

<b>Введение .....</b>	<b>3</b>
<b>Глава 1. Принципы моделирования случайных элементов .....</b>	<b>5</b>
<i>1.1. Моделирование случайных величин .....</i>	<i>7</i>
1.1.1. Моделирование базовых случайных величин (БСВ) .....	7
1.1.2. Статический анализ точности моделирования .....	14
1.1.3. Задания .....	15
<i>1.2. Моделирование дискретных случайных величин (ДСВ) .....</i>	<i>17</i>
1.2.1. Понятие ДСВ. Общая схема алгоритма моделирования ..	17
1.2.2. Алгоритмы моделирования для дискретных распределений	18
1.2.3. Графический анализ точности моделирования .....	22
1.2.4. Задания .....	23
<i>1.3. Моделирование непрерывных случайных величин (НСВ) .....</i>	<i>26</i>
1.3.1. Понятие НСВ. Универсальные методы моделирования	НСВ.26
1.3.2. Тесты проверки точности моделирования НСВ.....	29
1.3.3. Алгоритмы моделирования для основных непрерывных	распределений.....
1.3.4. Задания .....	48
<b>Глава 2. Метод Монте-Карло и его применения.....</b>	<b>53</b>
2.1. Общая схема метода Монте-Карло .....	53
2.2. Вычисление определенного интеграла методом Монте-Карло	54
2.3. Решение системы алгебраических уравнений методом Монте-	Карло .....
2.4. Задания.....	67
<b>Глава 3. Система моделирования GPSS.....</b>	<b>68</b>
3.1. Краткие сведения из теории .....	68
3.2. Основные правила и операторы языка GPSS .....	69
3.2.1. Структура операторов GPSS .....	70
3.2.2. Основные операторы языка GPSS.....	70
3.3. Основные блоки GPSS/PC и связанные с ними объекты.....	74
3.3.1. Блоки, связанные с транзактами .....	74
3.3.2. Блоки, связанные с аппаратными объектами .....	85
3.3.3. Блоки для сбора статистических данных.....	90
3.3.4. Блоки, изменяющие маршруты транзактов .....	94
3.3.5. Блоки, работающие с памятью .....	99

3.3.6.Блоки для работы со списками пользователя .....	103
3.4.Управляющие операторы GPSS/PC .....	108
3.5.Некоторые приемы конструирования GPSS-моделей .....	112
3.5.1.Косвенная адресация.....	112
3.5.2.Обработка одновременных событий.....	115
3.6.Команды GPSS/PC и технология работы с пакетом .....	118
3.6.1.Загрузка интегрированной среды.....	118
3.6.2.Ввод новой модели .....	118
3.6.3.Редактирование текста модели.....	119
3.6.4.Запись и считывание модели с диска.....	120
3.6.5.Прогон модели и наблюдение за моделированием .....	120
3.6.6.Получение и интерпретация стандартного отчета .....	125

## **Глава 4. Моделирование сложных систем на основе GPSS .... 131**

4.1.Моделирование работы производственного участка .....	131
4.1.1.Алгоритм решения .....	131
4.1.2.Имитационная модель системы .....	131
4.1.3.Анализ результатов имитационного моделирования ...	134
4.2.Моделирование работы участка цеха .....	138
4.2.1.Имитационная модель системы .....	140
4.2.2.Анализ результатов имитационного моделирования ...	142
4.3.Порядок оформления отчета.....	150
4.3.1.Содержание отчета.....	150
4.3.2.Создание выходного файла .....	151
4.3.3.Описание элементов выходного файла.....	151

## **Глава 5. Задания лабораторного практикума по GPSS..... 156**

## **Глава 6. Статистическое моделирование в финансовом анализе.186**

6.1.Моделирование наращенных сумм и современных величин платежа.187

6.2.Моделирование величины реальной наращенной суммы платежей с учетом инфляции.192

6.3.Моделирование обменного курса валют при двойной конверсии валют.195

## **Приложение ..... 197**

## **Литература ..... 204**

# Введение

Важнейшей, универсальной целью рациональной деятельности человека является создание высокоэффективных систем, объектов, явлений технологий процессов в экономике, технике, экологии, производстве, обществе. Высокая эффективность предполагает достижения экстремума некоторых числовых характеристик – показателей эффективности. В связи с этим необходимы математические методы. Но к реальному объекту, явлению, системе эти методы непосредственно не применимы. Необходимо, прежде всего, построить математическую модель системы, то есть приближенное описание системы с помощью математических соотношений. *Математическое моделирование (ММ)* представляет собой исследование математической модели, изучение ее и перенос полученных сведений на моделируемую систему.

Математическое моделирование существует фактически с тех далеких времен, когда математический аппарат начинал применяться для решения практических задач. Однако, эти попытки часто заходили в тупик из-за сложности моделей (нелинейные уравнения; большое число переменных, параметров, уравнений; малая априорная информация и др.) и невозможности их исследования традиционными аналитическими методами. С появлением и совершенствованием ЭВМ ситуация коренным образом изменилась. С математической моделью начали экспериментировать на ЭВМ. Появились новые понятия и подходы: вычислительный эксперимент, имитационное моделирование, статистическое моделирование, метод Монте-Карло. В настоящее время *сущность ММ* состоит в замене исходной (исследуемой, управляемой, эксплуатируемой) системы ее математической моделью и дальнейшем экспериментировании с этой моделью при помощи вычислительно логических алгоритмов.

Для компьютерного моделирования характерно, что математическая модель системы представлена в виде программы для ЭВМ – компьютерной модели, позволяющей производить с ней вычислительные эксперименты. В зависимости от математического аппарата, используемого при построении, и способа организации вычислительных экспериментов, можно выделить три взаимосвязанных вида моделирования: численное, статистическое и имитационное. При *численном моделировании* для построения компьютерной модели используются методы вычислительной математики, а вычислительный эксперимент заключается в численном решении некоторых математических уравнений при заданных

значениях параметров и начальных условий. *Имитационное моделирование* (ИМ) – это вид компьютерного моделирования, для которого характерно воспроизведение на ЭВМ (имитация) процесса функционирования исследуемой сложной системы. При этом имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры, последовательности протекания во времени, что позволяет получить информацию о состоянии системы  $S$  в заданные моменты времени. *Статистическое моделирование* (СМ) – это вид компьютерного моделирования, позволяющий получать статистические данные о процессах в моделируемой системе  $S$ .

# Глава 1. Принципы моделирования случайных элементов

При компьютерном моделировании случайных процессов и систем возникает необходимость в моделировании различных случайных элементов : случайных величин, случайных векторов, процессов, полей, множеств.

Моделирование на ЭВМ случайного элемента подчиняется следующим двум основным принципам.

1. Сходство между случайным элементом-оригиналом  $\Xi$  и его моделью  $\Xi'$  состоит в совпадении (близости) вероятностных законов распределения или их числовых характеристик;
2. Всякий случайный элемент  $\Xi$  определяется («конструируется») как некоторая функция  $f$  от простейших случайных элементов, называемых базовыми случайными величинами (БСВ)  $\alpha_1, \dots, \alpha_r$  (см. раздел «Моделирование БСВ»):

$$\Xi' = f(\alpha_1, \dots, \alpha_r) \quad (1)$$

Таким образом, задача моделирования произвольного случайного элемента  $\Xi^*$  разбивается на следующие подзадачи.

1. Моделирование на ЭВМ независимых БСВ  $\alpha_1, \dots, \alpha_r$ ;
2. Осуществление функционального преобразования (1) с помощью соответствующей функции  $f$ .

Для моделирования одного и того же случайного элемента  $\Xi$  может быть предложено несколько вариантов функциональных преобразований (1). Предпочтение отдаётся варианту  $f(\cdot)$ , использующему меньшее число  $r$  БСВ для моделирования одной реализации  $\Xi'$  случайного элемента  $\Xi$ , то есть требующему меньших

вычислительных затрат. Такому варианту соответствует максимальное значение коэффициента использования БСВ  $k$  :

$$k = 1/r, 0 < k < 1.$$

В качестве объектов моделирования в пакете СТАТМОД используются следующие случайные элементы :

- случайные (скалярные) величины;
- случайные векторы;
- случайные процессы.

Графический анализ результатов моделирования включает следующие графики :

- визуализация выборки (иллюстрирует зависимость  $\xi_t$  от  $t$  и может использоваться для графической поддержки критериев серий);
- графики эмпирической и теоретической функций распределения (графики функций  $\hat{F}_\xi(x)$  и  $F_\xi(x)$ );
- графики эмпирической и теоретической функций вероятности (графики функций  $\hat{P}_\xi(x)$  и  $P_\xi(x)$ )).

Два последних графика позволяют осуществлять сравнительный анализ теоретического и эмпирического распределений выборки и могут служить в качестве графической поддержки критериев согласия.

## 1.1. Моделирование случайных величин

### 1.1.1. Моделирование базовых случайных величин (БСВ)

#### Понятие БСВ

Базовой случайной величиной (БСВ) непрерывная СВ  $\alpha$ , равномерно распределённая на полуинтервале  $[0,1)$ .

Равномерный на  $[0,1)$  закон обозначается  $R(0,1)$ .

БСВ имеет следующие функциональные и числовые характеристики :

- функция распределения :

$$F_{\alpha}(x) = \begin{cases} 0, & x \leq 0, \\ x, & 0 < x < 1, \\ 1, & x \geq 1; \end{cases}$$

- плотность распределения :

$$P_{\alpha}(x) = \begin{cases} 1, & 0 \leq x \leq 1, \\ 0, & \text{иначе;} \end{cases}$$

- математическое ожидание (среднее значение) :

$$\mu = E\{\alpha\} = 1/2;$$

- дисперсия :

$$\sigma^2 = D\{\alpha\} = E\{(\alpha - \mu)^2\} = 1/12.$$

## Датчики БСВ

Для моделирования на ЭВМ реализаций БСВ используются специальные программы, называемые программными датчиками БСВ.

В основе программных датчиков БСВ лежат рекуррентные формулы вида :

$$\alpha_t = \varphi(\alpha_{t-1}, \dots, \alpha_{t-p}), t = 1, 2, \dots, \quad (2)$$

где  $\alpha_{1-p}, \alpha_{2-p}, \dots, \alpha_0 (p \geq 1)$  -- заданные «стартовые» значения. Соотношение (2) описывает детерминированный алгоритм, однако при соответствующем подборе преобразования  $\varphi(\cdot)$  получаемые на его основе псевдослучайные числа  $\{\alpha_t\}$  по своим функциональным и числовым характеристикам близки к БСВ.

Алгоритмы моделирования вида (2) обладают общим недостатком : начиная с некоторого момента  $t_0$  последовательность псевдослучайных чисел образует цикл, который повторяется бесконечное число раз. Длина  $T$  циклически повторяющейся последовательности называется периодом датчика БСВ.

Период  $T$  и коэффициент использования БСВ  $k$  являются основными показателями качества программных датчиков БСВ. Лучшим датчикам соответствуют большие значения  $T$  и  $k$ .

В пакете СТАТМОД реализованы два программных датчика БСВ, основанные на мультипликативном конгруэнтном методе и методе Макларена-Марсальи.

### Мультипликативный конгруэнтный метод

Согласно этому методу псевдослучайная последовательность реализаций  $\alpha_1, \dots, \alpha_n$  БСВ  $\alpha$  определяется по рекуррентным формулам :

$$\alpha_t = \alpha_t^* / M, \alpha_t^* = \{\beta \alpha_{t-1}^*\} \bmod M (t = 1, 2, \dots), \quad (3)$$

где  $\beta, M, \alpha_0^*$  -- параметры датчика (натуральные числа) :  $\beta$  -- множитель ( $\beta < M$ ),  $M$  -- модуль,  $\alpha_0^* \in \{1, \dots, M-1\}$  -- стартовое значение (нечётное число).

Операция  $y = \{z\} \bmod M$  означает вычет числа  $z$  по модулю  $M$  :

$$y = z - M \cdot [z / M],$$

где  $[x]$  -- целая часть числа  $x$ .

Период датчика  $T \leq M-1$ ; коэффициент использования БСВ  $k=1$ . значения параметров  $\beta, \alpha_0^*$  определяются из условия максимума



периода  $T$ . значение  $M$  зависит от способа представления целых чисел в ЭВМ. Типовые значения параметров :

$$M = 2^{31} = 2147483648, \alpha_0^* = \beta = 65539.$$

## Метод Макларена-Марсальи

Метод основан на комбинировании двух простейших программных датчиков БСВ (например, мультипликативных конгруэнтных).

Пусть  $\{\beta_i\}, \{c_i\}$  – псевдослучайные последовательности, порождаемые независимо работающими датчиками;  $\{\alpha_i\}$  -- результирующая псевдослучайная последовательность реализаций БСВ;  $V = \{V(0), V(1), \dots, V(K-1)\}$  -- вспомогательная таблица  $K$  чисел.

Процесс вычисления  $\{\alpha_i\}$  включает следующие этапы :

- первоначальное заполнение таблицы  $V$  :

$$V(i) = b_i, i = \overline{0, K-1};$$

- случайный выбор из таблицы :

$$\alpha_i = V(s), s = [c_i \cdot K];$$

- обновление табличных значений :

$$V(s) = b_{t+K}, t = 0, 1, 2, \dots$$

Данный метод позволяет ослабить зависимость между членами псевдослучайной последовательности  $\{\alpha_i\}$  и получить сколь угодно большие значения её периода  $T$  при условии, что периоды  $T_1, T_2$  исходных датчиков являются взаимно простыми числами. Коэффициент использования БСВ для данного датчика  $k=1/2$  (за исключением первой реализации, для моделирования которой используется  $K+1$  реализация).

## Специальные тесты проверки точности моделирования БСВ

Поскольку БСВ используются при моделировании всех других случайных элементов, точность моделирования для датчиков БСВ должна проверяться особенно тщательно. В терминах статистической проверки гипотез данная задача формулируется следующим образом.

По выборке значений  $A = \{a_1, \dots, a_n\}$ , полученной в результате  $n$ -кратного обращения к датчику БСВ необходимо проверить гипотезу

$H_0$ : выборочные значения  $a_1, \dots, a_n$  являются реализациями независимых СВ  $\alpha_1, \dots, \alpha_n$ , равномерно распределённых на  $[0,1)$ ; конкурирующая гипотеза  $H_1 = \overline{H_0}$ .

Для этой цели наряду с традиционными статистическими критериями проверки точности моделирования СВ (критериями согласия, критериями серий и др.), применяется ряд специальных тестов.

В пакете СТАТМОД реализованы следующие тесты данной группы:

- тест «совпадения моментов»;
- тест «ковариация»;
- тест «равномерность двумерного распределения».

## Тест «совпадения моментов»

Пусть в результате  $n$ -кратного обращения к датчику БСВ получена выборка значений  $A = \{a_1, \dots, a_n\}$ . Известно, что БСВ имеет среднее значение  $\mu = 1/2$  и дисперсию  $\sigma^2 = 1/12$ . Обозначим:

$$\xi_1 = m - 1/2, \quad \xi_2 = s^2 - 1/12$$

-- случайные отклонения выборочных оценок

$$m = \frac{1}{n} \sum_{i=1}^n a_i, \quad s^2 = \frac{1}{n-1} \sum_{i=1}^n (a_i - m)^2$$

от истинных характеристик  $\mu, \sigma^2$ .

Тест «совпадения моментов» -- это программа для ЭВМ, реализующая статистические критерии проверки по выборке  $A$  гипотез:

$$H_0: \mu = 1/2, \quad H_1: \mu \neq 1/2; \quad (4)$$

$$H_0: \sigma^2 = 1/12, \quad H_1: \sigma^2 \neq 1/12; \quad (5)$$

Решающее правило для проверки гипотез (4), (5) имеет вид:

$$\text{принимается} \begin{cases} H_0, c_i(n) \cdot |\xi_i| < \Lambda \\ H_1, \text{иначе,} \end{cases} \quad (6)$$

где :  $i=\{1$  — для соотношений (4);  $2$  — для соотношений (5)};

$$c_1(n) = \sqrt{12n}, \quad c_2(n) = \frac{n-1}{n} (0.0056n^{-1} + 0.0028n^{-2} - 0.0083n^{-3})^{\frac{1}{2}} --$$

нормировочные множители;  $\Delta$  -- порог критерия.

