подсчет объектов, удовлетворяющих условию, указанному в полях блока. Результат заносится в параметр транзакта, вошедшего в этот блок.

COUNT X A, B, C, D, E

Два последних поля могут отсутствовать. Блок может работать в двух режимах: условном и логическом. Выбор нужного режима зависит от значения логического оператора X.

Логический режим

Логический оператор в зависимости от типа проверяемых объектов может иметь следующие значения:

- NU, U, NI, I, FV, FNV – для группы нумерованных одноканальных устройств (ОКУ). Например, NU означает «не используется» (not used), т. е. канал находится в состоянии «свободно». Такой оператор позволяет подсчитать число свободных каналов. Другие варианты оператора: U, NI, I, FV, FNV, соответственно, позволяют подсчитать число используемых каналов, не находящихся в состоянии прерывания, находящихся в состоянии прерывания, доступных и недоступных каналов;

- SE, SNE, SF, SNF, SV, SNV – для группы нумерованных многоканальных устройств (МКУ). SE расшифровывается как «память пуста» (Storage is Empty). В таком случае будет проведен подсчет числа пустых памятей. Остальные значения: SNE, SF, SNF, SV, SNV позволяют провести подсчет, соответственно, числа непустых памятей, полностью заполненных, незаполненных, доступных и недоступных памятей;

- LR, LS – для подсчета числа логических ключей, находящихся либо в состоянии Reset, либо Set.

Условный режим

Позволяет выполнить сравнение двух величин. Для сравнения могут использоваться следующие варианты логического оператора: L (меньше), NE (не равно), LE (меньше или равно), E (равно), G (больше), GE (больше или равно). Также предусмотрены варианты MAX, MIN для выбора максимального или минимального значения.

Поля блока COUNT

В поле A указывается номер или имя параметра транзакта, куда будет заноситься вычисленное блоком COUNT значение. Поля B и C предназначены для указания границ интервалов номеров проверяемых объектов ($B < C$). Поля D и E используются только в условном режиме, причем в E указывается СЧА, по которому проводится анализ, а поле D может быть пустым или в нем может содержаться значение, используемое для сравнения с СЧА поля E.

Примеры

```
COUNT U 7, 2, 10
```

При входе транзакта в блок, будут проверены каналы с номерами от 2 по 10 и среди них подсчитано число используемых каналов. Результат будет сохранен в 7-й параметр этого транзакта.

```
COUNT SE countStorageEmpty, 1, 8
```

Среди памятей с номерами от 1 до 8 будет подсчитано число пустых и результат будет занесен в параметр countStorageEmpty вошедшего в блок транзакта.

```
COUNT LE countQueue, 1, 10, 7, Q
```

Среди очередей с номерами от 1 до 10 будут найдены те, у которых длина (их содержимое) не превысит 7, и их число будет занесено в параметр countQueue.

```
COUNT MAX maxQueue, 1, 10, , Q
```

Среди очередей с номерами от 1 до 10 будет подсчитано число очередей с максимальным содержимым и результат будет занесен в параметр maxQueue.

Блок SELECT

Данный блок позволяет выбрать объект, удовлетворяющий условию, указанному в блоке. Также, как и блок COUNT, предусматривает те же два режима, имеет логический оператор с теми же функциями, но вдобавок у него есть поле F, в котором указывается метка перехода, куда может быть перемещен транзакт. В случае, если поле F пропущено, то транзакт переходит на следующий блок.

Примеры

```
SELECT MIN 2,2,10,,FR
```

При входе транзакта в блок, будут проверены каналы с номерами от 2 по 10 и среди них будет найден наименее нагруженный канал (СЧА FR определяет степень загруженности канала [см. прил. 3]). Номер найденного канала будет сохранен во 2-й параметр транзакта. Если у нескольких каналов СЧА FR будет иметь минимальное значение, то номер первого из них будет сохранен.

```
SELECT SV memoryNumber,1,6,,,labelLosses
```

Будут проверены на доступность 6 памятей и номер первой из доступных памятей будет занесен в параметр memoryNumber. Транзакт при этом перейдет в следующий блок. Если не будет обнаружено ни одной доступной памяти, то транзакт перейдет по метке labelLosses. В данном блоке поля D и E не используются.

```
SELECT LE queueNumber,1,10,X7,Q
```

Длины очередей с номерами с 1 по 10 сравниваются со значением СЧА X7 и номер первой из них, чья длина не превышает значения X7, заносится в параметр queueNumber. Сам транзакт перемещается на следующий блок.

Моделирование СМО с нумерованными объектами

Транзакты прибывают в систему в соответствии с экспоненциальным распределением со средним временем прибытия 100. Их обслуживание может быть проведено на одном из 10 каналов со средним временем обслуживания 1000. Распределение времени обслуживания также экспоненциальное. Очереди перед каждым каналом имеют емкость, равную 1 транзакту. При отсутствии свободного места в очередях транзакты выбывают из модели необслуженными. Моделирование завершить, когда будет обработано 1000 транзактов.

Код модели имеет вид (<https://altaev-aa.narod.ru/km3/book/model12.gps>):

```
queue1 EQU 1
queue2 EQU 2
queue3 EQU 3
queue4 EQU 4
queue5 EQU 5
queue6 EQU 6
queue7 EQU 7
queue8 EQU 8
queue9 EQU 9
queue10 EQU 10
queue1 STORAGE 1
queue2 STORAGE 1
queue3 STORAGE 1
queue4 STORAGE 1
queue5 STORAGE 1
queue6 STORAGE 1
queue7 STORAGE 1
queue8 STORAGE 1
queue9 STORAGE 1
queue10 STORAGE 1
GENERATE (EXPONENTIAL(1,0,100))
SELECT SNF queueNumber,1,10,,,labelLosses ; есть свободные очереди?
```

```

ENTER      P$queueNumber
SEIZE     P$queueNumber
LEAVE     P$queueNumber
ADVANCE   (EXPONENTIAL(1,0,1000))
RELEASE   P$queueNumber
TERMINATE 1

```

```

labelLosses SAVEVALUE  countLosses+,1
TERMINATE

```

Примечание. Для присвоения номеров памятям была использована карта EQU.
 При прогоне модели с картой «START 1000» был получен отчет:

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE		1126		0	0	
	2	SELECT		1126		0	0	
	3	ENTER		1014		7	0	
	4	SEIZE		1007		0	0	
	5	LEAVE		1007		0	0	
	6	ADVANCE		1007		7	0	
	7	RELEASE		1000		0	0	
	8	TERMINATE		1000		0	0	
LABELLOSSES	9	SAVEVALUE		112		0	0	
	10	TERMINATE		112		0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
1	1	122	0.985	929.636	1	1114	0	0
1	2	108	0.980	1044.374	1	1117	0	0
1	3	105	0.961	1054.188	1	1060	0	0
1	4	116	0.943	936.238	1	1111	0	0
1	5	124	0.944	876.433	1	1119	0	0
1	6	108	0.929	990.114	1	1121	0	0
1	7	106	0.836	908.005	1	1124	0	0
0	8	89	0.785	1015.317	1	0	0	0
0	9	69	0.632	1054.912	1	0	0	0
0	10	60	0.549	1052.701	1	0	0	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY
DELAY									
QUEUE1	1	0	0	1	123	1	0.890	0.890	0
QUEUE2	1	0	0	1	109	1	0.882	0.882	0
QUEUE3	1	0	0	1	106	1	0.845	0.845	0
QUEUE4	1	0	0	1	117	1	0.805	0.805	0
QUEUE5	1	0	0	1	125	1	0.768	0.768	0
QUEUE6	1	0	0	1	109	1	0.740	0.740	0
QUEUE7	1	0	0	1	107	1	0.626	0.626	0
QUEUE8	1	1	0	1	89	1	0.537	0.537	0
QUEUE9	1	1	0	1	69	1	0.435	0.435	0
QUEUE10	1	1	0	1	60	1	0.357	0.357	0

SAVEVALUE	RETRY	VALUE
COUNTLOSSES	0	112.000

Из отчета видно, что более нагруженными были первые каналы (столбец «UTIL»). Потеряно 112 транзактов.

Проведение экспериментов с моделью model12.gps

Задание: определить номер очереди с максимальным средним содержимым (СЧА SA).
Рекомендации: используйте SELECT с режимом MAX.

Шаблон отчета: <https://disk.yandex.ru/i/PhdttDbAbg36DQ>

Повторение эксперимента с измененными условиями

Изменим предыдущую модель, поставив задачу выровнять нагрузку каналов (<https://altaev-aa.narod.ru/km3/book/model13.gps>):

```

queue1 EQU 1
<пропущено>
queue10 EQU 10
queue1 STORAGE 1
<пропущено>
queue10 STORAGE 1
GENERATE (EXPONENTIAL(1,0,100))
SELECT MIN lessLoadedChannel,1,10,,FR ; выявляем малонагруженный канал
GATE SNF P$lessLoadedChannel,labelSecondWay ; проверка очереди к каналу
ENTER P$lessLoadedChannel
SEIZE P$lessLoadedChannel
LEAVE P$lessLoadedChannel
ADVANCE (EXPONENTIAL(1,0,1000))
RELEASE P$lessLoadedChannel
TERMINATE 1
labelSecondWay SELECT SNF queueNumber,1,10,,labelLosses
ENTER P$queueNumber
SEIZE P$queueNumber
LEAVE P$queueNumber
ADVANCE (EXPONENTIAL(1,0,1000))
RELEASE P$queueNumber
TERMINATE 1
labelLosses SAVEVALUE countLosses+,1
TERMINATE

```

Отчет имеет вид:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1128	0	0
	2	SELECT	1128	0	0
	3	GATE	1128	0	0
	4	ENTER	335	1	0
	5	SEIZE	334	1	0
	6	LEAVE	333	0	0
	7	ADVANCE	333	2	0
	8	RELEASE	331	0	0
	9	TERMINATE	331	0	0
LABELSECONDWAY	10	SELECT	793	0	0
	11	ENTER	677	2	0
	12	SEIZE	675	0	0
	13	LEAVE	675	0	0
	14	ADVANCE	675	6	0
	15	RELEASE	669	0	0
	16	TERMINATE	669	0	0
LABELLOSSES	17	SAVEVALUE	116	0	0
	18	TERMINATE	116	0	0

FACILITY		ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	
DELAY										
	1	120	0.952	904.996	1	1104	0	0	0	
0										
	2	101	0.940	1061.360	1	1112	0	0	0	
1										
	3	90	0.925	1172.130	1	1091	0	0	0	
1										
	4	110	0.874	906.530	1	1122	0	0	0	
0										
	5	109	0.857	896.858	1	0	0	0	0	
0										
	6	113	0.858	865.871	1	1124	0	0	0	
0										
	7	98	0.862	1003.534	1	1115	0	0	0	
0										
	8	91	0.858	1075.992	1	1121	0	0	0	
0										
	9	84	0.856	1162.951	1	1127	0	0	0	
1										
	10	93	0.862	1057.492	1	1117	0	0	0	
0										
STORAGE		CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY
DELAY										
	QUEUE1	1	1	0	1	120	1	0.777	0.777	0
	QUEUE2	1	0	0	1	102	1	0.765	0.765	0
	QUEUE3	1	0	0	1	91	1	0.718	0.718	0
	QUEUE4	1	1	0	1	110	1	0.656	0.656	0
	QUEUE5	1	1	0	1	109	1	0.624	0.624	0
	QUEUE6	1	0	0	1	113	1	0.556	0.556	0
	QUEUE7	1	1	0	1	98	1	0.563	0.563	0
	QUEUE8	1	1	0	1	91	1	0.477	0.477	0
	QUEUE9	1	0	0	1	85	1	0.588	0.588	0
	QUEUE10	1	1	0	1	93	1	0.615	0.615	0
	SAVEVALUE									
	COUNTLOSSES									
			RETRY		VALUE					
			0		116.000					

В данном случае нагрузка у каналов выровнялась, но при этом увеличилось число потерь (116 транзактов). *Вывод:* более оптимальным вариантом является первый, когда проводится выбор канала по состоянию очереди перед ним.

Проведение экспериментов с моделью model13.gps

Задание: определить номер очереди с минимальным средним временем задержки транзакта в очереди. Требуемый тип СЧА определить самостоятельно.

Шаблон отчета: https://disk.yandex.ru/d/PcJO2rpz_nqtfq

Косвенная адресация

Применение косвенной адресации позволяет уменьшить объем модели, в которой используется большое число нумерованных объектов. В таких моделях номера объектов, как правило, сохраняются в параметрах транзактов. Косвенная адресация позволяет использовать сохраненные номера объектов для работы с ними.