Если  $H_0$  верна, а  $n \gg 1$  (практически  $n \geq 20$ ), то в силу ЦПТ :  $c_i(n)\xi_i \sim N_1(0,1)$  (распределено приближённо по стандартному нормальному закону). С учётом этого из ограничения на вероятность ошибки первого рода :

$$P\{H_0 | H_1\} = P\{c_i(n) | \xi_i | \geq 0 | H_0\} = \varepsilon (0 < \varepsilon < 1),$$

находится выражение для порога критериев :

$$\Delta = \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right),$$

где  $\Phi^{-1}(\cdot)$  -- квантиль стандартного нормального закона,  $\varepsilon$  -- заданный уровень значимости.

В пакете СТАТМОД предполагается эквивалентной формы решающих правил (6), связывающей задаваемый пользователем уровень значимости  $\varepsilon$  и вычисляемые по выборке  $A$  критические вероятности  $P_i$  ( $P$ -значения) :

$$\text{принимается} \begin{cases} H_0, \varepsilon < P_i, \\ H_1, \varepsilon \geq P_i, \end{cases}$$

где

$$P_i = 2(1 - \Phi(c_i(n) | \xi_i |)), i = \overline{1,2}.$$

## Тест «ковариация»

Ковариационной функцией случайной последовательности  $\alpha_1, \dots, \alpha_n$  называется функция целочисленной переменной  $\xi \in \{0, 1, \dots, n-1\}$  :

$$R(j) = E\{(\alpha_1 - E\{\alpha_1\})(\alpha_{1+j} - E\{\alpha_{1+j}\})\} = E\{\alpha_1 \cdot \alpha_{1+j}\} - E\{\alpha_1\} \cdot E\{\alpha_{1+j}\}.$$

Если  $\alpha_1, \dots, \alpha_n$  -- независимые, одинаково распределённые по закону  $R(0,1)$  случайные величины, то  $\alpha_1$  и  $\alpha_{1+j}$  независимы для любого  $j \geq 1$  и следовательно :

$$R(j) = \begin{cases} 1/12, & j = 0, \\ 0, & j \geq 1. \end{cases} \quad (7)$$

Пусть  $\hat{R}(j)$  -- оценка  $R(j)$  по выборке  $A = \{a_1, \dots, a_n\}$ , полученной в результате  $n$ -кратного обращения к исследуемому датчику :

$$\hat{R}(j) = \frac{1}{n-j-1} \sum_{i=1}^{n-j} a_i \cdot a_{i+j} - \frac{n}{n-1} m^2, j = 0, 1, \dots, t,$$

где  $1 < t \ll n, m$  -- выборочное среднее. Заметим, что  $\hat{R}(j) = s^2$  ( $s$  -- выборочная дисперсия).

Тест «ковариация» позволяет проверить свойство (7) (гипотезу  $H_0$ ) для последовательности  $a_1, \dots, a_n$  и описывается следующим решающим правилом :

$$\text{принимается} \begin{cases} H_0, |\hat{R}(j) - R(j)| < \frac{c_j \Delta}{12\sqrt{n-1}}, \\ H_1, \text{иначе,} \end{cases} \quad (8)$$

где :  $c_0 = \sqrt{2}$ ,  $c_j = 1$  для  $j \geq 1$ ;  $\Delta$  - порог, определённый для заданного уровня значимости  $\varepsilon$  по формуле :

$$\Delta = \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right).$$

В пакете СТАТМОД предполагается использование эквивалентной формы правила (8) :

$$\text{принимается} \begin{cases} H_0, \varepsilon < P_j, j = 0, 1, \dots, t; \\ H_1, \text{иначе,} \end{cases}$$

где

$$P_j = 2 \left( 1 - \Phi \left( \frac{12\sqrt{n-1} |\hat{R}(j) - R(j)|}{c_j} \right) \right).$$

Представим (8) в виде:

$$\text{принимается} \begin{cases} H_0, R_-(j) < R(j) < R_+(j), j = 0, 1, \dots, t; \\ H_1, \text{иначе.} \end{cases} \quad (9)$$

Здесь  $R_{\pm}(j) = \hat{R}(j) \pm \frac{c_j \Delta}{12\sqrt{n-1}}$  - верхняя и нижняя доверительные границы для  $R(j)$ , соответствующие доверительной вероятности  $\beta = 1 - \varepsilon$ .

Решающее правило в форме (9) удобно использовать при визуальном анализе графиков ковариационной функции  $R(j)$  и её оценки  $\hat{R}(j)$ .

## Тест «равномерность двумерного распределения»

Из выборочных значений  $a_1, \dots, a_n$  полученных в результате  $n$ -кратного обращения к датчику БСВ построим  $m = [n/2]$  ( $[z]$ -целая часть числа  $z$ ) векторов :

$$(a_1, a_2), (a_3, a_4), \dots, (a_{2(m-1)+1}, a_{2m}). \quad (10)$$

Единичный квадрат  $\Xi \subset R^2$  с центром в точке  $c = (0.5, 0.5) \in R^2$  разобьём на  $k$  ячеек :

$$\begin{aligned} \Xi_i &= \{x : x \in \Xi, r_{i-1} \leq |x - c| < r_i\}, i = \overline{1, k-1}; \\ \Xi_k &= \Xi \setminus \bigcup_{i=1}^{k-1} \Xi_i, \end{aligned}$$

где  $r_0 = 0 < r_1 < \dots < r_{k-1} \leq 0.5$  -- произвольные вещественные числа.

Вычислим частоты  $\{m_i\}$  попадания  $m$  точек с координатами (10) в  $k$  ячеек  $\{\Xi_i\} \left( \sum_{i=1}^k m_i = m \right)$  гиперкуба.

Если  $\alpha_1, \dots, \alpha_n$  -- независимые, одинаково распределённые по закону  $R(0,1)$  случайные величины, то теоретические вероятности  $\{p_i\}$  попадания точек с координатами  $(a_{2(j-1)+1}, a_{2j}), j = \overline{1, m}$  в ячейки  $\{\Xi_i\}$  равны площадям этих ячеек :

$$p_i = \begin{cases} \pi(r_i^2 - r_{i-1}^2), i = \overline{1, k-1} \\ 1 - \pi r_{k-1}^2, i = k. \end{cases}$$

Описываемый тест используется для проверки гипотезы  $H_0$  о равномерности двумерного распределения векторов  $\{(a_{2(j-1)+1}, a_{2j})\}$  и представляет собой следующее решающее правило:

$$\text{принимается} \begin{cases} H_0, x^2 < \Delta, \\ H_1, x^2 \geq \Delta, \end{cases} \quad (11)$$

где в случае истинной гипотезы  $H_0$  и  $n \rightarrow \infty$  статистика

$$x^2 = \sum_{i=1}^k \frac{(m_i - mp_i)^2}{mp_i} \quad (12)$$

имеет  $\chi^2$  – распределение с  $k - 1$  степенями свободы, а порог  $\Delta$  определяется как квантиль этого распределения:  $\Delta = F^{-1}(1 - \varepsilon)$ , где  $\varepsilon$  – заданный уровень значимости.

В пакете СТАТМОД предполагается использование эквивалентной формы правила (11):

$$\text{принимается} \begin{cases} H_0, \varepsilon < P, \\ H_1, \varepsilon \geq P, \end{cases}$$

где  $P$  - значение вычисляется по формуле:  $P = 1 - F(x^2)$ , здесь  $F(\cdot)$ -функция распределения статистики (12).

### **1.1.2. Статический анализ точности моделирования**

Тесты проверки точности моделирования БСВ допускают графическую интерпретацию, удобную для быстрого визуального анализа качества смоделированной выборки  $A = \{a_1, \dots, a_n\}$  реализаций БСВ.

В пакете СТАТМОД реализованы следующие графики:

1. диаграмма рассеяния (иллюстрирует зависимость между  $a_i$  и  $a_{i-1} (i = \overline{2, n})$ ; для проверки факта наличия зависимости используется тест «ковариация»);
2. гистограмма и плотность распределения (позволяет осуществить сравнительный анализ теоретического и эмпирического распределения выборки  $A$ ; для проверки согласия распределения  $A$  с равномерным законом используются критерии согласия);

3. график корреляционной функции с указанием доверительных границ (служит для графической поддержки теста «ковариация»);
4. визуализация выборки (иллюстрирует зависимость  $a_i$  от  $t$  ( $t = \overline{1, n}$ ) и может использоваться для графической поддержки критериев серий).

### 1.1.3. Задания

1. Осуществить моделирование  $n=1000$  реализаций БСВ с помощью мультипликативного конгруэнтного метода (МКМ-датчика) при следующих значениях параметров датчика:

$$\beta = a_0^* = 2^m + 3, m = 4, 8, 12, 16, 20, 24, 28.$$

(т.е.  $\beta \in \{19, 259, 4099, 65539, 1048979, 16777219, 268435459\}$ )

2. Сравнить эти датчики по точности, используя тесты проверки точности моделирования. Найти значение  $m^*$ , при котором достигается максимальная точность моделирования.
3. Выполнить задание 1 при следующих значениях параметров датчика:

$$\beta = a_0^* = 5^{2m+1}, m = 0, 1, 2, 3, 4, 5, 6.$$

(т.е.  $\beta \in \{5, 125, 3125, 78125, 1953125, 48828125, 1220703125\}$ )

4. Для МКМ-датчика при  $\beta = 65539$  исследовать зависимость точности моделирования и величины периода  $T$  от «стартового» числа  $a_0^*00000000$ . Положить  $a_0^* \in \{3, 19, 259, 4099, 65539, 1048979, 2147483647\}$

5. Сравнить по точности датчики, реализующие МКМ при  $\beta = 65539$  и  $\beta = 78125$

$$(a_0^* = \beta).$$

6. Сравнить по точности МКМ-датчики при различных значениях  $\beta$  из заданного в пакете множества возможных значений. Положить  $a_0^* = \beta$ .

7. Датчики Макларена-Марсальи (ММ-датчики)  $M_1$  и  $M_2$  используют в качестве простейших датчики  $D_1$  и  $D_2$ , реализующие МКМ при следующих значениях параметров:

$$M_1. \quad D_1 : \beta = a_0^* = 65539,$$

$$D_2 : \beta = 65539, a_0^* = 256959681;$$

$$M_2. \quad D_1 : \beta = a_0^* = 1078318301,$$

$$D_2 : \beta = 1078318301, a_0^* = 860574881;$$

8. Сравнить по точности датчики  $M_1$  и  $M_2$ .

9. Сравнить по точности и быстродействию ММ-датчик М1 из задания 6 и МКМ-датчик с параметрами  $\beta = a_0^* = 65539$ .

1.



## 1.2. Моделирование дискретных случайных величин (ДСВ)

### 1.2.1. Понятие ДСВ. Общая схема алгоритма моделирования

Дискретной случайной величиной (ДСВ) называется случайная величина  $\xi$ , имеющая дискретное распределение вероятностей, определяемое дискретным множеством значений  $c_1, c_2, \dots, c_N$  ( $N \leq \infty$ ) и заданными вероятностями значений

$$p_i = P\{\xi = c_i\}, \sum_{i=1}^N p_i = 1. \quad (13)$$

Основными функциональными и числовыми характеристиками ДСВ  $\xi$  являются ( $x \in \{c_1, c_2, \dots, c_N\}$ ):

- функция распределения:

$$F_{\xi}(x) = P\{\xi < x\} = \sum_{i=1}^N p_i \cdot 1(x - c_i),$$

где  $1(z) = \begin{cases} 0, & z \leq 0, \\ 1, & z > 0; \end{cases}$

- функции вероятности:

$$P_{\xi}(x) = P\{\xi = x\} = \prod_{i=1}^N p_i^{\delta}$$

где  $\delta = \delta(x, c_i)$  – символ Кронекера;

- математическое ожидание:

$$\mu = E\{\xi\} = \sum_{i=1}^N c_i \cdot p_i;$$

- дисперсия:

$$\sigma^2 = D\{\xi\} = \sum_{i=1}^N (c_i - \mu)^2 p_i.$$

Алгоритм моделирования ДСВ  $\xi$ , заданной распределением (13), состоит из вычисления вспомогательного вектора  $q = (q_1, \dots, q_N) = (p_1, p_1 + p_2, \dots, p_1 + \dots + p_{N-1}, 1)$  и двух шагов, повторяющихся при каждом обращении к алгоритму:

1. Моделирование с помощью датчика БСВ реализации  $a$ .
2. Принятие решения о том, что реализацией  $\xi$  является  $x$ , определяемое по правилу:

$$x = c_i, \text{ если } q_{i-1} \leq a < q_i, q_0 = 0, i = \overline{1, N}.$$

На практике для описания ДСВ используются модельные дискретные законы распределения с числом параметров  $N' \ll N$ . Это позволяет построить более экономичные и точные алгоритмы моделирования ДСВ.

В пакете СТАТМОД реализованы алгоритмы моделирования ДСВ для таких модельных законов распределения как:

1. распределение Бернулли;
2. биномиальное распределение;
3. геометрическое распределение;
4. отрицательное биномиальное распределение;
5. гипергеометрическое распределение;
6. распределение Пуассона;
7. дискретное равномерное распределение.

## Проверка точности моделирования ДСВ

Проверка точности моделирования ДСВ включает проверку гипотез согласия распределения смоделированной выборки  $A = \{\xi_1, \dots, \xi_n\}$  заданному модельному закону распределения, а также проверку случайности и независимости выборочных значений  $\xi_1, \dots, \xi_n$ . Для этих целей используются соответственно описанные выше статистические критерии согласия, а также критерии серий.

### 1.2.2. Алгоритмы моделирования для дискретных распределений

#### Распределение Бернулли

ДСВ  $\xi$  имеет распределение Бернулли  $Bi(1, p)$ , если:  $\xi \in \{0, 1\}$ ,  $P\{\xi = 1\} = p$ ,  $P\{\xi = 0\} = 1 - p$ , где  $p \in (0, 1)$  – параметр распределения.

Характеристики распределения Бернулли ( $x \in \{0, 1\}$ ):

- функция распределения:

$$F_{\xi}(x) = \sum_{i=0}^x p^i (1-p)^{1-i};$$

- функция вероятности:

$$P_{\xi}(x) = p^x (1-p)^{1-x};$$

- среднее значение:  $\mu = p$ ;
- дисперсия:  $\sigma^2 = p(1-p)$ .

## Содержательная интерпретация

Распределение Бернулли описывает случайный эксперимент (испытание Бернулли) с двумя исходами: успех ( $\xi = 1$ ) и неудача ( $\xi = 0$ ), причем вероятность успеха равна  $p$ .

## Алгоритм моделирования

Алгоритм моделирования одной реализации случайной величины Бернулли состоит из двух шагов:

1. Моделирование реализации БСВ.
2. Принятие решения о том, что реализацией  $\xi$  является значение  $x$  определяемое по правилу:

$$x = \begin{cases} 1, a \leq p, \\ 0, a > p. \end{cases}$$

Коэффициент использования БСВ  $k = 1$ .

## Биномиальное распределение

ДСВ  $\xi$  имеет биномиальное распределение  $Bi(m, p)$ , если:  
 $\xi \in \{0, 1, \dots, m\}$ ,  $P\{\xi = i\} = C_i^m p^i (1-p)^{m-i}$ ,  $i \in \{0, 1, \dots, m\}$ .

Параметры распределения:  $m$  – натуральное число;  $p \in (0, 1)$ .

Характеристики распределения  $Bi(m, p)$  ( $x \in \{0, 1, \dots, m\}$ ):

- функция распределения:

$$F_{\xi}(x) = \sum_{i=0}^x C_i^m p^i (1-p)^{m-i};$$

- функция вероятности:

$$P_{\xi}(x) = C_x^m p^x (1-p)^{m-x};$$

- среднее значение:  $\mu = m \cdot p$ ;
- дисперсия:  $\sigma^2 = mp(1-p)$ .

## Содержательная интерпретация

Биномиальная СВ  $\xi$  – это число успехов в  $m$  независимых испытаниях Бернулли, если вероятность успеха в каждом испытании равна  $p$ .

## Алгоритмы моделирования

1. Метод браковки. Алгоритм моделирования реализации биномиальной СВ  $\xi$  по методу браковки состоит из двух шагов:

- (а) Моделирование  $m$  реализаций  $a_1, \dots, a_m$  БСВ.
- (б) Принятие решения о том, что реализацией  $\xi$  является значение  $x$ , вычисляемое по формуле:

$$x = \sum_{i=1}^m 1(p - a_i),$$

$$\text{где } 1(z) = \begin{cases} 0, z \leq 0; \\ 1, z > 0. \end{cases}$$

Таким образом,  $x$  – количество значений из  $\{a_i\}$ , меньших  $p$ .

Коэффициент использования БСВ  $k = \frac{1}{m}$ .

2. Использование геометрического распределения. Описываемый алгоритм моделирования биномиальной СВ  $\xi$  обладает при малых значениях  $p$  (практически при  $p < 0.1$ ) высоким быстродействием. Алгоритм состоит из двух шагов:

- (а) Моделирование  $v$  реализаций  $\eta_1, \dots, \eta_v$  ДСВ, распределенной по геометрическому закону  $G(p)$ , где  $v$  – случайная величина, определяемая соотношением:

$$v = \min \left\{ k : \sum \eta_i' > m, k = 0, 1, 2, \dots \right\},$$

где  $\eta_i'$  получается из  $\eta_i$  округлением до ближайшего целого значения.

- (б) Принятие решения о том, что реализацией  $\xi$  является величина  $x = v - 1$ .

Таким образом,  $x$  – число серий независимых испытаний до первого успеха (с вероятностью успеха, равной  $p$ ) в предложении, что общее число испытаний равно  $m$  (то есть,  $x$  – число успехов в  $m$  независимых испытаниях).

Коэффициент использования БСВ – случайная величина  $k = \frac{1}{\xi + 1}$ .

Биномиальное распределение допускает аппроксимацию другими законами распределения:

- нормальным законом  $N_1(mp, mp(1-p))$ , если  $mp(1-p) > 5, 0.1 \leq p \leq 0.9$  либо  $mp(1-p) > 25$  для любого  $p$ ;
- законом Пуассона  $\Pi(mp)$ , если  $p < 0.1$ .

## Геометрическое распределение

ДСВ  $\xi$  имеет геометрическое распределение  $G(p)$ , если:  $\xi \in \{1, 2, \dots\}$ ,  $P\{\xi = i\} = p \cdot (1-p)^{i-1}$ ,  $i \in \{1, 2, \dots\}$ , где  $p \in (0, 1)$  – параметр распределения.

Характеристики распределения  $G(p)$  ( $x \in \{1, 2, \dots\}$ ):

- функция распределения:

$$F_{\xi}(x) = 1 - q^x, q = 1 - p;$$

- функция вероятности:

$$P_{\xi}(x) = p \cdot q^{x-1};$$

- среднее значение:  $\mu = 1/p$ ;
- дисперсия:  $\sigma^2 = q/p^2$ .

## Содержательная интерпретация

ДСВ  $\xi$  с законом распределения  $G(p)$  есть число испытаний Бернулли до первого успеха (включая первый успех), если вероятность успеха в каждом испытании равна  $p$ .

## Алгоритм моделирования

Алгоритм моделирования ДСВ  $\xi$  с законом распределения  $G(p)$  состоит из двух шагов:

1. Моделирование реализации  $a$  БСВ.
2. Принятие решения о том, что реализацией  $\xi$  является значение  $x$ , определяемое соотношением:

$$x = \lceil \ln a / \ln q \rceil,$$

здесь  $\lceil z \rceil$  означает округление числа  $z$  в большую сторону до ближайшего целого значения.

Коэффициент использования БСВ  $k = 1$ .

## Отрицательное биномиальное распределение

ДСВ  $\xi$  имеет отрицательное биномиальное распределение  $\bar{B}i(m, p)$ , если:  $\xi \in \{0, 1, \dots\}$ ,  $P\{\xi = i\} = C_{i+m-1}^{i+m-1} p^m (1-p)^i$ ,  $i \in \{0, 1, \dots\}$ . Параметры распределения:  $m$  – натуральное число,  $p \in (0, 1)$ .

Характеристики распределения  $\bar{B}i(m, p)$  ( $x \in \{0, 1, \dots\}$ ):

Описываемый тест используется для проверки гипотезы  $H_0$  о равномерности двумерного распределения векторов  $\{(a_{2(j-1)+1}, a_{2j})\}$  и представляет собой следующее решающее правило:

$$\text{принимается} \begin{cases} H_0, x^2 < \Delta, \\ H_1, x^2 \geq \Delta, \end{cases} \quad (11)$$

где в случае истинной гипотезы  $H_0$  и  $n \rightarrow \infty$  статистика

$$x^2 = \sum_{i=1}^k \frac{(m_i - mp_i)^2}{mp_i} \quad (12)$$

имеет  $x^2$  – распределение с  $k - 1$  степенями свободы, а порог  $\Delta$  определяется как квантиль этого распределения:  $\Delta = F^{-1}(1 - \varepsilon)$ , где  $\varepsilon$  – заданный уровень значимости.

В пакете СТАТМОД предполагается использование эквивалентной формы правила (11):

$$\text{принимается} \begin{cases} H_0, \varepsilon < P, \\ H_1, \varepsilon \geq P, \end{cases}$$

где  $P$  - значение вычисляется по формуле:  $P = 1 - F(x^2)$ , здесь  $F(.)$ - функция распределения статистики (12).

### 1.2.3. Графический анализ точности моделирования

Тесты проверки точности моделирования БСВ допускают графическую интерпретацию, удобную для быстрого визуального анализа качества смоделированной выборки  $A = \{a_1, \dots, a_n\}$  реализаций БСВ.

В пакете СТАТМОД реализованы следующие графики:

- диаграмма рассеяния (иллюстрирует зависимость между  $a_i$  и  $a_{i-1}$  ( $i = \overline{2, n}$ ); для проверки факта наличия зависимости используется тест «ковариация»);
- гистограмма и плотность распределения (позволяет осуществить сравнительный анализ теоретического и эмпирического распределения выборки  $A$ ; для проверки согласия распределения  $A$  с равномерным законом используются критерии согласия);
- график корреляционной функции с указанием доверительных границ (служит для графической поддержки теста «ковариация»);

- визуализация выборки (иллюстрирует зависимость  $a_i$  от  $t$  ( $t = \overline{1, n}$ ) и может использоваться для графической поддержки критериев серий).

## 1.2.4. Задания

1. Осуществить моделирование  $n=1000$  реализаций БСВ с помощью мультипликативного конгруэнтного метода (МКМ-датчика) при следующих значениях параметров датчика:

$$\beta = a_0^* = 2^m + 3, m = 4, 8, 12, 16, 20, 24, 28.$$

(т.е.  $\beta \in \{19, 259, 4099, 65539, 1048979, 16777219, 268435459\}$ )

Сравнить эти датчики по точности, используя тесты проверки точности моделирования. Найти значение  $m^*$ , при котором достигается максимальная точность моделирования.

2. Выполнить задание 1 при следующих значениях параметров датчика:

$$\beta = a_0^* = 5^{2m+1}, m = 0, 1, 2, 3, 4, 5, 6.$$

(т.е.  $\beta \in \{5, 125, 3125, 78125, 1953125, 48828125, 1220703125\}$ )

3. Для МКМ-датчика при  $\beta = 65539$  исследовать зависимость точности моделирования и величины периода  $T$  от «стартового» числа  $a_0^*$ . Положить  $a_0^* \in \{3, 19, 259, 4099, 65539, 1048979, 2147483647\}$

Сравнить по точности датчики, реализующие МКМ при  $\beta = 65539$  и  $\beta = 78125$  ( $a_0^* = \beta$ ).

Сравнить по точности МКМ-датчики при различных значениях  $\beta$  из заданного в пакете множества возможных значений. Положить  $a_0^* = \beta$ .

4. Датчики Макларена-Марсальи (ММ-датчики)  $M_1$  и  $M_2$  используют в качестве простейших датчики  $D_1$  и  $D_2$ , реализующие МКМ при следующих значениях параметров:

$$M_1 . \quad D_1 : \beta = a_0^* = 65539,$$

$$D_2 : \beta = 65539, a_0^* = 256959681;$$

$$M_2 . \quad D_1 : \beta = a_0^* = 1078318301,$$

$$D_2 : \beta = 1078318301, a_0^* = 860574881;$$

Сравнить по точности датчики  $M_1$  и  $M_2$ .

Сравнить по точности и быстродействию ММ-датчик  $M_1$  из задания 4 и МКМ-датчик с параметрами  $\beta = a_0^* = 65539$ .

5. Осуществить моделирование  $M = 100$  реализаций СВ  $\xi \in H(N, m, n)$ . Исследовать точность моделирования с помощью тестов и

графического анализа. Рассмотреть случаи:  $N = 100$ ,  $m = 5, 10, 25, 50$ ,  $n = 10, 20$ .

6. С помощью моделирования СВ с законами распределения  $H(N, m, n)$ ,  $Bi(n, p)$  ( $p = m/N$ ),  $\Pi(\lambda)$  ( $\lambda = n \cdot m/N$ ),  $N_1(\mu, 1)$ , ( $\mu = \lambda$ ) исследовать возможность и точность аппроксимации гипергеометрического распределения другими законами: биномиальным (14.1), Пуассона (14.2), нормальным (14.3):

- a)  $N = 1000$ ,  $m = 100, 50, 25, 10, 5$ ;
- b)  $N = 1000$ ,  $m = 100$ ,  $n = 10$ ;
- c)  $N = 1000$ ,  $m = 50$ ,  $n = 20$ ;
- d)  $N = 2000$ ,  $m = 100$ ,  $n = 200$ ;
- e)  $N = 3000$ ,  $m = 150$ ,  $n = 300$ ;
- f)  $N = 4000$ ,  $m = 200$ ,  $n = 300$ .

### Распределение Пуассона

7. Осуществить моделирование  $M = 100$  реализаций СВ  $\xi \sim \Pi(\lambda)$ . Исследовать точность моделирования с помощью тестов и графического анализа. Рассмотреть случаи:  $\lambda = 0.5, 1, 2, 4, 6, 10$ .
8. С помощью моделирования  $M = 100$  реализаций СВ  $\xi \sim \Pi(\lambda)$  и СВ  $\eta \sim N_1(\lambda, \lambda)$  исследовать возможность и точность аппроксимации распределения Пуассона нормальным. Рассмотреть случаи:  $\lambda = 6, 9, 12, 15, 18$ .
9. С помощью моделирования  $M = 100$  реализаций СВ  $\xi \sim \Pi(\lambda)$  и СВ  $\eta \sim \chi^2(2(1+x))$  проиллюстрировать соотношение между распределениями:

$$P\{\xi \leq x\} = P\{\eta > 2\lambda\}.$$

Положить:  $\lambda = 0.5, 1, 2$ ;  $x = 0, 1, 2$ .

10. Сравнить по точности и быстродействию методы моделирования СВ  $\xi \sim \Pi(\lambda)$ , реализованные в ППП СТАТМОД.

### Дискретное равномерное распределение

11. Для выборочного контроля качества штучной продукции требуется извлечь случайную выборку единиц продукции объема  $n$  из контролируемой партии. Партия продукции включает  $N$



контейнеров по  $L$  ящиков в каждом. Один контейнер содержит  $K$  единиц продукции ( $n < N \cdot L \leq K$ ). При формировании выборки необходимо обеспечить равновероятный выбор каждой единицы продукции из партии. Сформировать возможные планы контроля, включающие: номер контейнера, номер ящика и номера единиц продукции в ящике. Нумерация ящиков в контейнерах, а также единиц продукции в ящиках, - автономная. Положить:  $N = 5$ ,  $L = 20$ ,  $K = 100$ ,  $n = 50$ .

12. Осуществить моделирование 20 десятизначных номеров серий лотереи, используя алгоритм моделирования дискретного равномерного распределения. Исследовать точность моделирования.

## 1.3. Моделирование непрерывных случайных величин (НСВ)

### 1.3.1. Понятие НСВ. Универсальные методы моделирования НСВ.

#### Понятие НСВ.

Непрерывной случайной величиной (НСВ) называется случайная величина  $\xi \in \mathbf{R}^1$ , имеющая абсолютно-непрерывное распределение вероятности, описываемое функцией распределения

$$F_{\xi}(x) = P\{\xi < x\} = \int_{-\infty}^x p_{\xi}(x) dx$$

и плотностью распределения

$$p_{\xi}(x) = \frac{dF_{\xi}(x)}{dx}, x \in \mathbf{R}^1.$$

Основными числовыми характеристиками НСВ  $\xi$  являются:

- среднее значение  $\mu = E\{\xi\} = \int_{\mathbf{R}^1} x p_{\xi}(x) dx$ ;
- дисперсия  $\sigma^2 = E\{(\xi - \mu)^2\} = \int_{\mathbf{R}^1} (x - \mu)^2 p_{\xi}(x) dx$ .

На практике для описания НСВ используются модельные непрерывные законы распределения с функциональными характеристиками, заданными в параметрическом виде:

$$F_{\xi}(x) = F_{\xi}(x, \theta), p_{\xi}(x) = p_{\xi}(x, \theta),$$

где  $\theta \in \mathbf{R}^m (m \leq 1)$  - параметр распределения.

В пакете СТАТМОД реализованы алгоритмы моделирования НСВ для следующих модельных законов распределения.

1. равномерное распределение;
2. одномерное нормальное распределение;
3. лог - нормальное распределение;
4. экспоненциальное распределение;
5. распределение Вейбулла-Гнеденко;
6. распределение Лапласа;
7. гамма-распределение;
8. бета- распределение;
9. распределение Коши;
10. логистическое распределение;

11.  $\chi^2$  - распределение;
12. распределение Стюдента;
13. распределение Фишера.

Во многих практических задачах выборку наблюдений нельзя считать однородной, поскольку выборочные наблюдения соответствуют не одной, а нескольким моделям. Распределение такой выборки описывается смесью распределений. В связи с этим актуальной задачей является задача моделирования смеси распределений. Алгоритм моделирования смеси двух нормальных распределений реализован в ППП СТАТМОД.

Основными методами построения моделирующих алгоритмов для указанных законов распределения являются:

- метод обратной функции;
  - метод исключения;
  - метод функциональных преобразований,
- и другие методы, основанные на учете свойств распределений.

В приложениях часто возникает задача моделирования НСВ  $\xi$  в условиях априорной неопределенности, когда плотность  $p_{\xi}(x)$  неизвестна. В этих случаях может осуществляться:

- моделирование СВ с заданной гистограммой;
- моделирование СВ с заданным полигоном частот.

Гистограмма и полигон частот выступают как оценки плотности, построенные по имеющейся выборке экспериментальных данных.

Универсальными методами проверки точности моделирования НСВ являются критерии согласия ( $\chi^2$  - Пирсона, Колмогорова и др. ), а также критерии серий, реализованные в пакете.

Графические методы анализа точности моделирования НСВ включают:

- анализ гистограммы частот распределения;
- анализ эмпирической функции распределения.

Дадим характеристику перечисленным выше универсальным методам моделирования и проверки точности моделирования НСВ.

## **Метод обратной функции.**

Метод обратной функции является одним из универсальных методов моделирования НСВ  $\xi$  с заданной плотностью  $p_{\xi}(x)$  и функцией распределения  $F_{\xi}(x)$ . Приведем математическое обоснование метода и сформулируем моделирующий алгоритм.

Пусть  $F_{\xi}(x)$  - строго монотонная возрастающая функция. Найдем обратную функцию  $F_{\xi}^{-1}(x)$ , решая относительно  $x$  следующее уравнение:

$$F_{\xi}(x) = y.$$

Известно, что, если  $\alpha$  - БСВ, то СВ  $\xi$ , определяемая выражением:

$$\xi = F_{\xi}^{-1}(\alpha),$$

имеет заданную плотность  $p_{\xi}(x)$  (функцию распределения  $F_{\xi}(x)$ ).

Таким образом, имеет место следующий алгоритм моделирования:

Моделируется реализация БСВ  $\alpha$ .