При применении косвенной адресации используется запись в виде СЧА*СЧА. Например, в картах описания

```

memory1      EQU    1
...
memory10     EQU    10
memory1      STORAGE 510500
...
memory10     STORAGE 510500
memoryAvailability VARIABLE 510500-S*pServer

```

с помощью переменной `memoryAvailability` вычисляется объем доступной памяти у МКУ, чей номер указан в параметре `pServer` транзакта.

Рассмотрим пример модели, в которой не используются памяти и применена косвенная адресация. Пусть прибытие транзактов подчинено экспоненциальному закону со средним временем прибытия 150. Обслуживание проводится как в предыдущем случае, на одном из 10 каналов со средним временем обслуживания 1500 (распределение СВ также экспоненциальное). Очереди перед каналами имеют емкость, равную 5. Транзакты в очереди могут занимать от одного до трех мест. При отсутствии свободного места в очередях транзакты удаляются из модели необслуженными. Моделирование завершить, когда будет обработано 1000 транзактов.

Предварительно построим вспомогательную модель, позволяющую случайно вычислить требуемые значения числа мест, занимаемых одним транзактом в очереди (числа 1, 2 и 3). В данной модели (<https://altaev-aa.narod.ru/km3/book/model14.gps>) использована функция `numberSeats` для определения числа мест и таблица `tableNumberSeats` для контроля за частотой выпадения требуемых чисел:

```

numberSeats  FUNCTION          RN1, D3
              .33, 1/.67, 2/1, 3
tableNumberSeats TABLE P$pNumberSeats, .9, .1, 25
              GENERATE (EXPONENTIAL(1, 0, 150))
              ASSIGN    pNumberSeats, FN$numberSeats
              TABULATE  tableNumberSeats
              TERMINATE 1

```

Значения, вычисленные с помощью функции, заносятся в параметр транзакта и переносятся в таблицу. Прогон через модель тысячи транзактов дает следующую гистограмму генерируемых чисел (рис. 47).

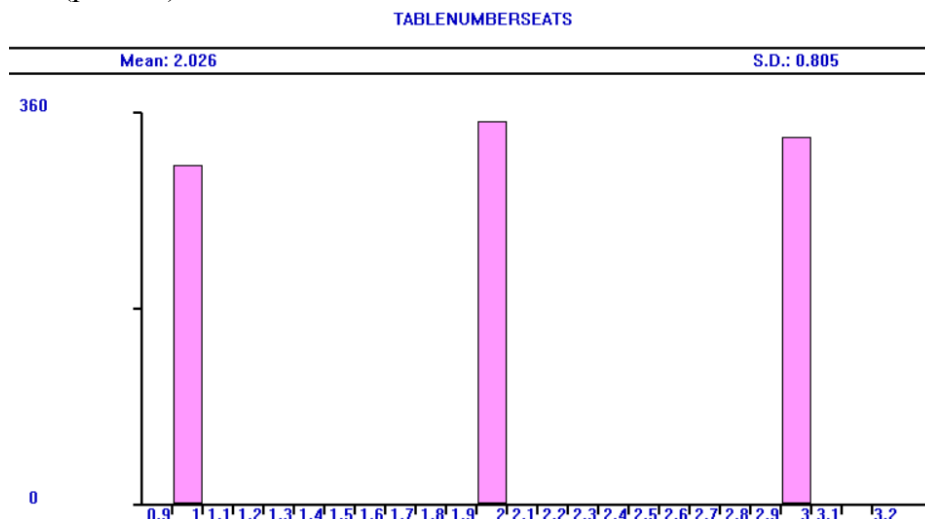


Рисунок 47 – Гистограмма частот СВ

Как видим, функция `numberSeats` позволяет получить три целых числа с примерно равной вероятностью.

Проведение экспериментов с моделью model14.gps

Задание:

- Измените условия задачи так, чтобы было четыре типа транзактов и их вероятности были следующими – 10 % для первого типа и по 30 % для остальных.
- В отчет поместить скрин отчета эксперимента и скрин гистограммы. На скрине должен быть выделен участок модели, на котором присутствует фамилия.
- В отчете должен быть вывод о результатах эксперимента.

Шаблон отчета: <https://disk.yandex.ru/i/ifjrGvuvtOYkpw>

Пример применения модели с нумерованными очередями

Как было оговорено в условии задачи, в модели не использованы памяти для моделирования очередей. В таком случае для корректной работы блока SELECT в модели предварительно должны быть созданы все очереди. С учетом сказанного модель имеет следующий вид (<https://altaev-aa.narod.ru/km3/book/model15.gps>):

```
queue1 EQU 1
queue2 EQU 2
queue3 EQU 3
queue4 EQU 4
queue5 EQU 5
queue6 EQU 6
queue7 EQU 7
queue8 EQU 8
queue9 EQU 9
queue10 EQU 10
numberSeats FUNCTION RN1,D3
.33,1/.67,2/1,3
```

```
GENERATE ,,,1 ; данный сегмент отвечает за создание очередей
ASSIGN countQueue,10
labelLoop ASSIGN numberQueue+,1
QUEUE P$numberQueue
DEPART P$numberQueue
LOOP countQueue,labelLoop
TERMINATE
```

```
GENERATE (EXPONENTIAL(1,0,150))
ASSIGN pNumberSeats,FN$numberSeats
SELECT MIN pMinimalQueue,1,10,,Q ; выявляем очередь с минимальной длиной
TEST L P$pNumberSeats,(5-Q*pMinimalQueue),labelRefusal ; проверка очереди, от 5 вычитывается текущая длина очереди
QUEUE P$pMinimalQueue,P$pNumberSeats
SEIZE P$pMinimalQueue
DEPART P$pMinimalQueue,P$pNumberSeats
ADVANCE (EXPONENTIAL(1,0,1500))
RELEASE P$pMinimalQueue
TERMINATE 1
```

```
labelRefusal SAVEVALUE countLosses+,1
TERMINATE
```

Отчет прогона модели с TG1 = 1000 имеет вид

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1	0	0
	2	ASSIGN	1	0	0
LABELLOOP	3	ASSIGN	10	0	0
	4	QUEUE	10	0	0
	5	DEPART	10	0	0
	6	LOOP	10	0	0
	7	TERMINATE	1	0	0

		8	GENERATE	1085		0	0		
		9	ASSIGN	1085		0	0		
		10	SELECT	1085		0	0		
		11	TEST	1085		0	0		
		12	QUEUE	1032		22	0		
		13	SEIZE	1010		1	0		
		14	DEPART	1009		0	0		
		15	ADVANCE	1009		9	0		
		16	RELEASE	1000		0	0		
		17	TERMINATE	1000		0	0		
	LABELREFUSAL	18	SAVEVALUE	53		0	0		
		19	TERMINATE	53		0	0		
	FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY	1	103	0.986	1682.596	1	1046	0	0	0
2	2	117	0.992	1490.953	1	1058	0	0	0
2	3	132	0.976	1299.547	1	1055	0	0	0
3	4	106	0.975	1616.536	1	1047	0	0	0
2	5	101	0.964	1677.061	1	1040	0	0	0
2	6	100	0.966	1697.343	1	1052	0	0	0
3	7	103	0.915	1561.943	1	1060	0	0	0
2	8	90	0.845	1650.528	1	1061	0	0	0
2	9	88	0.717	1431.466	1	1035	0	0	0
2	10	70	0.712	1788.843	1	1045	0	0	0
2									
	QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	
RETRY	1	4	3	219	22	2.279	1829.306	2033.594	
0	2	4	4	234	12	2.142	1608.969	1695.940	
0	3	4	4	264	44	2.006	1335.800	1602.960	
0	4	4	4	209	18	1.970	1656.694	1812.822	
0	5	4	4	194	33	1.840	1666.742	2008.373	
0	6	4	4	210	22	2.042	1709.290	1909.314	
0	7	4	4	215	42	1.661	1358.169	1687.898	
0	8	4	4	198	61	1.644	1459.874	2109.891	
0	9	4	4	189	67	1.257	1169.290	1811.441	
0	10	4	4	145	33	1.287	1559.888	2019.498	
0									
	SAVEVALUE		RETRY		VALUE				
	COUNTLOSSES		0		53.000				

Из отчета видно, что при 1000 обслуженных транзактов было допущено 53 потери.

Проведение экспериментов с моделью model15.gps

Задание:

- Подсчитать по отдельности число потерянных транзактов всех трех типов и их процентное соотношение к числу обработанных их типа. Все подсчеты организовать в самой модели.

- Полученные результаты занести в таблицу отчета.

Шаблон отчета: <https://disk.yandex.ru/i/4sQgaycRcvt6KQ>

Варианты заданий

Вариант 1. Моделирование канала связи

Два приложения отправляют серверу данные через общий канал связи, имеющий ограниченную пропускную способность. Сообщения, отправляемые обеими приложениями, вмещают от 10 до 100 символов (СВ распределена по данному интервалу равномерно). Сервер может управлять потоком данных, поступающих от первого приложения, за счет регулировки длительности пауз между его сообщениями. Сообщения от второго приложения поступают в канал в среднем через 90 миллисекунд (мс), распределение СВ экспоненциальное.

Канал связи имеет свой накопитель, рассчитанный на 1000 символов. Сообщения, для которых в накопителе нет места, отфильтровываются. Пропускная способность канала составляет 1 символ в миллисекунду.

В начале моделирования считать, что первое приложение отправляет свои сообщения детерминировано с промежутком 200 мс. Сервер, для которого более критичным является время прихода сообщений, чем потеря данных, может в ответных сообщениях менять указанное время как в сторону увеличения, так и в сторону уменьшения с шагом 10 мс. При отсутствии потерь время отправки уменьшается на 10 мс. Если среди следующих 100 сообщений также нет потерь, то время вновь уменьшается на 10 мс и т. д., пока не начнут появляться потери. Допустимым считается потеря не более 10 сообщений от первого приложения на 100 отправленных.

Считая, что должно быть принято 100000 сообщений, провести моделирование и найти оптимальный интервал времени отправки сообщений от первого приложения.

Вариант 2. Моделирование работы цеха упаковки

В упаковочный цех завода поступают готовые изделия партиями по 70 единиц через каждые 390 минут. Упаковка изделий происходит в соответствии с размером заказов дистрибуторов:

Частота заказов	.10	.25	.30	.15	.12	.05	.03
Размер заказа в единицах	6	12	18	24	30	36	48

Заказы поступают в среднем через 15 мин (распределение СВ экспоненциальное). Упаковка заказа на станке занимает 2 мин с добавлением дополнительных 10 с на каждое изделие.

Определить среднее количество заказов, ожидающих своего выполнения в упаковочном цехе. Время моделирования: 5 дней.

Вариант 3. Моделирование работы САУ

Микроконтроллер САУ производственной линии каждые 3 с поочередно опрашивает три датчика на наличие в них готовой информации. Информация появляется в датчиках с периодичностью [9, 15] с (распределение СВ в данном интервале равномерное), размер этой информации распределен в интервале [2000, 4000] символов также по равномерному закону. Считанную информацию микроконтроллер передает на компьютер. Производительность компьютера составляет 1000 символов в секунду. Если компьютер не успевает обработать часть

полученной информации, то ее остаток перенаправляет в очередь неоконченных заданий. Информация из очереди затем извлекается и дорабатывается лишь тогда, когда на очередном опрошенном датчике не оказалось готовых к отправке данных.

Необходимо определить степень загруженности ЭВМ, проведя моделирование в течение 5 ч. Найти основные характеристики очереди.

Вариант 4. Моделирование цепи Маркова

Смоделировать работу вероятностного автомата, представленного на рисунке 5. У этой версии автомата входной алфавит состоит из одного символа. Матрица вероятностей переходов имеет вид:

$$A = \begin{pmatrix} 0 & 0,75 & 0,25 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0,5 & 0 & 0 & 0,5 \end{pmatrix},$$

вектор вероятностей начальных состояний:

$$c = \begin{pmatrix} 0,8 \\ 0,1 \\ 0,1 \\ 0 \end{pmatrix}.$$

Смоделировать работу автомата из расчета, что на его вход поступит входное слово длиной в 1000 символов, и построить гистограмму частот выходных сигналов $\{0, 1, 2, 3\}$.

Вариант 5. Моделирование сети видеохостинга

Сеть видеохостинга включает пять региональных и один главный сервер. Региональные сервера поддерживают одновременное подключение 640 пользователей, главный – подключение 4000 пользователей. Балансировщик нагрузки запрос пользователя направляет в первую очередь на региональный сервер, используя критерий близости пользователя к этому серверу. Региональные сервера содержат лишь 50 % общего контента и, если нужное клиенту видео не обнаруживается на них, то запрос перенаправляется на главный сервер. В период пиковой нагрузки к ресурсам видеохостинга обращается 5000 пользователей в час (распределение СВ принять экспоненциальным). Длительность просмотра видео составляет у 80 % пользователей от 20 мин до одного часа, у остальных – от одного часа до трех (распределение СВ принять равномерным).