Принимается решение о том, что реализацией СВ  $\xi$  является величина  $x$ , определяемая в соответствии с (14) по формуле:

$$x = F_{\xi}^{-1}(\alpha).$$

Коэффициент использования БСВ  $k = 1$ .

На этом методе основываются алгоритмы моделирования НСВ со следующими распределениями, реализованными в ППП СТАТМОД: равномерным, экспоненциальным, Лапласа, Вейбулса-Гнеденко, Коши, логистическое, гамма-распределение.

## Метод исключения

В тех случаях, когда плотность распределения  $p_{\xi}(x)$  моделируемой НСВ  $\xi$  имеет сложный аналитический ряд, нахождение функции распределения  $F_{\xi}(x)$ , а тем более обратной функции  $F_{\xi}^{-1}(y)$  затруднительно, что делает невозможным применение метода обратной функции для моделирования СВ  $\xi$  /

В этом случае может оказаться полезным другой универсальный метод моделирования, называемый методом исключения (или методом Дж.Неймана). Опишем моделирующий алгоритм, основанный на данном методе.

Обозначим:  $F_0 = \{(x, y) : 0 \leq y \leq p_{\xi}(x), x \in R^1\}$  - область, ограниченная кривой  $y = p_{\xi}(x)$  и осью абсцисс. Определим мажорирующую функцию  $y = g(x) : 0 \leq p_{\xi}(x) \leq g(x)$  и область  $G = \{(x, y) : 0 \leq y \leq g(x), x \in R\} \supset F_0$ .

Заметим, что мажорирующая функция должна иметь значительно более простой аналитический вид, чем  $p_{\xi}(x)$ . Область  $G$  при этом имеет также простой вид (треугольный, прямоугольный), позволяющий легко моделировать случайный вектор  $(\xi', \eta')$ , равномерно распределённый в области  $G$  (например, при помощи метода обратной функции).

Алгоритм моделирования, основанный на методе исключения, включает следующие этапы:

1. Подбор мажорирующей функции  $g(x)$ ;

2. Моделирование реализации  $(x_k, y_k) (k = 1, 2, \dots)$  случайного вектора  $(\xi', \eta')$  с равномерным распределением в области  $G$ ;

3. Принятие решения о том, что реализацией  $\xi$  является  $x_{k^*}$  при выполнении следующего условия:

$$k^* = \min\{k : (x_k, y_k) \in F_0, k = 1, 2, \dots\}.$$

Запись  $(x_k, y_k) \in F_0$  означает, что точка с координатами  $(x_k, y_k)$  принадлежит области  $F_0$ . Точки  $(x_k, y_k) \in G(k = \overline{1, k^* - 1})$ , не попавшие в  $F_0$ , исключаются из рассмотрения. Отсюда происходит название метода.

Для моделирования случайного вектора  $(\xi', \eta')$  с равномерным распределением в области  $G$  полагают:

$$p_{\xi'}(x) = \frac{g(x)}{\text{mes}G}, p_{\eta|\xi'}(y|x) = \frac{I_{[0, g(x)]}(y)}{g(x)},$$

где:  $\text{mes}G$  - мера Лебега (площадь) области  $G$ ;

$$I_A(y) = \begin{cases} 0, & \text{если } y \notin A \\ 1, & \text{если } y \in A \end{cases} \text{ - индикаторная функция.}$$

Моделирование СВ  $\xi'$  и  $\eta'$  (при условии, что  $\xi' = x$ ) осуществляется по методу обратной функции.

Средний коэффициент использования БСВ  $k = \frac{\ln(\text{mes}G)}{l(\text{mes}G - 1)}$ , где  $l$  -

количество БСВ (обычно  $l = 2$ ), используемых для получения одной реализации  $(x, y)$  случайного вектора  $(\xi', \eta')$ .

В ППП СТАТМОД данный метод используется для построения одного из алгоритмов моделирования гамма - распределения.

## 1.3.2. Тесты проверки точности моделирования НСВ

### Критерии согласия

Пусть  $X = \{x_1, \dots, x_n\}$  - случайная выборка реализаций СВ  $\xi$ , полученная с помощью некоторого алгоритма моделирования;  $F_{\xi}(x)$  - фактическое распределение выборочных наблюдений;  $F_0(x)$  - функция распределения требуемого модельного закона распределения. Одной из главных задач оценки точности моделирования является проверка выполнения условия:  $X$  есть выборка из распределения с функцией  $F_0(x)$ .

В терминах статистической проверки гипотез данная задача формулируется следующим образом: на заданном уровне значимости

$\varepsilon$  ( $0 < \varepsilon \leq 1$ ) по выборке  $X$  проверить гипотезу  $H_0$  при альтернативе  $H_1$  следующего вида:

$$H_0: F_{\xi}(x) = F_0(x), \quad H_1 = \overline{H_0}: F_{\xi}(x) \neq F_0(x). \quad (15)$$

Гипотеза  $H_0$  и соответствующий критерий проверки называется гипотезой и критерием согласия. Кратко опишем критерии согласия, реализованные в ППП СТАТМОД.

## $\chi^2$ - критерий согласия Пирсона

Данный критерий широко используется в задачах статистического анализа данных для проверки соответствия экспериментальных данных заданному модельному непрерывному или дискретному закону распределения, определяемому функцией распределения  $F_0(x) = F_0(x, \theta_0)$ . При этом истинные значения параметров  $\theta_0$  могут быть неизвестны. В задачах проверки точности моделирования значения  $\theta_0$  задаются при описании условий экспериментов, поэтому функцию  $F_0(x)$  можно считать полностью заданной.

Пусть как и при построении гистограммы (см. “Моделирование СВ с заданной гистограммой”) вычислены частоты  $\{v_k\}$  ( $k = \overline{1, K}$ ) попадания выборочных значений  $\{x_i\}$  ( $i = \overline{1, n}$ ) в  $K$  ячеек гистограммы. Гипотетические вероятности попадания значений  $\xi$  в ячейки гистограммы при истинной гипотезе  $H_0$  и полностью заданной функции  $F_0(x)$  равны:

$$p_k = P\{\xi \in [x_{k-1}, x_k)\} = F_0(x_k) - F_0(x_{k-1}), k = \overline{1, K},$$

где  $\{x_l\}$  ( $l = \overline{0, K}$ ) - границы ячеек гистограммы. Статистика критерия проверки гипотез (15) имеет вид:

$$\chi^2 = \sum_{k=1}^K \frac{(v_k - n \cdot p_k)^2}{n \cdot p_k} \geq 0 \quad (16)$$

и характеризует взвешенную сумму квадратов отклонений частот  $\{v_k\}$  от гипотетических значений  $\{n \cdot p_k\}$ . Чем больше  $\chi^2$ , тем “сильнее” выборка  $X$  не согласуется с  $H_0$ .

Статистика (16) имеет, в предположении, что гипотеза  $H_0$  верна,  $\chi^2$  - распределение с  $K - 1$  степенями свободы (см. соответствующий раздел).

$\chi^2$  - критерий Пирсона основан на (16) и имеет вид:

$$\text{принимается гипотеза } \begin{cases} H_0, & \text{если } \chi^2 < \Delta, \\ H_1, & \text{если } \chi^2 \geq \Delta, \end{cases} \quad (17)$$

где порог критерия  $\Delta$  находится из ограничения на ошибку первого рода:  $P\{H_1 | H_0\} = \varepsilon$  и имеет вид:  $\Delta = G^{-1}(1 - \varepsilon)$ , здесь  $G(\square)$  - функция распределения статистики (16).

В ППП СТАТМОД предполагается использование эквивалентной формы критерия (17):

$$\text{принимается гипотеза } \begin{cases} H_0, & \text{если } \varepsilon < P, \\ H_1, & \text{если } \varepsilon \geq P, \end{cases}$$

(18)

где  $P = 1 - G(\chi^2)$  - вычисляемая по выборке  $X$  критическая вероятность или

“ $P$ -значение”,  $\varepsilon$  - задаваемый пользователем уровень значимости (обычно  $\varepsilon = 0.05$ ).

Примечание. Если  $\xi \in \{c_1, c_2, \dots, c_k\}$  - дискретная СВ с функцией распределения  $F_0(x)$ ,  $X$  - выборка реализаций СВ  $\xi$ , тогда  $\{v_k\}, \{p_k\}$  - соответственно частоты и гипотетические вероятности значений  $\{c_k\}$ .

## Критерий согласия Колмогорова.

Данный критерий позволяет осуществить проверку гипотез (15) в условиях, когда функция распределения  $F_0(x)$  модельного закона известна полностью, то есть не зависит от неизвестных параметров. Он основан на анализе мер уклонения эмпирической и модельной функций распределения.

Эмпирическая функция распределения по случайной выборке  $X = \{x_1, x_2, \dots, x_n\}$  реализаций СВ  $\xi$  определяется по формуле:

$$\hat{F}_\xi(x) = \frac{1}{n} \sum_{i=1}^n I_{[-\infty, x]}(x_i), \quad x \in R^1.$$

Введём статистику

$$Dn = \sup_{x \in R^1} |\hat{F}_\xi(x) - F_0(x)| \in [0, 1],$$

называемую расстоянием Колмогорова между  $\hat{F}_\xi(\square)$  и  $F_0(\square)$ .

Известно, что гипотеза  $H_0$  верна и  $n \rightarrow \infty$  (практически  $n > 20$ ), то статистика  $\sqrt{n}Dn$  имеет распределение Колмогорова с функцией распределения  $K(y)$  вида:

$$K(y) = 1 - 2 \sum_{j=1}^{\infty} (-1)^{j-1} \exp(-2j^2 y^2), \quad y \geq 0.$$

Критерий согласия Колмогорова представляет собой следующее решающее правило:

$$\text{принимается гипотеза } \begin{cases} H_0, & \text{если } \sqrt{n}Dn < \Delta, \\ H_1, & \text{если } \sqrt{n}Dn \geq \Delta, \end{cases}$$

Порог  $\Delta = K^{-1}(1 - \varepsilon)$  - квантиль уровня  $(1 - \varepsilon)$  распределения Колмогорова,  $\varepsilon$  - задаваемый пользователем уровень значимости.

В ППП СТАТМОД предполагается использование эквивалентной формы данного критерия вида (18), где  $P = 1 - K(\sqrt{n}D)$ .

## Критерий серий

Критерий серий предназначен для проверки гипотезы о случайности выборки  $\{x_1, x_2, \dots, x_n\}$ . Критерий основан на исследовании знаковой последовательности разностей:

$$x_i - x_{med} \quad (i = \overline{1, n}), \quad (19)$$

где  $x_{med}$  - медиана выборки  $\{x_i\}$ . Знаковая последовательность состоит из знаков "+", "-", соответствующих разностям (19) и характеризуется:

1. общим числом серий  $\gamma(K)$ ,
2. протяжённостью самой длинной среди  $T(K)$ , где  $K$  ( $K \leq n$ ) - число элементов знаковой последовательности.

Под "серией" понимается последовательность подряд идущих одинаковых знаков.

Очевидно, если  $\{x_i\}$  - случайная выборка, то знаковая последовательность не должна содержать слишком длинных серий, а общее число серий не должно быть слишком малым.

В зависимости от величины  $K$  целесообразно использовать одно из следующих правил принятия решений.

### Правило в случае $K < 20$ .

Если хотя бы одно из неравенств:

$$\begin{aligned} \gamma(K) &> [0.5(K+1) - 1.96\sqrt{K-1}]; \\ T(K) &< [3.3 \log_{10}(K+1)] \end{aligned}$$

(где  $[a]$  – целая часть числа  $a$ ) оказывается нарушенным, то гипотеза о случайности выборки  $\{x_i\}$  отвергается с вероятностью ошибки  $\alpha: 0.05 < \alpha < 0.0975$ .

**Правило в случае  $K > 20$**  основано на использовании статистики:

$$\gamma^*(K) = \frac{\gamma - m - 0.5}{((m-1)m/(2m-1))^{1/2}}; \quad (20)$$

где  $m$  – число наблюдений, меньших (больших)  $x_{med}$ . Известно, что статистика (20) распределена асимптотически (при  $K \rightarrow \infty$ ) по стандартному нормальному закону. Отсюда правило принятия решения следующее: гипотеза о случайности выборки  $\{x_i\}$  принимается, если  $\gamma^* < \Phi^{-1}(1 - \alpha/2)$  и отвергается в противном случае ( $\alpha$  - заданный уровень значимости).

## Критерий "нисходящих" и "восходящих" серий

Данный критерий позволяет обнаружить (с увеличением порядкового номера наблюдений) смещение среднего значения наблюдения как монотонного, так и более общего, например,



циклического характера. Так же, как и в случае критерия серий, исследуется последовательность знаков разностей. Однако используется другая последовательность разностей:  $x_{i+1} - x_i$  ( $x = \overline{1, n}$ ), причём, если два или несколько следующих друг за другом наблюдения равны между собой, то принимается во внимание только одно из них.

Серия из плюсов называется “восходящей”, а серия из минусов - “нисходящей”, поскольку имеют место при возрастании и убывании наблюдений соответственно.

Критерий основывается на тех же принципах и статистиках  $\gamma(K), T(K)$ , что и критерий серий.

Используется следующее решающее правило: если хотя бы одно из неравенств

$$\gamma(K) > [(2K-1)/3 - 1.96\sqrt{(16K-29)/90}],$$

$$T(K) < T_0(K) \quad (T_0 \text{ выбирается из таблицы 1})$$

оказывается нарушенным, то гипотеза о случайности выборки  $\{x_i\}$  отвергается с вероятностью ошибки  $\alpha : 0.05 < \alpha < 0.0975$ .

**Таблица 1: Зависимость  $T_0$  от  $K$**

$K$	$K \leq 26$	$16 < K \leq 153$	$153 < K \leq 1170$
$T_0$	5	6	7

### 1.3.3. Алгоритмы моделирования для основных непрерывных распределений

#### Равномерное распределение

НСВ  $\xi$  имеет равномерно е распределение на интервале  $[a, b)$ , обозначаемое  $R(a, b)$ , если функция и плотность распределения  $\xi$  определяются соотношениями:

$$F_{\xi}(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & x \in [a, b), \\ 1, & x > b \end{cases}, \quad p_{\xi}(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b) \\ 0, & \text{иначе} \end{cases} \quad (21)$$

Для произвольных значений параметров распределения  $a, b$  распределение  $R(a, b)$  обобщает распределение  $R(0, 1)$  БСВ  $\alpha$ .

Среднее значение и дисперсия  $\xi \in R(a, b)$  равны:

$$\mu = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}.$$

#### Алгоритм моделирования

Алгоритм моделирования СВ  $\xi \in R(a,b)$  основан на методе обратной функции. Обратная функция для  $F_\xi(x)$  из (21) находится при решении уравнения  $F_\xi(x) = y$  относительно  $x$ :

$$x = F_\xi^{-1}(y) = (b-a)y + a, \quad y \in [0,1].$$

Далее в соответствии с указанным методом алгоритм моделирования реализации СВ  $\xi$  включает два шага:

- моделирование реализации  $\eta$  БСВ
- принятие решения о том, что реализацией  $\xi$  является величина  $x$ :

$$x = (b-a) \cdot \eta + a.$$

Коэффициент использования БСВ  $k = 1$ .

## Одномерное нормальное распределение

НСВ  $\xi \in R^1$  с плотностью распределения

$$p_\xi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

имеет одномерное нормальное (гауссово) распределение с параметрами: средним значением  $\mu \in R^1$  и дисперсией  $\sigma^2 > 0$  (обозначается  $N_1(\mu, \sigma^2)$ ).

Распределение  $N_1(0,1)$  называется стандартным нормальным распределением, а НСВ  $\eta \sim N_1(0,1)$ -стандартной нормальной (гаусовской) величиной. Функция распределения  $N_1(0,1)$  обозначается  $\Phi(x)$  и имеет вид:

$$\Phi(x) = F_\eta(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt \quad (22)$$

и называется функцией Лапласа.

Случайные величины  $\xi$  и  $\eta$  связаны соотношением:

$$\xi = \eta + \sigma \cdot \eta, \quad (23)$$

где  $\sigma$ -среднее квадратическое (стандартное) отклонение. Таким образом, задача моделирования  $\xi \sim N_1(\mu, \sigma^2)$  сводится к моделированию стандартной гаусовской СВ  $\eta$  и применению формулы (23). Опишем некоторые алгоритмы моделирования  $\eta \sim N_1(0,1)$ .

### Алгоритмы моделирования

Первый алгоритм реализуем методом суммирования, основанном на центральной предельной теореме: если  $a_1, a_2, \dots, a_N$ -независимые БСВ, то при  $N \rightarrow \infty$  случайная величина

$$\zeta = \sqrt{\frac{12}{N}} \left( \sum_{i=1}^N a_i - \frac{N}{2} \right)$$

распределена асимптотически нормально, так что  $F_{\zeta}(x) \rightarrow \Phi(x)$ ,  $x \in R^1$ .

На практике приемлемая точность аппроксимации стандартной гауссовской СВ достигается при  $N=12$ . Таким образом, алгоритм моделирования  $\eta \sim N_1(0,1)$  состоит из следующих шагов:

(а) моделирование  $N=12$  реализацией  $a_1, a_2, \dots, a_{12}$  БСВ.

(б) принятие решения о том, что реализацией СВ  $\eta$  является величина  $x$ , равная :

$$x = \sum_{i=1}^{12} a_i - 6.$$

Коэффициент использования БСВ  $k = \frac{1}{N}$ .

Второй алгоритм основан на методе функционального преобразования БСВ. Известно, что если  $a_1, a_2$  -независимые БСВ, то случайные величины

$$\eta_1 = \sqrt{-2 \ln a_1} \cdot \cos(2\pi a_2), \quad \eta_2 = \sqrt{-2 \ln a_1} \cdot \sin(2\pi a_2) \quad (24)$$

Являются независимыми стандартными гауссовскими. Таким образом, алгоритм моделирования  $\eta \sim N_1(0,1)$  на основе данного метода позволяет получить из двух реализаций  $a_1, a_2$  БСВ две независимые реализации СВ  $\eta$  с помощью преобразований (24).

### Лог - нормальное распределение

НСВ  $\xi \in [0, +\infty)$  с плотностью распределения

$$p_{\xi}(x) = \frac{1}{x \cdot \sqrt{2\pi\sigma}} \left\{ -\frac{1}{2} \left( \frac{\ln(x/\mu)}{\sigma} \right)^2 \right\}, \quad x \in [0, +\infty).$$

имеет логарифмически-нормальное распределение (лог-нормальное распределение)  $LN(m, \sigma)$  с параметрами формы  $\sigma > 0$  (стандартное отклонение СВ  $\ln \xi$ ).

Среднее значение и дисперсия  $\xi \sim LN(m, \sigma)$  определяются формулами:

$$E\{\xi\} = m\sqrt{w}, \quad D\{\xi\} = m^2 w(w-1), \quad w = \exp\{\sigma^2\}.$$

Очевидно, СВ  $\xi \geq 0$  имеет распределение  $LN(m, \sigma)$ , если СВ  $\eta = \ln \xi$  распределена по нормальному закону  $N_1(\mu, \sigma^2)$ , причем  $\mu = \ln m$ . И наоборот: если  $\eta \sim N_1(\mu, \sigma^2)$ , то СВ  $\xi = \exp\{\eta\} \sim LN(m, \sigma)$ . Эта связь распределений лежит в основе моделирующего алгоритма ( $m = \exp\{\mu\}$ ).

## Алгоритм моделирования

Алгоритм моделирования  $\xi \sim LN(m, \sigma)$  основан на методе функциональных преобразований и состоит из следующих шагов:

1. Моделирование  $z$  реализации стандартной гауссовской СВ.
2. Получение  $y$  реализацией СВ  $\eta \sim N_1(\mu, \sigma^2)$  (где  $\mu = \ln m$ ) по формуле:  
$$y = \mu + \sigma \cdot z.$$

3. вычисление реализации  $x$  СВ  $\xi \sim LN(m, \sigma)$  по формуле  
$$x = m \cdot \exp\{\sigma \cdot z\}.$$

## Экспоненциальное распределение

НСВ  $\xi \in [0, +\infty)$  с функцией и плотностью распределения, определяемые соотношениями:

$$F_{\xi}(x) = 1 - e^{-\lambda x}, \quad p_{\xi}(x) = \lambda e^{-\lambda x}, \quad x \geq 0, \quad (25)$$

имеет экспоненциальное(потенциальное) распределение  $E(\lambda)$ , где  $\lambda > 0$ -параметр распределения ( $\lambda = \frac{1}{E\{\xi\}}$ ).

Среднее значение и дисперсия СВ  $\xi \sim E(\lambda)$  равны:  $\mu = \frac{1}{\lambda}, \sigma^2 = \frac{1}{\lambda^2}.$

Связь с другими распределениями

Экспоненциальное распределение  $E(\lambda)$  можно рассматривать как частный случай распределений:

- гамма - распределения  $\gamma(b, \nu)$  при  $\nu = 1, \lambda = \frac{1}{b}$ ;
- Вейбулла-Гнеденко  $WG(\lambda, c)$  при  $c=1$ .

## Алгоритм моделирования

Алгоритм моделирования СВ  $\xi \sim E(\lambda)$  основа на методе обратной функции. Обратная функция для  $F_{\xi}(x)$ , определяемой (25), находится при решении уравнения  $F_{\xi}(x) = y$  относительно  $x$ :

$$x = F_{\xi}^{-1}(y) = -\ln(1-y)/\lambda. \quad (26)$$

Далее в соответствии с методом обратной функции алгоритм моделирования СВ  $\xi$  состоит из двух шагов:

1. Моделирование реализации  $a$  БСВ  $a$ .
2. Вычисление в соответствии с (26) реализации  $x$  СВ  $\xi$ :

$$x = -\frac{1}{\lambda} \ln a,$$

где учтено, что  $a$  и  $a-1$  одинаково распределены.  
Коэффициент использования БСВ  $k=1$ .

## Распределение Лапласа

НСВ  $\xi \in R^1$  с плотностью распределения

$$p_{\xi}(x) = \frac{\lambda}{2} e^{-\lambda|x|}, \quad x \in R^1, (27)$$

имеет распределение Лапласа  $L(\lambda)$ , где  $\lambda > 0$  параметр распределения. Распределение  $L(\lambda)$  иногда называется двусторонним экспоненциальным распределением, поскольку график функции (27) можно рассматривать как результат “склеивания” графика плотности распределения  $E(\lambda)$  со своим зеркальным относительно вертикальной оси отражением (с учетом необходимой нормировки). Согласно (27) функция распределения  $L(\lambda)$  имеет вид:

$$F_{\xi}(x) = \begin{cases} \frac{1}{2} e^{\lambda x}, & \text{если } x < 0 \\ 1 - \frac{1}{2} e^{-\lambda x}, & \text{если } x \geq 0 \end{cases} \quad (28)$$

Среднее значение и дисперсия СВ  $\xi \sim E(\lambda)$ :  $\mu = 0, \sigma^2 = 2/\lambda^2$ .

## Алгоритм моделирования

Алгоритм моделирования  $\xi \sim L(\lambda)$  основан на методе обратной функции. Обратная для  $F_{\xi}(x)$  функция согласно (28) имеет вид:

$$x = F_{\xi}^{-1}(y) = \frac{1}{\lambda} \ln(2y) < 0, \quad y \in [0, 0.5) \quad (29)$$

$$x = F_{\xi}^{-1}(y) = -\frac{1}{\lambda} \ln(2(1-y)) \geq 0, \quad y \in [0.5, 1) \quad (30)$$

Для моделирования реализации  $x$  СВ  $\xi \sim L(\lambda)$  выполняются следующие действия:

1. Моделируется реализация  $y$  БСВ.
2. Принимается решение о том, что реализацией СВ  $\xi$  является величина  $x$ , вычисляемая по формуле (29), если  $y < 0.5$ ; по формуле (30), если  $y \geq 0.5$ .

Коэффициент использования БСВ  $k=1$ .

## Распределение Вейбулла-Гнеденко.

НСВ  $\xi \in [0, +\infty)$  с плотностью распределения

$$p_{\xi}(x) = c \cdot \lambda x^{c-1} \exp\{-\lambda x^c\}, \quad x \geq 0, (31)$$

имеет распределение Вейбулла-Гнеденко  $WG(\lambda, c)$  имеет вид:

$$F_{\xi}(x) = 1 - \exp\{-\lambda x^c\}, \quad x \geq 0. (32)$$

Среднее значение и дисперсия равны:

$$\mu = \lambda^{-1/c} \cdot \Gamma\left(1 + \frac{1}{c}\right),$$

$$\sigma^2 = \lambda^{-2/c} \left( \Gamma\left(1 + \frac{2}{c}\right) - \Gamma^2\left(1 + \frac{1}{c}\right) \right),$$

здесь  $\Gamma(x)$ -гамма-функция Эйлера, то есть

$$\Gamma(x) = \int_0^{+\infty} x^{x-1} e^{-x} dx; \Gamma(k) = (k-1)!, \quad k = 1, 2, \dots$$

## Связь с другими распределениями

Частными случаями распределения  $WG(\lambda, c)$  с плотностью (31) являются:

1. экспоненциальное распределение  $E(\lambda)$  при  $c=1$ ;
2. распределение Релея, имеющее плотность

$$P_\eta(x) = \frac{x}{\gamma^2} \exp\{-x^2/2\gamma^2\}, \quad x \geq 0,$$

при  $c=2$  и  $\lambda = 1/2\gamma^2$ .

## Алгоритм моделирования

Алгоритм моделирования СВ  $\xi \sim WG(\lambda, c)$  основан на методе обратной функции с состоит из следующих шагов:

1. Моделирование реализации а БСВ.
2. Принятие решения о том, что реализацией СВ  $\xi$  является величина  $x$ , вычисляемая с учетом (32) по формуле:

$$x = F_\xi^{-1}(x) = \left( -\frac{1}{\lambda} \ln a \right)^{1/c}.$$

Коэффициент использования БСВ  $k=1$ .

## Гамма-распределение.

НСВ  $\xi \in [0, +\infty)$  с плотностью распределения

$$p_\xi(x) = \frac{(x/b)^{\nu-1} \exp\{-x/b\}}{b \cdot \Gamma(\nu)}, \quad x \geq 0,$$

имеет гамма-распределение  $\gamma(b, \nu)$  с параметрами:  $\nu > 0$  - параметр формы;  $b > 0$  – параметр масштаба. Здесь  $\Gamma(\nu)$  -гамма-функция Эйлера:

$$\Gamma(\nu) = \int_0^{+\infty} t^{\nu-1} e^{-t} dt.$$

Среднее значение и дисперсия  $\xi \sim \gamma(b, \nu)$  равны:

$$\mu = \nu \cdot b, \quad \sigma^2 = \nu \cdot b^2$$

## Связь с другими распределениями.

При  $\nu = 1$  гамма-распределение совпадает с экспоненциальным:  
 $\gamma(b,1) = E(\lambda), \quad \lambda = 1/b$ .

Для произвольного целого  $\nu > 1$  гамма-распределение  $\gamma(b,\nu)$  называется распределением Эрланга порядка  $\nu$  с параметром  $\lambda = 1/b$ .

Если  $\nu \geq 1$ -целое число,  $\eta_1, \dots, \eta_\nu$ -независимые случайные величины, распределенные по стандартному экспоненциальному закону  $E(\lambda)$  ( $\lambda = 1$ ), то СВ  $\xi$  вида:

$$\xi = b \cdot \sum_{i=1}^{\nu} \eta_i \quad b > 0 \quad (33)$$

имеет распределение  $\gamma(b,\nu)$ .

В соответствии с методом обратной функции:

$$\eta_i = -\ln \alpha_i \quad (i = \overline{1, \nu}), \quad (33)$$

$\alpha_1, \dots, \alpha_\nu$  - независимые БСВ. С учётом этого из (33) следует:

$$\xi = -\ln\left(\prod_{i=1}^{\nu} \alpha_i\right) \cdot b \quad (34)$$

Если  $\nu = N + 0.5$  ( $n = 0, 1, 2, \dots$ ),  $\alpha_1, \dots, \alpha_\nu$  - независимые БСВ,  $\varsigma \in N_1(0,1)$ , то СВ вида:

$$\xi = \left(-\ln\left(\prod_{j=1}^N \alpha_j\right) + \frac{\varsigma^2}{2}\right) \cdot b \quad (35)$$

имеет распределение  $\gamma(b,\nu)$ .

## Алгоритмы моделирования:

1. Для целых значений  $\nu \geq 1$  алгоритм моделирования  $\xi \in \gamma(b,\nu)$  описывается формулой (34). Коэффициент использования БСВ  $k = \frac{1}{\nu}$ .
2. Для  $\nu = N + 0.5$  ( $n = 0, 1, 2, \dots$ ) алгоритм моделирования  $\xi \in \gamma(b,\nu)$  описывается формулой (35). Если для моделирования  $\frac{\varsigma^2}{2}$  используется формула (см. «Одномерное нормальное распределение»):

$$\frac{\varsigma^2}{2} = -\ln \alpha_{N+1} \cdot \cos^2(2\pi \alpha_{N+2}),$$

где  $\alpha_{N+1}, \alpha_{N+2}$  - независимые БСВ, то коэффициент использования БСВ  $k = 1/(N+2)$ .

3. Для произвольных значений  $\nu > 0$  моделирование СВ  $\xi \in \gamma(b, \nu)$  может осуществляться с применением алгоритма, основанного на методах исключения и обратной функции.

Справедливо представление (для простоты полагаем  $b=1$ ):

$$\xi = -\ln\left(\prod_{i=1}^N \alpha_i\right) + \xi_*,$$

где:  $\nu_* = \nu - N$  ( $N = [\nu]$  – целая часть числа  $\nu, 0 \leq \nu_* < 1$ ), СВ  $\xi_* \in \gamma(1, \nu_*)$ . Для моделирования  $\xi_*$  используется метод исключения с мажорирующей функцией  $g(x)$ :

$$p_{\xi_*}(x) \leq g(x) = \begin{cases} x^{\nu_*-1}, & \text{если } 0 \leq x < 1, \\ e^{-x}, & \text{если } x \geq 1. \end{cases}$$

Пусть:  $G$  и  $F_0$  ( $F_0 \subset G$ ) – области, ограниченные осью абсцисс и кривыми  $y = g(x)$  и  $y = p_{\xi_*}(x)$  соответственно;  $(\xi'_*, \eta'_*)$  – случайный вектор, имеющий равномерное распределение в области  $G$ , а точка с координатами  $(x, z)$  принадлежит области  $F_0$ . Тогда в соответствии с методом исключения реализацией СВ  $\xi_*$  с плотностью  $g(x)/\text{mes } G$  является величина  $x$ . Для моделирования  $\xi_*$  удобно использовать метод обратной функции:

$$x = F_{\xi_*}^{-1} = \begin{cases} (1 + \xi_* a_1 / e)^{1/\nu_*}, & 0 < a_1 < \frac{1}{1 + \nu_*/e}; \\ -\ln((1 - a_1)\nu_*^{-1} + e^{-1}), & \text{иначе} \end{cases}$$

$z = g(x) \cdot a_2$ ; здесь  $a_1, a_2$  – независимые реализации БСВ.

## Бета-распределение.

НСВ  $\xi \in [0, 1]$  с плотностью распределения

$$p_{\xi}(x) = \frac{\Gamma(\nu + \omega) x^{\nu-1} (1-x)^{\omega-1}}{\Gamma(\nu) \cdot \Gamma(\omega)} I_{[0,1]}(x) \quad (36)$$

имеет бета-распределение  $\beta(\nu, \omega)$  с параметрами  $\nu > 0, \omega > 0$  (где  $\Gamma(z)$  – гамма-функция Эйлера).