Провести наблюдение за сетью в течение 5 ч и определить:

- степень загруженности серверов;
- процент отказов в обслуживании из-за перегруженности серверов.

Вариант 6. Моделирование фондового рынка

Фондовый рынок характерен противоборством двух типов игроков: быков (bull) и медведей (bear). Первые делают ставку на возможное повышение стоимости продаваемых на фондовом рынке акций, вторые, напротив, играют на их понижении. Если на рынке активность проявляют в основном быки, то такой рынок принято называть «бычьим», если же преобладают медведи, то – «медвежьим». В ситуациях, когда нет явного преобладания тех или иных, то говорят, что рынок стагнирует (stagnant).

Для моделирования рынка введем множество состояний $\{1 = \text{бык}, 2 = \text{медведь}, 3 = \text{застой}\}$, а переходы рынка из одного состояния в другое представим в виде матрицы вероятностей переходов (пример взят из [12]).

$$P = \begin{pmatrix} 0,9 & 0,075 & 0,025 \\ 0,15 & 0,8 & 0,05 \\ 0,25 & 0,25 & 0,5 \end{pmatrix}$$

Первая строка матрицы означает, что если на текущей неделе на рынке были активными быки, то имеется 90 % вероятности, что и на следующей неделе рынок останется «бычьим» и

существует лишь 7,5%-ная вероятность, что верх возьмут медведи и имеется незначительная вероятность (2,5 %), что рынок перейдет в состояние стагнации. Граф такой модели рынка показан на рисунке 48.

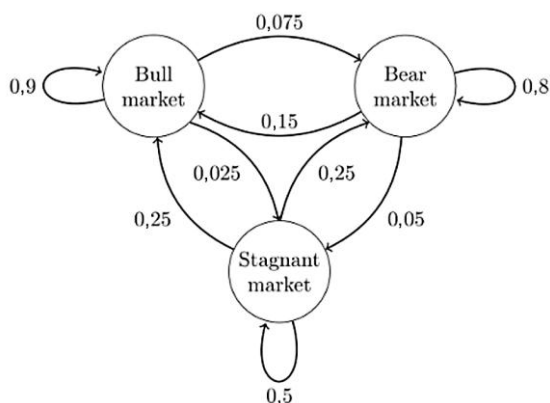


Рисунок 48 – Граф модели рынка

Необходимо смоделировать фондовый рынок в течении года и определить процентное соотношение всех трех состояний рынка. Считать, что все три состояния равновероятны на первой неделе года.

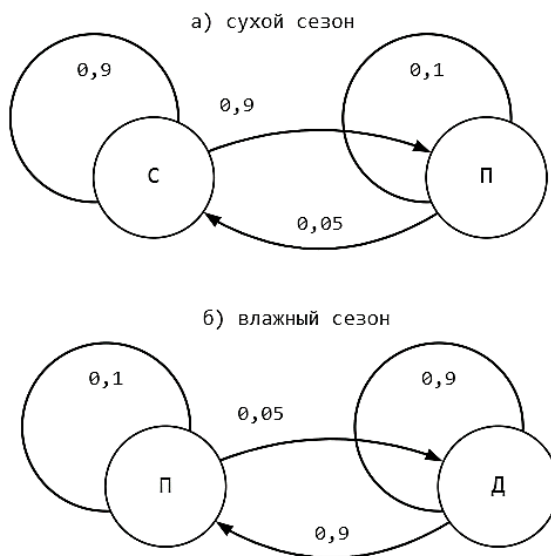
Вариант 7. Моделирование погоды

В одном регионе в течение года наблюдается следующая сменяемость погодных сезонов:

- с января по май – сухой сезон;
- с июня по сентябрь – влажный сезон;
- с октября по декабрь – сухой сезон.

Считая, что есть только три типа дней (пасмурный, солнечный и дождливый), смену дней по сезонам можно представить в виде ориентированных графов (на ребрах указаны вероятности переходов) для сухого (рис. 49 а) и влажного (рис. 49 б) сезонов.

Определить в днях максимальные значения длительностей непрерывных солнечных, пасмурных и дождливых дней за трехгодичный период.



П - пасмурный день, С - солнечный день, Д - дождливый день

Рисунок 49 – Графы сменяемости типов дней

Учесть следующие условия:

- Первый день июня считать 151-м днем года и первый день октября – 274-м днем.
- Считать, что всего в году 365 дней.
- В начале моделирования с учетом того, что в начале первого года будет идти сухой сезон, погоду на первый день определить, исходя из 90%-ной вероятности, что она может быть солнечной и 10%-ной – что пасмурной.
- При смене сезона первый день начавшего сезона считать пасмурным.

Вариант 8. Моделирование VoIP канала

Два узла (компьютер или IP телефон) связаны виртуальным каналом связи, основанным на технологии VoIP (Voice over IP, голос поверх IP). Для кодирования голосового трафика оба устройства используют кодеки, которые формируют сжатые фрагменты речи через каждые 20 мс. Задержка фрагментов при передаче составляет:

- | | | |
|---|---|-----------|
| • Компрессия (кодирование речи) | – | 20–45 мс |
| • На передачу между процессами (в исходном узле) | – | 10–15 мс |
| • Доступ к сети (добавление служебных заголовков) | – | 0,25–7 мс |
| • Задержка на передачу по сети | – | 20–200 мс |
| • Выход из сети (удаление служебных заголовков) | – | 0,25–7 мс |
| • На передачу между процессами (в конечном узле) | – | 10–20 мс |
| • Задержка в компенсирующем буфере | – | 10–20 мс |
| • Декомпрессия | – | 10 мс |

Распределение СВ внутри интервалов времени подчиняется равномерному закону. Предусмотреть очередь перед выходом в сеть. В остальных частях маршрута движения фрагменты речи перемещаются независимо друг от друга.

Промоделировать VoIP канал в течение 1 мин. Построить гистограмму частот времени передачи фрагментов речи через канал, определить среднее время задержки. Считая, что максимальное время задержки не должно превышать 300 мс, определить процент сообщений, чье время оказалось выше указанного значения, по отношению к общему числу переданных фрагментов.