Среднее значение и дисперсия  $\xi \in \beta(\nu, \omega)$  равны:

$$\mu = \frac{\nu}{\nu + \omega}, \quad \sigma^2 = \frac{\nu\omega}{(\nu + \omega)^2(\nu + \omega + 1)}.$$

## Связь с другими распределениями

1. Если  $\nu = \omega = 1$ , то СВ  $\xi$  с плотностью распределения (36) имеет распределение  $R(0, 1)$ .
2. Если  $\nu > 0, \omega > 0$  – целые числа,  $\eta$  и  $\varsigma$  – случайные величины, имеющие гамма-распределения  $\gamma(\nu, 1)$  и  $\gamma(\omega, 1)$  соответственно,  $\xi \in \beta(\nu, \omega)$ , тогда:



$$\xi = \frac{\eta}{\eta + \varsigma} . \quad (37)$$

### Алгоритмы моделирования:

1. Для целых значений  $\nu > 0, \omega > 0$  алгоритм моделирования  $\xi \square \beta(\nu, \omega)$  определяется формулой (37) (алгоритмы моделирования  $\eta, \varsigma$  описаны в разделе «Гамма-распределение»). Коэффициент использования БСВ  $k = \frac{1}{\nu + \omega}$ .

2. Другой алгоритм для целочисленных значений параметров  $\nu, \omega$  основан на методе функционального преобразования и описывается формулой:

$$\xi_* = \alpha_{(\nu)},$$

где  $\alpha_{(\nu)}$  -  $\nu$ -я порядковая статистика, соответствующая последовательности независимых БСВ  $\alpha_1, \dots, \alpha_{\nu+\omega-1} : \alpha_1 < \alpha_2 < \dots < \alpha_{\nu+\omega-1}$ .

Коэффициент использования БСВ  $k = \frac{1}{\nu + \omega - 1}$ .

3. Для нецелых параметров  $\nu, \omega$  используется алгоритм (метод Йонка), составленный из следующих шагов:
4. Моделирование пар независимых реализаций БСВ:  $\eta_k, \varsigma_k$  ( $k = 1, 2, \dots$ ) и вычисление величин  $S_k = \eta_k^{1/\nu} + \varsigma_k^{1/\nu}$ .
5. Принятие решения о том, что реализацией СВ  $\xi \square \beta(\nu, \omega)$  является величина вида:

$$x = \eta_{k^*}^{1/\nu} / S_{k^*},$$

где  $k^* = \min\{k : S_k \leq 1, k = 1, 2, \dots\}$ .

### Распределение Коши.

НСВ  $\xi \in R^1$  с плотностью распределения

$$p_\xi(x) = \frac{1}{\pi c} \cdot \frac{1}{((x-m)/c)^2}, \quad x \in R^1, \quad (38)$$

имеет распределение Коши  $C(m, c)$  с параметрами:  $c > 0$  - параметр масштаба;  $m \in R^1$  - параметр положения (мода, медиана).

Функция распределения СВ  $\xi \square C(m, c)$  имеет вид:

$$F_\xi(x) = \frac{1}{2} + \frac{1}{\pi} \cdot \arctg \frac{x-m}{c}, \quad x \in R^1, \quad (39)$$

Известно, что если  $\eta_1, \eta_2$  - независимые стандартные гаусовские величины, то СВ  $\xi$  вида

$$\xi = \eta_1 / \eta_2$$

имеет распределение Коши  $C(0, 1)$ .

## Алгоритмы моделирования

Первый алгоритм основывается на формуле (39) и состоит из двух шагов:

1. моделирование независимых реализаций  $y_1, y_2$  СВ  $\eta \in N_1(0,1)$ ;
2. принятие решения о том, что реализацией СВ  $\xi \in C(m,c)$  является величина  $x$ :

$$x = F_{\xi}^{-1}(a) = m + c \cdot \operatorname{tg}(\pi(a - 0.5)).$$

Коэффициент использования БСВ  $k = 1$ .

## Логистическое распределение

НСВ  $\xi \in R^1$  с плотностью распределения

$$p_{\xi}(x) = \frac{\exp\{(x - \mu)/k\}}{k(1 + \exp\{(x - \mu)/k\})^2}, \quad x \in R^1,$$

имеет логистическое распределение  $LG(\mu, k)$  с параметрами  $\mu \in R^1$  - среднее значение,  $k = \sqrt{3}\sigma/\pi > 0$  (где  $\sigma$  - стандартное отклонение СВ  $\xi$ ).

Функция распределения закона  $LG(\mu, k)$  имеет вид:

$$F_{\xi}(x) = (1 + \exp\{-(x - \mu)/k\})^{-1}. \quad (40)$$

## Алгоритм моделирования

Алгоритм моделирования  $\xi \in LG(\mu, k)$  основан на методе обратной функции. С учётом (40) обратная для  $F_{\xi}(x)$  имеет вид:

$$x = F_{\xi}^{-1}(y) = \mu + k \cdot \ln\left(\frac{y}{1-y}\right). \quad (41)$$

Для моделирования реализации  $x$  СВ  $\xi \in LG(\mu, k)$  выполняются следующие действия.

1. Моделируется реализация  $y$  БСВ.
2. Вычисляется значение  $x$  по формуле (41).

Коэффициент использования БСВ  $k = 1$ .

## Хи-квадрат распределение

НСВ  $\xi \in [0, +\infty)$  с плотностью распределения

$$p_{\xi}(x) = \frac{x^{(m-2)/2} e^{-x/2}}{2^{m/2} \Gamma(m/2)}, \quad x \geq 0, \quad (42)$$

имеет хи-квадрат распределение  $\chi^2(m)$  с  $m$  степенями свободы ( $m > 0$  - натуральное число, параметр распределения). Здесь  $\Gamma(z)$  - гамма-функция Эйлера (см. «Распределение Вейбулла-Гнеленко»).

Среднее значение и дисперсия  $\xi \in \chi^2(m)$  равны:  $\mu = m, \sigma^2 = 2m$ .

Известно, что, если  $\eta_1, \dots, \eta_m \in N_1(0,1)$  - независимые стандартные гаусовские СВ, то СВ

$$\xi = \sum_{i=1}^m \eta_i^2 \quad (43)$$

имеют плотность распределения (42).

## Алгоритмы моделирования

В основе первого алгоритма моделирования СВ  $\xi \in \chi^2(m)$  лежит свойство (43) : в качестве реализации СВ  $\xi$  принимается величина  $x$ , вычисленная по независимым реализациям  $z_1, \dots, z_m$  СВ  $\eta \in N_1(0,1)$  по формуле:

$$x = \sum_{i=1}^m z_i^2.$$

Коэффициент использования БСВ  $k = \frac{1}{mN_\eta}$ , где  $N_\eta$  - число реализаций БСВ, необходимых для моделирования одной реализации СВ  $\eta \in N_1(0,1)$ .

Пусть  $a_1, \dots, a_m$  - независимые реализации БСВ,  $z$  - независимая от  $\{a_i\}$  реализация СВ  $\eta \in N_1(0,1)$ . Второй алгоритм моделирования СВ  $\xi \in \chi^2(m)$  предполагает, что в качестве реализации СВ  $\xi$  принимается величина  $x$ , вычисляемая по формулам:

$$x = -\frac{1}{2} \ln \left( \prod_{i=2}^{m/2} a_i \right), \quad m \text{ чётно}, \quad (44)$$

$$x = -\frac{1}{2} \ln \left( \prod_{i=2}^{(m-1)/2} a_i \right) + z^2, \quad m \text{ нечётно}, \quad (45)$$

Коэффициент использования БСВ для случаев (44), (45) соответственно равен:  $k = (m/2)^{-1}$ ,  $k = ((m-1)/2 + N_\eta)^{-1}$ .

## Распределение Стьюдента

НСВ  $\xi \in R^1$  с плотностью распределения

$$p_\xi(x) = \frac{\Gamma\left(\frac{m+1}{2}\right)}{\sqrt{m\pi} \Gamma\left(\frac{m}{2}\right) \left(1 + \frac{x^2}{m}\right)^{(m+1)/2}}, x \in R^1$$

Имеет распределение Стьюдента (t-распределение)  $t(m)$  с  $m$  степенями свободы ( $m > 0$  - натуральное число, параметр распределения).

Среднее значение и дисперсия  $\xi \sim t(m)$  равны:  
 $\mu = 0, \sigma^2 = m(m-2), m > 2$ .

## Связь с другими распределениями

СВ  $\xi \sim t(m)$  связана с независимыми случайными величинами  $\xi_1 \sim \chi^2(1), \xi_m \sim \chi^2(m), \eta \sim N_1(0,1)$  следующими соотношениями:

1.  $\xi = \frac{\xi_1}{\xi_m/m} \sim F(1, m);$
2.  $\xi = \frac{\eta^2}{\xi_m/m};$
3.  $\xi = \frac{\eta}{(\xi_m/m)^{1/2}}.$

## Алгоритм моделирования

- определяется формулой (2).

## Распределение Фишера

НСВ  $\xi \in [0, +\infty)$  с плотностью распределения

$$p_\xi(x) = \frac{\Gamma\left(\frac{l+m}{2}\right) x^{\frac{l}{2}-1} (l/m)^{l/2}}{\Gamma\left(\frac{1}{2}\right) \Gamma\left(\frac{m}{2}\right) \left(1 + \frac{1}{m}x\right)^{(l+m)/2}}, x \geq 0$$

имеет распределение Фишера (F-распределение)  $F(1, m)$  с 1 и m числом степеней свободы. Здесь: l, m - натуральные числа, параметры распределения;  $\Gamma(z)$  - гамма-функция Эйлера (см. раздел «Распределение Вейбулла-Гнеденко»).

Среднее значение и дисперсия  $\xi \sim F(1, m)$  равны:

$$\mu = \frac{m}{m-2}, m > 2, \sigma^2 = 2m^2(l+m-2)/l(m-2)^2(m-4), m > 4.$$

## Связь с другими распределениями

Пусть  $\xi_1 \sim \chi^2(1), \xi_m \sim \chi^2(m)$  - независимые случайные величины, а СВ  $\xi$  определяются соотношением

$$\xi = \frac{\xi_1/l}{\xi_m/m}. \quad (46)$$

Тогда СВ  $\xi$  имеет распределение  $F(l, m)$ . Если  $l, m \rightarrow \infty$ , то СВ  $\xi$  распределена асимптотически нормально. Если  $l = \text{const}$ , а  $m \rightarrow \infty$ , то СВ  $\xi$  распределена асимптотически по закону  $\chi^2(l)$ .

## Алгоритм моделирования

- определяется формулой (46).

## Моделирование смеси распределений

Конечной смесью  $L < \infty$  распределений с плотностями распределения компонент смеси  $\{f_l(x)\} \{l=1, \bar{L}\}$  называется плотность распределения  $p_\xi(x)$  вида:

$$p_\xi(x) = \sum_{l=1}^L \pi_l f_l(x),$$

где  $\{\pi_l\}$  – удельные веса (априорные вероятности) компонент смеси:

$$\pi_1 + \dots + \pi_L = 1$$

В ППП СТАТМОД реализован алгоритм моделирования смеси двух нормальных распределений вида:

$$p_\xi(x) = (1 - \pi) f(x; \mu_1, \sigma_1^2) + \pi f(x; \mu_2, \sigma_2^2), \quad (47)$$

где:  $1 - \pi, \pi (\pi > 0)$  – соответственно удельные веса первой и второй компонент смеси;  $f(x; \mu_2, \sigma_2^2)$  – плотность распределения  $N_1(\mu, \sigma^2)$ .

В частном случае, когда  $\mu_1 = \mu_2, \sigma_1 \neq \sigma_2, 0 < \pi < 0.5$ , модель (47) называется моделью засорений Тьюки-Хьюбера. При этом:  $f(x; \mu_2, \sigma_2^2)$  плотность засоряющего распределения;  $\pi$  – уровень засорения;  $f(x; \mu_1, \sigma_1^2)$  – плотность исходного распределения. Если  $0 < \pi < 0.5$ , то наблюдения с распределением  $N_1(\mu, \sigma_2^2)$  в модели (47) интерпретируются как аномальные, то есть резко выделяющиеся из основной массы наблюдений, распределенных по закону  $N_1(\mu_1, \sigma_1^2)$ .

## Алгоритм моделирования

Алгоритм моделирования СВ  $\xi$  с плотностью распределения (47) состоит из следующих шагов:

1. моделирование реализации а БСВ;
2. моделирование реализации  $x$  СВ  $\eta \sim N_1(\mu_v, \sigma_v^2)$ , где

$$v = \begin{cases} 1, & \text{если } a < 1 - \pi; \\ 2, & \text{если } 1 - \pi \leq a < 1. \end{cases}$$

3. принятие решения о том, что реализацией  $\xi$  является величина  $x$ .

Коэффициент использования БСВ  $k = \frac{1}{l+1}$ , где  $l$ - количество БСВ, используемых для моделирования одной реализации стандартной нормальной СВ(в реализованном в ППП СТАТМОД алгоритме моделирования смеси  $l=12$ ).

## Моделирование случайных величин с заданной гистограммой

Гистограммой  $\hat{p}_\xi(x)$  называется непараметрическая оценка плотности  $p_\xi(x)$  НСВ по выборке реализаций  $x_1, \dots, x_n$  данной случайной величины, вычисленная по формуле:

$$\hat{p}_\xi(x) = \sum_{k=1}^K c_k I_{[z_{k-1}, z_k]}(x), x \in R^1, (48)$$

где  $K$  – число ячеек гистограммы;  $[z_{k-1}, z_k]$  –  $k$ -я ячейка ( $z_0 = x_{(1)}, z_k = x_{(n)}$  – порядковые статистики),  $c_k$  значение гистограммы в  $k$ -ой ячейке,  $I_A(x) = \{0, \text{если } x \notin A; 1, \text{если } x \in A\}$ . Значение  $c_k$  – это нормированная относительная частота попадания реализации из выборки  $\{x_i\}$  в  $k$ -ую ячейку гистограммы:

$$c_k = \frac{v_k}{n(z_k - z_{k-1})}, v_k = \sum_{i=1}^n I_{[z_{k-1}, z_k]}(x_i),$$

$$\sum_{k=1}^K v_k = n, \bigcup_{k=1}^K [z_{k-1}, z_k] = R^1.$$

Известно, что оценка (48) в общем случае является смешанной и несостоятельной, однако на практике графическое изображение (48) используется для визуализации распределения выборки.

Пусть по выборке реализаций  $\{x_i\}$  СВ  $\xi$ , полученных в результате натуральных или физических экспериментов, построена гистограмма (48). Опишем алгоритм моделирования реализаций  $\xi$  по заданной функции  $\hat{p}_\xi(x)$ .

## Алгоритм моделирования

Обозначим:  $p_i = P\{\xi \in [z_{i-1}, z_i]\}$ ,  $q_0 = 0, q_k = \sum_{j=1}^k p_j$ ,  $k = \overline{1, K}$ . Из (48) и условия нормировки следует, что:  $p_k = c_k(z_k - z_{k-1})$ ,  $k = \overline{1, K}$ ,  $q_K = 1$ . Тогда функция распределения, соответствующая плотности  $\hat{p}_\xi(x)$  вида (48) определяется по формуле:

$$F(x)=\begin{cases} 0, \text{если } x \leq z_0 \\ q_{k-1} + c_k (x - z_{k-1}), \text{если } z_{k-1} \leq x \leq z_k, k = \overline{1, K}, \\ 1, \text{если } x \geq z_K. \end{cases}$$

Алгоритм моделирования основан на методе обратной функции и включает следующие шаги:

1. моделирование реализации а БСВ;
2. принятие решения о том, что реализацией СВ  $\xi$  является величина  $\xi'$ , вычисляемая по формуле:

$$\xi' = F^{-1}(a) = z_{k-1} + \frac{a - q_{k-1}}{c_k}, (49)$$

если  $q_{k-1} \leq a \leq q_k, 1 \leq k \leq K$ .

Обычно полагают:  $p_k = \text{const} = 1/K (k = \overline{1, K})$ . При этом в выражении (49) имеем:  $k = [Ka] + 1$ .

Коэффициент использования БСВ  $k = 1/$

## Моделирование случайных величин с заданным полигоном частот

Данный метод можно рассматривать как модификацию метода моделирования СВ с заданной гистограммой (см. соответствующий раздел). Гистограмма  $\hat{p}_\xi(x)$  – разрывная (кусочно-постоянная) функция, в то время как истинная плотность  $p_\xi(x)$  является непрерывной. Это является причиной смещенности и несостоятельности гистограммы как оценки плотностей. В связи с этим будем в качестве оценки плотности использовать непрерывную кусочно-линейную функцию  $\Psi(x)$ , получающуюся путем сглаживания гистограммы и называемую полигоном частот. Определим функцию  $\Psi(x)$  и опишем основанный на ней алгоритм моделирования НСВ.