Вариант 9. Моделирование канала связи

В накопителе канала связи в среднем через каждые 9 мс появляются сообщения. Канал передает эти сообщения за 7 мс, но в его работе часто наблюдаются отказы. Частота отказов составляет 1 на каждые 200 мс. Восстановление работоспособного состояния канала занимает 23 мс. На период отказа включается резервный канал связи с такой же пропускной способностью, что и основной. Сообщение, передача которого была прервана по основному каналу, передается по резервному с самого начала. Сообщения удаляются из накопителя лишь после того как оно будет полностью передано по каналу. После восстановления основного канала резервный отключается. Распределение СВ принять экспоненциальным.

Моделирование провести в течение одной минуты. Построить гистограмму времени передачи сообщения по каналу, число сообщений, которые повторно передавались из-за сбоя, определить загруженность запасного канала.

Вариант 10. Моделирование работы АСУ ТП

Программа автоматизированной системы управления производственным процессом (АСУ ТП) работает в восьмипоточном режиме. У каждого потока предусмотрена очередь емкостью в пять сообщений. Сообщения поступают от датчиков через 2 миллисекунды (мс) по экспоненциальному закону распределения СВ. Поступающие сообщения перенаправляются в ту очередь, которая имеет в данный момент минимальное число сообщений. Потоки обрабатывают сообщения в среднем за 15 мс (распределение СВ также экспоненциальное). Если в очередях нет свободных мест, то поступившее сообщение считается потерянным.

Определить в процентах соотношение числа потерянных сообщений к числу обработанных, построить гистограмму времени обработки сообщений (время нахождения их в очереди учитывать). Время моделирования принять равным 10 мин.

Вариант 11. Моделирование работы ЦОД

Центр обработки данных (ЦОД) имеет 5 серверов и систему балансировки нагрузки (СБН). СБН предназначена для выравнивания нагрузки путем ее распределения по нескольким серверам с применением критерия их равномерной загруженности. Указанный критерий учитывает размер используемой запущенными приложениями оперативной памяти (другие критерии приняты некритичными). У всех серверов объем ОЗУ составляет 512 ГБ. Считать, что операционная система со своими службами занимает 1,5 ГБ памяти в ОЗУ.

ЦОД обрабатывает запросы от 1700 клиентов, каждый клиент в среднем выполняет свой запрос один раз в сутки (распределение СВ подчинено экспоненциальному закону). На обработку одного запроса в ОЗУ сервера выделяется от 100 МБ до 3 ГБ памяти, длительность обработки занимает от 1 мин до 2 сут. Распределение СВ внутри указанных интервалов принять равномерным.

Если у сервера, принявшего запрос на обработку, отсутствует объем свободной оперативной памяти, достаточной для обслуживания запроса, то клиент в среднем через час (распределение СВ экспоненциальное) может повторить свой запрос с сохранением его параметров (требуемый объем ОЗУ и длительность обработки). Таких повторных запросов может быть от одного до пяти (вероятность первого повторного запроса составляет 80 %, второго – 40, третьего – 20, четвертого – 10 и пятого – 5 %). При повторях СБН может этот запрос перенаправить на другой сервер.

Подсчитать общее количество обслуженных запросов и отказов в обслуживании, процент отказов по отношению к общему количеству запросов в течение недели.

Подсказка: для определения объема доступной памяти используйте косвенную адресацию.

Вариант 12. Моделирование службы автодозвона клиентам сотовой компании

У сотовой компании имеется 20000 клиентов, к которым она периодически обращается с кейсами своих новых предложений (смена тарифного плана, подключение дополнительных услуг и т. д.). Эту функцию выполняет робот, выполняющий регулярные автодозвоны. Для звонка робот в случайном порядке выбирает данные одного из клиентов из своего списка. При ответе клиента робот автоматически его соединяет с любым свободным в данный момент оператором. У оператора при этом открывается карточка лида клиента с его контактными данными, данными текущего договора и подключенными услугами. Робот временно прекращает делать автодозвоны, если на линии отсутствуют свободные в этот момент операторы.

Операторы работают удаленно, длительность их рабочей смены составляет 8 ч. В среднем каждые 50 мин (распределение СВ экспоненциальное) операторы делают перерыв, длительность перерыва 10 мин (СВ также распределена экспоненциально). Когда у оператора открывается карточка лида, то он в силу разных причин не сразу приступает к общению с клиентом. Время задержки с началом разговора принять равным 10 с с экспоненциальным законом распределения. Клиент, с которым оператор не начал разговор, может прервать сеанс связи через 15 с (распределение СВ экспоненциальное). Среднее время, затрачиваемое на разговор с клиентами на обслуживание одного звонка, у разных операторов в зависимости от их квалификации разное и лежит в интервале $[0,8; 1,2]$ (постоянный закон распределения СВ). Длительность обслуживания звонка подчинено экспоненциальному закону.

Длительность автодозвона составляет 5 с. Если клиент не ответил в течение этого времени, то автодозвон продолжается и на второй, и на третий дни. Вероятность того, что клиент ответит на первый звонок составляет 40 %, на второй – 15 и на третий – 5 %. После третьей неудачной попытки дозвона клиент исключается из опросного списка. Он также будет исключен из списка, если его удалось подключить к оператору. Опрос по кейсу будет завершен тогда, когда опросной список будет убавлен до 1000 клиентов. В таком случае кейс заменяется

на новый, а опросный список при этом обновляется до полного списка клиентов, и затем процедура опроса начинается заново.

Процент клиентов, согласившихся на предложения кейса, составляет 5 %.

Считая, что у компании работает 4 оператора, выяснить число клиентов компании, до которых удалось дозвониться, или хотя бы делались попытки связаться с ними, в течение 5-дневной рабочей недели. Подсчитать число клиентов, согласившихся на предложения кейса. Выяснить процент клиентов, подключенных роботом к оператору, но так и не дождавшихся разговора с ним. Определить коэффициент загрузки операторов.

Вариант 13. Моделирование работы TCP протокола

Протокол TCP (Transmission Control Protocol, протокол управления передачей) является одним из основных сетевых протоколов и отвечает за успешную передачу сообщений. Сообщение, которое необходимо отправить в сеть, делится на фрагменты, которые после оснащения их TCP заголовками получают название сегментов. Сегменты отправляются не все одновременно, а порциями. Число сегментов, попавших в одну порцию, называется размером скользящего окна [11]. Сообщение считается полностью доставленным, если получатель получит все его сегменты. Получатель имеет обратную связь с отправителем, посылая ему квитанции ACK (acknowledgment, подтверждение). ACK может быть следующих типов:

- сегмент доставлен;
- сегмент не получен.

Сегмент считается недоставленным, если истек определенный тайм-аут. Это может быть связано с плохой работой сети, когда из-за низкой пропускной способности на каких-то ее участках сегмент задержался в накопителе сетевого устройства, или же сегмент был уничтожен из-за обнаруженной ошибки передачи.

Работа протокола TCP с применением скользящего окна заключается в том, что отправитель создает для сегментов два накопителя: общий и накопитель окна. Сегменты первой порции помещаются в накопитель окна и отправляются в сеть. Когда на какой-либо сегмент приходит квитанция ACK об его успешном приеме, то сегмент удаляется с обоих накопителей. На освободившееся место в накопитель окна помещается следующий сегмент из общего накопителя и отправляется в сеть. Передача идет до тех пор, пока оба накопителя не опустеют.

Если приходит квитанция о том, что какой-то сегмент не доставлен, то этот сегмент из накопителя окна повторно отправляется в сеть. Тем самым гарантируется, что все сегменты сообщения будут доставлены получателю.

Исходные данные для построения модели следующие:

- Размер сообщений (в килобайтах) определять по непрерывной функции, определяемой по четырем точкам с координатами (0; 0,1), (0,1; 10), (0,9; 100) и (1; 1000). Аргументом функции принять один из СЧА RN.
- Размер одного сегмента – 1,5 килобайта. Последний сегмент меньшего размера, так как он представляет собой остаток сообщения. При вычислении числа сегментов обеспечивать ему целые значения.
- Размер накопителя скользящего окна установить в 10 сегментов.
- По завершении приема всего сообщения, получатель может запросить отправку другого сообщения. Число таких сообщений случайно и лежит в интервале [5, 10], распределение СВ в интервале равномерное.
- Передача сегмента и квитанции по сети подчиняется экспоненциальному закону со средним значением 0,1 с, но не меньше, чем 0,03 с.
- При передаче по сети сегменты передаются независимо друг от друга и очереди не создают.
- Вероятность потери сегмента составляет 5 %.
- С вероятностью 1 % сегмент при своем перемещении по сети может попадать в зазоры со средним временем задержки 10 с (распределение СВ экспоненциальное).

- Тайм-аут принять равным 45 с.

Провести моделирование одного сеанса обмена сообщениями между двумя приложениями. Определить среднее время доставки сегментов.

Вариант 14. Моделирование работы колл-центра

Колл-центр принимает 6 звонков за одну минуту. Среднее время разговора составляет 2 мин. Закон распределения СВ в обоих случаях экспоненциальный. Абоненты после 5 мин ожидания в очереди завершают вызов, и звонок считается потерянным. Рабочая смена колл-центра длится 9 ч. С помощью модели определить какое минимальное количество операторов должно быть в штате, рассмотрев три сценария работы колл-центра:

- Не допустить ни одного потерянного звонка.
- Количество потерянных звонков должно быть не более 1 % (30 потерянных за смену).
- Придерживаться рекомендуемого для колл-центров значения $SL = 80/20$ (Service Level, уровень сервиса) – в течение 20 с давать ответы на 80 % звонков.

Для всех трех вариантов подсчитать процент абонентов, получивших ответ на свой вопрос в пределах 20 с и найти процент загрузки операторов.

Вариант 15. Моделирование работы двоичного автомата

Построить модель конечного автомата, представленного на рисунке 4. Множества алфавитов входного и выходного алфавитов представлены битами: $I = \{0, 1\}$ и $O = \{0, 1\}$, множество состояний представлено символами: $S = \{a, b, c, d\}$. Входное слово содержит случайные сочетания битов и вероятности их появления в слове одинаковы.

Смоделировать работу автомата из расчета, что на его вход поступит входное слово из 1000 битов, и построить гистограммы частот состояний и выходных битов. Считать, что в начале моделирования автомат будет находиться в состоянии 'а'.

Заключение

В данном учебном пособии последовательно рассмотрены основные понятия компьютерного моделирования, дана классификация моделей, их преимущества и недостатки. Изложена краткая теория конечных и вероятностных автоматов, систем массового обслуживания. Имитационное моделирование систем рассмотрено с применением языка GPSS. Все темы сопровождаются примерами проведения экспериментов с соответствующими моделями. Для оценки приобретенных учащимися навыков моделирования применяются контролирующие программы, размещенные на авторском сайте. Тема, посвященная имитационному моделированию, содержит 15 индивидуальных заданий для выполнения во внеаудиторное время.

Пособие предназначено студентам, изучающим курс «Компьютерное моделирование».

Рекомендуемая литература

1. Is a Markov Chain the Same as a Finite State Machine? – URL: <https://www.baeldung.com/cs/markov-chain-vs-finite-state-machine> (дата обращения: 11.07.2024).
2. Компьютерное моделирование: лабораторный практикум / сост. А.А. Алтаев, В.В. Бундаев – Улан Удэ: Изд-во ВСГТУ, 2007. – 84 с.
3. Имитационное моделирование на языке GPSS: метод. пособие по дисц. «Компьютерное моделирование» для спец. «Программное обеспечение ВТ и АС» / ВСГТУ; сост. А.А. Алтаев. – Улан-Удэ: Изд-во ВСГТУ, 2002. – URL: <http://simulation.su/uploads/files/default/2002-uch-posob-altaev-1.pdf> (дата обращения: 31.07.2024).
4. Авторский сайт Алтаева А.А. – URL: <http://www.altayev-aa.narod.ru/km3/> (дата обращения: 10.05.2024).
5. Учебник «Моделирование систем». Авторский сайт Мухина О.И. – URL: <https://stratum.ac.ru/education/textbooks/modelir/contents.html> (дата обращения: 15.07.2024).
6. Kendall's notation – Wikipedia. – URL: https://en.wikipedia.org/wiki/Kendall%27s_notation (дата обращения: 29.07.2024).
7. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World: учеб. пособие. – СПб.: БХВ-Петербург, 2004. – URL: https://books.4nmv.ru/books/modelirovanie_sistem_instrumentalnye_sredstva_gpss_world_3642706.pdf (дата обращения: 02.08.2024).
8. Имитационное моделирование на GPSS: учеб.-метод. пособие для студентов технических специальностей / Д.Н. Шевченко, И.Н. Кравченя; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель: Изд-во БелГУТ, 2007. – URL: <http://simulation.su/uploads/files/default/2007-uch-posob-snevchenko-kravchenko-1.pdf> (дата обращения: 03.08.2024).
9. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004. – URL: <http://simulation.su/uploads/files/default/2004-uch-posob-kudryavcev-1.pdf> (дата обращения: 03.08.2024).
10. Имитационное моделирование на языке GPSS: метод. указ. / О.Н. Евсеева, В.В. Шишкин. – Ульяновск: Изд-во УлГТУ, 1995. – 40 с. – URL: <http://simulation.su/uploads/files/default/1995-method-evseeva-schischkin-1.pdf> (дата обращения: 16.08.2024).
11. В поисках утерянного гигабита или немного про окна в TCP. – URL: <https://habr.com/ru/articles/336780/> (дата обращения: 22.08.2024).
12. Examples of Markov chains – Wikipedia. – URL: https://en.wikipedia.org/wiki/Examples_of_Markov_chains (дата обращения: 08.10.2024).