Пусть гистограмма плотности  $p_\xi(x)$  по выборке  $\{x_i\} (i = \overline{1, n})$  имеет  $K$  ячеек  $\{[z_{k-1}, z_k]\}, (k = \overline{1, K})$ , и  $c_k$  – значение гистограммы в ячейке (см. раздел «Моделирование СВ с заданной гистограммой»). Полигон частот  $\Psi(x)$ , имеет  $K+1$  ячейку  $\{[z'_{k-1}, z'_k]\}$  с границами:

$$z'_0 = z_0 - \frac{z_1 - z_0}{2}, z'_{k+1} = z_k + \frac{z_k - z_{k-1}}{2},$$

$$z'_k = z_{k-1} + \frac{z_k - z_{k-1}}{2}, k = \overline{1, K}.$$

Аналитическое выражение функции  $\Psi(x)$ ,

$$\Psi(x) = \sum_{k=1}^{K+1} \left( c_{k-1} + \frac{c_k - c_{k-1}}{z'_k - z'_{k-1}} (z - z'_{k-1}) \right) I_{[s'_{k-1}, s'_k]}(x),$$

где полагается  $c_0 = c_{K+1} = 0$ .

Площадь области  $S$ , ограниченной кривой  $y = \Psi(x)$  и осью абсцисс, равна:

$$S = \frac{1}{2} \sum_{k=1}^{K+1} (c_{k-1} + c_k) (z'_k - z'_{k-1}).$$

Заметим, что если  $z_k - z_{k-1} = \text{const} = h$ , то  $z'_k - z'_{k-1} = h$  и  $S=1$ ,  $p_\xi(x) = \Psi(x)$ .

Функция распределения  $F(x)$ , соответствующая плотности  $\Psi(x)$ , имеет вид:

$$F(x) = \sum_{h=1}^{K+1} (q_{k-1} + c'_{k-1} (x - z'_{k-1}) +$$

### 1.3.4. Задания

#### Равномерное распределение

1. Осуществить моделирование  $n$  реализаций СВ  $\xi \sim R(a, b)$  и исследовать точность моделирования. Положить:  $a=5$ ,  $b=10$ ,  $n=100$ .

#### Одномерное нормальное распределение

2. Используя случайные выборки реализаций объема  $n=1000$ , сравнить по точности и быстродействию методы моделирования СВ  $\xi \sim N(\mu\sigma^2)$  реализованные в ППП СТАТМОД.. Положить:  $\mu=0$ ,  $\sigma^2=1$ ;  
2.  $\mu=1$ ,  $\sigma^2=0.1$

Получить последовательность реализаций СВ  $\xi$  с «усеченным» нормальным распределением:

$$\xi = \begin{cases} x, & \text{если } x > 0 \text{ (где } x - \text{реализация СВ } \eta \sim N_1(\mu\sigma^2), \\ & \text{пропускается в противном случае} \end{cases}$$

Оценить долю пропущенных реализаций СВ  $\eta$  из  $n=1000$  смоделированных. Положить:  $\sigma^2=9$ ,  $\mu=3, 6, 9$

3. С помощью моделирования  $n = 1000$  реализаций СВ  $\xi \sim N_1(\mu\sigma^2)$  оценить вероятности

$$H\{\xi \in \Delta_k\}, \Delta_k = [\mu - k\xi, \mu + k\xi], k=1, 2, 3$$

Положить  $\sigma=1$ ,  $\mu=0$

4. Контролируемый признак  $\xi$  распределен по закону  $N_1(\mu\sigma^2)$ . Множество допустимых значений признака ограничено полем допуска  $[a, b]$  ( $\mu = (a+b)/2$ ). Оценить вероятность выхода за поле допуска по  $n=1000$  реализациям СВ  $\xi$ . Рассмотреть случаи:



- a) Увеличение степени рассеяния.  $a=57, b=63, \sigma=1, 2, 3$
- b) Смещение средних  $a=57, b=63, \sigma=1, \mu=58, 65$

### Лог-нормальное распределение.

- 5. Осуществить моделирование  $n$  реализаций  $CB \xi \sim LN(m\sigma)$   
Исследовать точность и быстродействие алгоритма моделирования.  
Положить  $m=1, \sigma=0.6, 1, 2, n=100$
- 6. Экспоненциальное распределение
- 7. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim E(\lambda)$   
Исследовать точность и быстродействие алгоритма моделирования.  
Положить  $\lambda=0.5, 1, 1.5$
- 8. С помощью моделирования  $CB \xi \sim E(\lambda)$ ,  $CB \eta \sim \gamma(1/\lambda, 1)$  и  $CB \zeta \sim WG(\lambda, 1)$  установить тождественность этих случайных величин
- 9. С помощью моделирования  $CB \xi \sim E(\lambda)$ , получить реализации  $CB \eta$ , имеющей распределение Эрланга  $\nu$ -го порядка  $\gamma(1/\lambda, \nu)$  Положить  $\lambda=1, \nu=1, 2, 3$

### Распределение Лапласа

- 10. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim E(\lambda)$   
Исследовать точность и быстродействие алгоритма моделирования.  
Положить  $\lambda=0.5, 1, 1.5$

### Распределение Вейбулла-Гнеденко

- 11. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim WG(\lambda, c)$   
Исследовать точность и быстродействие алгоритма моделирования.  
Рассмотреть случаи  $\lambda=1, 2, c=0.5, 1, 0.3, 0$   
*Примечание.* Распределение  $WG(2, c)$  называется распределением Релея,  $WG(\lambda, 1) = E(\lambda)$ .

### Гамма распределение

- 12. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim \gamma(b, \nu)$   
Исследовать точность и быстродействие алгоритма моделирования.  
Рассмотреть случаи  $b=1, 2, \nu=0.5, 2$
- 13. С помощью моделирования  $CB \xi \sim \gamma(b, \nu)$  сравнить функциональные и числовые характеристики распределений Эрланга  $\nu$ -го порядка  
Положить  $b=1, \nu=1, 2, 3$

14. Сравнить по точности и быстродействию моделирования  $n=1000$  реализаций  $CB \xi \sim \gamma(b, v)$  методы моделирования, реализованные в ППП СТАТМОД

## Бета-распределение

15. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim \gamma(v, w)$   
Исследовать точность и быстродействие алгоритма моделирования  
Положить  $v=2, w=4$
16. С помощью моделирования  $CB \xi \sim \beta(v, w)$  сравнить различные частные случаи распределения, полученные: при  $v=1, 2, w = 1, 2; v=0.2, w=1$
17. Доказать симметричность распределений  $\beta(v, w)$  и  $\beta(w, v)$
18. Осуществить моделирование и исследовать характеристики распределение по закону арксинуса, совпадающего с распределением  $\beta(0.5, 0.5)$
19. На основании анализа характеристик с помощью моделирования проиллюстрировать тождественность двух случайных величин:  $CB \xi \sim \beta(1, 1)$  и  $CB \eta \sim R(0, 1)$
20. С помощью моделирования  $CB \xi \sim \beta(v/2, w/2)$  и  $CB \eta \sim F(v, w)$  проиллюстрировать справедливость соотношения между распределениями. Положить:  $v=2, w=4, x=2$

$$P\left\{\xi \leq \frac{w}{w + vx}\right\} = P\{\eta > x\}$$

21. Сравнить по точности и быстродействию моделирования  $n=1000$  реализаций  $CB \xi \sim \beta(v, w)$  методы моделирования, реализованные в ППП СТАТМОД

## Распределение Коши

22. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim C(m, c)$   
Исследовать точность и быстродействие алгоритма моделирования  
Положить  $m=-1, 0, 1, c=1, 2, 3$
23. Сравнить по точности и быстродействию моделирования  $n$  реализаций  $CB \xi \sim C(m, c)$  методы, реализованные в ППП СТАТМОД. Положить:  $m=0, c=1$ .

## Логистическое распределение

24. Осуществить моделирование  $n=1000$  реализаций  $CB \xi \sim LG(\mu, k)$   
Исследовать точность и быстродействие алгоритма моделирования  
Положить  $\mu=0, k=1$

25. С помощью статистического моделирования СВ  $\xi \sim LG(\mu, k)$  оценить зависимость стандартного отклонения  $\delta$  СВ  $\xi$  от  $k$ . Положить:  $\mu=0, k=0.55, 1.1, 1.65, 2.2$ .

## Хи-квадрат распределение

26. Осуществить моделирование  $n=1000$  реализаций СВ  $\xi \sim \chi^2(m)$   
Исследовать точность и быстродействие алгоритма моделирования  
Положить  $m=1, 2, 3, 5, 50$ .
27. С помощью моделирования СВ  $\xi \sim \chi^2(m)$  и СВ  $\eta \sim \gamma(2, m/2)$   
графического анализа и анализа характеристик проиллюстрировать  
тождественность этих случайных величин
28. С помощью моделирования СВ  $\xi \sim \chi^2(m)$  и СВ  $\eta \sim \Pi(\lambda)$  ( $\lambda=x/2$ )  
проиллюстрировать справедливость соотношения между  
распределениями

$$P\{\xi > x\} = P\left\{\eta \leq \frac{m}{2} - 1\right\}$$

## Распределение Стьюдента

29. Осуществить моделирование  $n=1000$  реализаций СВ  $\xi \sim t(m)$   
Исследовать точность и быстродействие алгоритма моделирования  
Положить  $m=1$ .
30. С помощью моделирования СВ  $\xi \sim t(m)$  и СВ  $F(1, m)$ , СВ  $\eta \sim N_1(0, 1)$   
исследовать возможность и точность аппроксимации  
распределения  $\chi^2$  распределением Фишера и стандартным  
нормальным. Положить:  $m=20, 30, 40, 50$ .

## Распределение Фишера

31. Осуществить моделирование  $n$  реализаций НСВ  $\xi \sim F_{m,l}(z)$  Провести  
графически анализ и исследовать точность моделирования при  
следующих значениях параметров:  
а)  $m=3, l=5$ ; б)  $m=6, l=3$ ; и)  $m=6, l=12$ . Положить:  $n=20, 50, 100, 500$

Моделирование НСВ с заданной гистограммой и полигоном частот

Обозначим:  $K$  – число ячеек гистограммы;  $[z_{i-1}, z_i]$  –  $i$  – ячейка гистограммы,  $i=1, K$ ;  $c_i$  – значение гистограммы в  $i$  – ой ячейке;  $p_i$  – вероятность попадания СВ  $\xi$  в  $i$  – ячейку.

Требуется осуществить моделирование НСВ  $\xi$  по заданной гистограмме и полигону частот, провести графический анализ и исследовать точность моделирования для следующих значений параметров.

$K=6; z_i=i, i=0,1,\dots,K; c_1=0.1, c_2=c_4=0.2, c_3=0.4, c_5=c_6=0.05$

$K=10; z_0=0\{z_i, p_i\}$  заданы в таблице:

<b>i</b>	1	2	3	4	5	6	7	8	9	10
<b>z<sub>i</sub></b>	0. 1	0. 3	0. 4	0. 5	0. 6	0. 7	0. 9	1. 1	1. 4	2. 0
<b>p<sub>i</sub></b>	0. 03	0. 18	0. 15	0. 20	0. 15	0. 08	0. 07	0. 06	0. 05	0. 03

## Смесь гаусовских распределений

32. Плотность распределения НСВ  $\xi$  представляет собой смесь двух гаусовских распределений  $N_1(\mu_1, \sigma_1^2)$  и  $N_2(\mu_2, \sigma_2^2)$  (модель засорений Тьюки-Хьюбера с уровнем засорений  $\varepsilon$ ). Осуществить  $n$  реализаций моделирования СВ  $\xi$  и провести графический анализ результатов моделирования при следующих значениях параметров:

- 1)  $\mu_1=\mu_2=0; \sigma_2=\alpha\sigma_1, \alpha=1/2, 2, 4, \sigma_1=1; \varepsilon=0.05, 0.45, n=20, 50, 100, 500$
- 2)  $\mu_1=0, \mu_2=6; \sigma_2=\alpha\sigma_1, \alpha=1/2, 1, 2, \sigma_1=1; \varepsilon=0.05, 0.45, n=20, 50, 100, 500$

# Глава 2. Метод Монте-Карло и его применения

## 2.1. Общая схема метода Монте-Карло

Метод Монте-Карло – это численный метод исследования математических моделей сложных систем, основанный на моделировании случайных элементов и последующем статистическом анализе результатов моделирования.

Метод Монте-Карло реализуется с помощью следующих шагов:

1) подбирают такую случайную величину  $\xi$ , что выполняются условия

$$E\{\xi\} = a, \quad D\{\xi\} < \infty,$$

где  $a$  – скалярная величина, приближенное значение, которое надо найти;

2) моделируют  $n$  независимых реализаций случайной величины  $\xi$ :  $\xi_1, \xi_2, \dots, \xi_n$ ;

3) по случайной выборке  $\{\xi_i\}, i = \overline{1, n}$ , строят выборочную оценку  $a$ :

$$\hat{a}_n = \frac{1}{n} \sum_{i=1}^n \xi_i.$$

Точность вычислений характеризуют либо с помощью доверительного интервала, определяемого интервалом

$$\left| \hat{a}_n - a \right| < x_\beta \sqrt{\frac{D\{\xi\}}{n}},$$

вероятность выполнения которого приблизительно равна  $\beta$ ; (например при  $\beta = 0,997$   $x_\beta = 3$ ), либо с помощью так называемой вероятной ошибки метода Монте-Карло:

$$\tau_n = 0,6745 \sqrt{\frac{D\{\xi\}}{n}},$$

где численный множитель 0,6745 является решением уравнений

$$P\left\{\left|\hat{a}_n - a\right| < r_n\right\} = \frac{1}{2} = P\left\{\left|\hat{a}_n - a\right| > r_n\right\},$$

т.е. одинаково вероятны ошибки, больше, чем  $r_n$  и ошибки, меньшие, чем  $r_n$ .

## 2.2. Вычисление определенного интеграла методом Монте-Карло

Рассмотрим задачу приближенного вычисления интеграла

$$a = \int_A g(x) dx,$$

где  $x \in R^n$ ,  $g(x) \in R^1$ ,  $A \in R^n$  - подмножество из  $R^n$ . При  $n=1$  имеем определенный интеграл вида

$$a = \int_{x_0}^{x_1} g(x) dx.$$

Заметим, что схема вычислений как многомерных, так и одномерных интегралов, абсолютно аналогична.

Пусть  $\eta$  – произвольная случайная величина с плотностью распределения вероятностей  $P_\eta(x)$ ,  $x \in [x_0, x_1]$

Предполагается только, что существуют моменты случайных величин, встречающиеся ниже. Рассмотрим случайную величину, являющуюся функциональным преобразованием случайной величины  $\eta$ .

$$\xi = \frac{g(\eta)}{P_\eta(\eta)}$$

Можно показать, что  $M\{\xi\} = a$ ,  $D\{\xi\} < \infty$ . Поэтому в качестве приближенного значения интеграла можно использовать статистическую оценку  $\hat{a}_n$ , построенную в выборке из  $n$  независимых случайных величин  $\xi_1, \xi_2, \dots, \xi_n$ :

$$\hat{a}_n = \frac{1}{n} \sum_{i=1}^n \xi_i = \frac{1}{n} \sum_{i=1}^n \frac{g(\eta_i)}{P_\eta(\eta_i)}.$$

### Пример.

Вычислить интеграл  $\int_0^\infty \int_0^\infty e^{-\left(2x + \frac{y}{2}\right)} \sqrt{1 + \ln((y-x)^2)} dx dy$  Методом Монте-Карло.

Решение.