Java код программы генерации случайных чисел с применением равномерного закона распределения случайной величины для среднего значения $\bar{t} = 8$ и полуразмаха от среднего $B = 3$.

```
package test;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;

/**
 *
 * @author altaev
 */
public class Uniform {
    //Вычисление времени time с применением равномерного распределения
    вероятностей случайной величины

    public static void main(String[] args) throws IOException {
        Random rand = new Random();
        int time, max, min,
            averageTime = 8, // среднее значение
            B = 3;           // полуразмах от среднего
        File file = new File("C:/Temp/Times.txt");
        if (file.exists()) {
            file.delete();
        }
        final FileWriter writer = new FileWriter(file, true);
        min = averageTime - B;
        max = averageTime + B;
        for (int i = 0; i < 1000; i++) {
            time = rand.nextInt((max - min) + 1) + min;
            System.out.println("    " + time + " ");
            writer.append("" + time + "\n");
            writer.flush();
        }
    }
}
```

Java код программы генерации случайных чисел с применением нормального закона распределения случайной величины для среднего значения $\bar{t} = 8$ и среднеквадратичного отклонения $\text{standardDeviation} = 2$.

```

package test;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

/**
 *
 * @author altaev
 */
public class Normal {
    //Вычисление времени time с применением Функции нормированного нормального распределения
    //со средним значением averageTime и среднеквадратичным отклонением standardDeviation ( $\sigma$ )

    public static int time(int averageTime, long standardDeviation)
    {
        double[] arg = {0.0, 0.00003, 0.00135, 0.00621, 0.02275, 0.06681, 0.11507, 0.15866, 0.21186, 0.27425, 0.34458, 0.42074, 0.5, 0.57926, 0.65542, 0.72575, 0.78814, 0.84134, 0.88493, 0.93319, 0.97725, 0.99379, 0.99865, 0.99997, 1.0};
        double[] funNorm = {-5.0, -4.0, -3.0, -2.5, -2.0, -1.5, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0};
        double rnd = Math.random();
        int f = 0;
        for (int i = 0; i < arg.length; i++) {
            if (rnd < arg[i]) {
                // используем формулу линейного интерполирования
                f = (int) Math.round((funNorm[i - 1] + (funNorm[i] - funNorm[i - 1]) / (arg[i] - arg[i - 1]) * (rnd - arg[i - 1])) * standardDeviation) + averageTime);
                break;
            }
        }
        if (f == 0) { //time не может быть нулевым
            f = 1;
        }
        return f;
    }

    public static void main(String[] args) throws IOException {
        File file = new File("C:/Temp/Times.txt");
        if (file.exists()) {
            file.delete();
        }
        final FileWriter writer = new FileWriter(file, true);
        int tm;
    }
}

```



```
        for (int i = 0; i < 10000; i++) {
            tm = time(8, 1);
            System.out.println("" + tm);
            writer.append("" + tm + "\n");
            writer.flush();
        }
    }
```

Стандартные числовые атрибуты

Объект	СЧА	Назначение
Блоки	N	счетчик входов в блок
	W	счетчик текущего содержимого блока
Счетчик	TG1	счетчик завершения моделирования
Время	C1	значение относительного времени
Генераторы случайных чисел	RN	либо вещественное число из диапазона [0,000000; 0,999999], либо целое из диапазона [0; 999]
Транзакты	P	значение параметра
	PR	значение приоритета
	M1	время пребывания транзакта в модели
Одноканальные устройства	F	состояние ОКУ (1 – занят, 0 – свободен)
	FC	счетчик числа занятий
	FR	коэффициент использования ОКУ
	FT	среднее время задержки одного транзакта в ОКУ
Многоканальные устройства	R	число свободных каналов в памяти
	S	число занятых каналов
	SA	среднее содержимое
	SC	счетчик числа входов
	SR	коэффициент использования памяти
	SM	максимальное число каналов, одновременно занятых транзактами
	ST	среднее время задержки транзакта в памяти
Логические ключи	LS	состояние ключа (1 – Set, 0 – Reset)
Очереди	Q	текущее содержимое
	QA	среднее содержимое
	QC	счетчик общего числа входов
	QM	максимальное число транзактов, находившихся в очереди
	QZ	счетчик числа нулевых входов (транзакт прошел очередь без задержки в ней)
	QT	среднее время пребывания в очереди одного транзакта
	QX	среднее время пребывания без учета нулевых входов
Сохраняемые величины	X	значение сохраняемой величины
Таблицы	TB	среднее значение взвешенных входов
	TC	количество невзвешенных входов
	TD	стандартное отклонение невзвешенных входов
Переменные	V	значение переменной
	BV	значение булевской переменной
Функции пользователя	FN	значение функции

Принятые в пособии сокращения

СВ – случайная величина;
СЧА – стандартный числовой атрибут;
ОКУ – одноканальное устройство, канал;
МКУ – многоканальное устройство, память.

Учебное издание

*Анатолий Александрович АЛТАЕВ
Эржена Цыренжаповна ДОРЖИЕВА
Иван Васильевич СТАРИНСКИЙ
Инна Степановна ТУЛОХОНОВА*

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Учебное пособие

Редактор *Е.В. Белоплотова*

Подписано в печать 11.11.2024. Формат 60×84 1/8.

Усл. печ. л. 9,76. Тираж 300 экз. Заказ № 142.

Издательство ВСГУТУ

670013, г. Улан-Удэ, ул. Ключевская, д. 40В, строение 1.

Отпечатано в типографии ВСГУТУ

670013, г. Улан-Удэ, ул. Ключевская, д. 40В, строение 9.

ISBN 978-5-907942-01-1



9 785907 942011 >