Выберем случайный вектор  $\eta = (\eta_1, \eta_2)$  с плотностью распределения  $\rho_\eta(x, y)$ , равной  $\rho_\eta(x, y) = e^{-\left(2x + \frac{y}{2}\right)}$ . Для  $\rho_\eta(x, y)$  выполняются условия

$$\rho_\eta(x, y) > 0, (x, y) \in [0, \infty) \times (0, \infty], \int_0^\infty \int_0^\infty \rho_\eta(x, y) dx dy = 1.$$

Ее можно рассматривать как совместную плотность распределения двух независимых экспоненциально распределенных случайных

величин  $\eta_1$  и  $\eta_2$  с плотностями распределения  $\rho_{\eta_1}(x) = 2e^{-2x}$  и

$\rho_{\eta_2}(y) = \frac{1}{2}e^{-\frac{y}{2}}$ . Тогда

$$\int_0^\infty \int_0^\infty e^{-\left(2x + \frac{y}{2}\right)} \sqrt{1 + \ln((y-x)^2)} dx dy = \int_0^\infty \int_0^\infty \rho_\eta(x, y) \sqrt{1 + \ln((y-x)^2)} dx dy = E\left\{\sqrt{1 + \ln((\eta_2 - \eta_1)^2)}\right\}$$

Пусть в результате моделирования независимых компонент случайного вектора  $\eta = (\eta_1, \eta_2)$  получена случайная выборка  $(x_1, y_1), \dots, (x_n, y_n)$ . Согласно методу Монте-Карло в качестве приближенного значения интеграла следует использовать статистику

$$\hat{a} = \frac{1}{n} \sum_{i=1}^n \sqrt{1 + \ln((y_i - x_i)^2)}.$$

### Текст программы:

```
#include <fstream.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>
#include <time.h>
// basic variate
double bv(double*a, double*b, double*c, int m, int
n) {
    int i, tmp;
    double result = c;
    for (i = 0; i < n; i++)
        result += b[i] * a[i];
    tmp = (int) (result / m);
    result -= ((double) tmp) * ((double) m);
    for (i = 1; i < n; i++)
        a[i-1] = a[i];
    a[n-1] = result;
    return result / m;
}
// exponential variate p(x) = l * exp(-lx), x >= 0
double ev(double l, double a) {
    return -log(1-a) / l;
}
double avg(double*a, int n) {
    double s = 0;
    for (int i = 0; i < n; i++)
        s += (a[i]);
    return s / n;
}
double var(double*a, int n) {
```

```

double s=0,s=2;
for(int i=0;i<n;i++){
s+=a[i];
s2+=a[i]*a[i];
}
return(s2-s*s/n)/(n-1);
}

void main() {
int i,j;
const int n=1000;
double a[2][n]; //two pseudorandom sequences
double b[2][n]; //coefficients
double c[2];      //increment
int m[2]          //modules

//initial values
srand((unsigned) time (NULL) );
for(j=0;j<2;j++){
m[j] =RAND_MAX-rand();
for(i=0;i<n;i++){
a[j][i]=rand()%m[j];
b[j][i]=rand();
}
c[j]=rand();
}

//computation
double tmp;
double*A =new double[n];
double*x =new double[n];
double*y =new double[n];
ofstream out("integral.txt");
for(j=1;j<=5;j++){
for(i=0;i<n;i++){
do{
do{
x[i]=ev(2,dv(a[0],b[0],c[0],m[0],n));
y[i]=ev(0.5,dv(a[1],b[1],c[1],m[1],n));
}while(x[i]-y[i]==0);
tmp= 1+log(pow(y[i]-x[i],2));
}while(tmp<0);
A[i]=sqrt(tmp);
}
out<<avg(A,n)<<"\n";
out<<var(A,n)<<"\n";
}

```



```
out<<0.6745*sqrt(var(A,n)/n)<<"\n\n";  
}  
}
```

Некоторые статистические результаты моделирования:  
Запустим программу по 5 раз для различных значений  $n$ .

№ таблицы	№ эксперимент а	n	Ответ	Дисперсия	Вероятная ошибка
1	1	10	1.48915	0.268116	0.110444
	2		1.37017	0.446164	0.142472
	3		1.33855	0.097724	0.0656633
	4		1.49458	0.13842	0.0793562
	5		1.45071	0.334751	0.123408
2	6	20	1.4981	0.322398	0.0856373
	7		1.21044	0.326385	0.0861652
	8		1.55455	0.21716	0.070284
	9		1.3085	0.366371	0.0912909
	10		1.47881	0.150203	0.0584529
3	11	50	1.43942	0.268284	0.0494077
	12		1.47837	0.217818	0.0445189
	13		1.40335	0.258855	0.045317
	14		1.43447	0.295186	0.0518256
	15		1.42966	0.288469	0.0512326
4	16	100	1.39962	0.219702	0.0316154
	17		1.43421	0.254309	0.0340144
	18		1.42465	0.236301	0.032788
	19		1.38404	0.253826	0.0339821
	20		1.52162	0.257893	0.0342533
5	21	500	1.41239	0.272168	0.0157368
	22		1.42945	0.271058	0.0157046
	23		1.41602	0.259105	0.0153545
	24		1.43384	0.243818	0.0148946
	25		1.40812	0.246367	0.0149723
6	26	1000	1.42914	0.259182	0.00343388
	27		1.42468	0.260844	0.00344486
	28		1.42849	0.255745	0.00341103
	29		1.42793	0.262582	0.00345633
	30		1.42642	0.257236	0.00342295

Вычислить интеграл аналитически не удастся. Полученное с помощью программы Mathematica4.1 точное значение интеграла равно 1.4221. Как видим, полученные в результате выполнения программы значения интеграла близки к точному, однако точность в данном случае невысока. Точность возрастает с увеличением объема выборки. Вероятная ошибка уменьшается.

## 2.3. Решение системы алгебраических уравнений методом Монте-Карло

Моделирование цепей Маркова оказывается естественным образом связанным с решением большого числа задач, и, в первую очередь, с решением систем линейных алгебраических уравнений.

Рассмотрим простейшую вычислительную схему для решения линейных алгебраических уравнений, которая была впервые предложена Дж. Нейманом и С. Уламом [12].

Пусть система алгебраических уравнений задана в виде

$$x = Ax + f \quad (4.3.1)$$

где  $x = (x_1, \dots, x_n)^T$  – вектор-столбец неизвестных,  $f = (f_1, \dots, f_n)^T$  – вектор правых частей и  $A = (a_{ij})$ ,  $i, j = \overline{1, n}$  – матрица системы.

Предположим, что наибольшее по модулю характеристическое число матрицы  $A$  меньше единицы, так что сходиться метод последовательных приближений

$$x^{(k)} = Ax^{(k-1)} + f, \quad k = 1, 2, \dots \quad (4.3.2)$$

Достаточным условием для того, чтобы все характеристические числа матрицы  $A$  лежали внутри единичного круга на комплексной плоскости, то есть, чтобы  $|\lambda_i| < 1$ ,  $i = \overline{1, n}$ , может служить неравенство

$$\sum_{i,j=1}^n a_{ij}^2 < 1 \text{ или неравенство } \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| < 1.$$

Если положить, что  $x^{(0)} = f$ , то

$$\begin{aligned} x^{(k)} &= (A^k + A^{k-1} + \dots + A + I_n)f, \quad k = 1, 2, \dots, \\ \bar{x} &= \lim_{k \rightarrow \infty} x^{(k)} = \lim_{k \rightarrow \infty} (I_n + A + \dots + A^k)f = (I_n - A)^{-1}f \end{aligned} \quad (4.3.3)$$

– точное решение системы (4.3.1)

Рассмотрим задачу о вычислении скалярного произведения  $(h, \bar{x})$ , где  $h$  – заданный вектор.

Мы будем связывать с системой (4.3.1) и вектором  $h$  некоторую фиксированную цепь Маркова из множества цепей, определяемых парой  $\{\pi, P\}$ :

$$\pi = (\pi_1, \dots, \pi_n)^T, \pi_i \geq 0, \sum_{i=1}^n \pi_i = 1, \quad (4.3.4)$$

$$P = (p_{ij}), \sum_{j=1}^n p_{ij} = 1, p_{ij} \geq 0, i, j = \overline{1, n}, \quad (4.3.5)$$

для которых выполнены условия:

$$\pi_i > 0, \text{ если } h_i \neq 0,$$

$$p_{ij} > 0, \text{ если } a_{ij} \neq 0; i, j = \overline{1, n}; k = 1, 2, \dots \quad (4.3.6)$$

Положим

$$g_i^{(0)} = \begin{cases} h_i / \pi_i, & \pi_i > 0, \\ 0, & \pi_i = 0, \end{cases} \quad g_{ij}^{(k)} = \begin{cases} a_{ij} / p_{ij}, & p_{ij} > 0, \\ 0, & p_{ij} = 0, \end{cases} \quad (4.3.7)$$

Зададимся некоторым целым  $N$ , и будем рассматривать траектории цепи Маркова длины  $N > 0$ . Объект, переходы которого описываются цепью Маркова, условимся называть частицей и считать, что эта частица изменяет свои состояния (движется в соответствии с траекторией цепи). Движущейся частице приписывается "вес"  $Q_k$ , который изменяется при движении ее по траектории  $i_0 \rightarrow i_1 \rightarrow \dots \rightarrow i_n$  следующим образом. В начальный момент, когда она находится в состоянии  $i_0$ , частица имеет вес  $Q_0 = g_{i_0}^{(0)}$ , при переходе из состояния  $i_0$  в состояние  $i_1$  ее вес становится равным  $Q_1 = g_{i_0, i_1}^{(0)} g_{i_0, i_1}^{(1)}$  и т.д., т.е.

$$Q_m = Q_{m-1} g_{i_{m-1}, i_m}^{(m)}. \quad (4.3.8)$$

Введем случайную величину  $\xi_N$ , определенную на траекториях Марковской цепи длины  $N$

$$\xi_N = \sum_{m=0}^N Q_m f_{i_m}, \quad (4.3.9)$$

Используя формулу умножения вероятностей, найдем

$$P\{i_0 \rightarrow i_1 \rightarrow \dots \rightarrow i_N\} = \pi_{i_0} p_{i_0, i_1} \dots p_{i_{N-1}, i_N} \quad (4.3.10)$$

Тогда математическое ожидание СВ  $\xi_N$  равно

$$\begin{aligned} E\{\xi_N\} &= \sum_{i_0=1}^n \dots \sum_{i_{N-1}=1}^n \sum_{m=0}^N Q_m f_{i_m} p_{i_0} p_{i_0, i_1} \dots p_{i_{N-1}, i_N} = \\ &= \sum_{i_0=1}^n \dots \sum_{i_{N-1}=1}^n p_{i_0} p_{i_0, i_1} \dots p_{i_{N-1}, i_N} \sum_{m=0}^N g_{i_0}^{(0)} g_{i_0, i_1}^{(1)} \dots g_{i_{m-1}, i_m}^{(m)} f_{i_m} \end{aligned} \quad (4.3.11)$$

Подставим выражения для  $g_{i_0}^{(0)} g_{i_0, i_1}^{(1)} \dots g_{i_{m-1}, i_m}^{(m)}$ , определяемые по формуле (4.3.7), получим

$$E\{\xi_N\} = \sum_{m=0}^N \sum_{i_0=1}^n \dots \sum_{i_{N-1}=1}^n h_{i_0} a_{i_0, i_1} \dots a_{i_{N-1}, i_N} f_{i_m} \quad (4.3.12)$$

Используя определение произведения матриц получим

$$A^2 = \|a_{i_0, i_1}\|_1^n \times \|a_{i_1, i_2}\|_1^n = \left\| \sum_{i_1=1}^n a_{i_0, i_1} a_{i_1, i_2} \right\|_{i_0, i_2=1}^n, \quad (4.3.13)$$

т.е.

$$E\{\xi_N\} = \left( h, \sum_{m=0}^N A^m f \right). \quad (4.3.14)$$

Из (4.3.3) и (4.3.14) следует, что при  $N \rightarrow \infty$   $E\{\xi_N\}$  стремиться к  $(h, \bar{x})$ .

Итак, доказана следующая теорема.

**Теорема 4.3.1.** Если все характеристические числа матрицы  $A$  по абсолютной величине меньше единицы, то математическое ожидание случайной величины  $\xi_h = \sum_{m=0}^{\infty} Q_m f_{i_m}$  равно

$$E\{\xi_N\} = (h, \bar{x}). \quad (4.3.15)$$

Для получения приближенного значения  $x_j$  —  $j$ -й компоненты вектора  $\bar{x}$  (решения системы (4.3.1)) выбираем в качестве  $h$  единичный вектор  $e_j = (0, 0, \dots, 0, 1, 0, \dots, 0)^T$ , в котором лишь на  $j$ -м месте стоит единица. Тогда скалярное произведение (4.3.15) равно  $x_j$ . Используя результаты (3) главы, моделируем  $l$  реализаций цепи Маркова:

$$i_0^{(k)} \rightarrow i_1^{(k)} \rightarrow \dots \rightarrow i_j^{(k)} \rightarrow \dots, \quad k = \overline{1, l}, \quad j \leq N. \quad (4.3.16)$$

Вычисляем вдоль цепей (4.3.16) веса  $Q_{i_m}^{(k)}$  согласно формуле (4.3.8), тогда

$$\xi_{e_j}^{(k)} = \sum_{m=0}^N Q_m^{e_j} f_{i_m}. \quad (4.3.17)$$

Приближенное значение для  $x_j$  имеет вид

$$x_j \approx \frac{1}{l} \sum_{k=1}^l \xi_{e_j}^{(k)}. \quad (4.3.18)$$

Пример.

Решить систему линейных алгебраических уравнений с использованием метода Монте-Карло:

$$\begin{cases} 1.1x - 0.8y = 0.1 \\ -0.7x - 0.3y = 0.1 \end{cases}$$

Решение:

Приведем систему к виду, при котором применим метод Монте-Карло:

$$\begin{cases} x = -0.1x + 0.8y + 0.1 \\ y = 0.4x - 0.1y - 0.2 \end{cases}$$

Таким образом, система имеет вид:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -0.1 & 0.8 \\ 0.4 & -0.1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.1 \\ -0.2 \end{pmatrix}$$

Все собственные значения матрицы  $\Phi$  по модулю меньше 1 и можно применять метод Монте-Карло.

Определим некоторый вектор  $h$  следующим образом:

$$h = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ для вычисления } x, \quad h = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ для вычисления } y.$$

Моделируем вспомогательную цепь Маркова:

$$i_0 \rightarrow i_1 \rightarrow \dots \rightarrow i_{N-1}, \text{ где } i_k \in X = \{1, 2\}.$$

$$\text{Вектор вероятностей начальных состояний цепи Маркова: } \pi = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}.$$

$$\text{Матрица переходных вероятностей имеет вид: } P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

Каждому состоянию цепи Маркова приписываем веса, которые вычисляются по формулам:

$$Q_{i_0} = g_{i_0} = \begin{cases} \frac{h_{i_0}}{\pi_{i_0}}, \pi_{i_0} > 0 \\ 0, \pi_{i_0} = 0 \end{cases}, \quad Q_{i_k} = Q_{i_{k-1}} g_{i_{k-1}} = \begin{cases} \frac{a_{i_{k-1}, i_k}}{p_{i_{k-1}, i_k}}, p_{i_{k-1}, i_k} > 0 \\ 0, p_{i_{k-1}, i_k} = 0 \end{cases}$$

Теперь можем построить СВ  $\xi_N$  по формуле:

$$\xi_N^{(l)} = \sum_{n=0}^N Q_{i_n} f_{i_n},$$

где  $l = \overline{1, L}$  - номер реализации цепи Маркова. Тогда приближенное решение вычисляется по формуле:

$$x \approx \frac{1}{L} \sum_{l=1}^L \xi_N^{(l)} \text{ или } y \approx \frac{1}{L} \sum_{l=1}^L \xi_N^{(l)} \text{ в зависимости от вектора } h.$$

Данный алгоритм реализован на языке C++.

Текст программы:

```
#include <iostream.h>
#include <stdlib.h>
#include <time.h>

void main(){
    int n;           //Размерность системы
    float x=0,y=0;   //Решение системы
    float **a;       //Исходная матрица
    float *f;        //Правая часть системы
    float *h;
    float *pi;       //Вектор нач. вероятностей цепи Маркова
    float **p;       //Матрица переходных состояний цепи Маркова
    int N;           //Длина цепи Маркова
    int *j;          //Цепь Маркова
    float *Q;        //Веса состояний цепи Маркова
    float *ksi;      //CB
    int m;           //Количество реализаций цепи Маркова
    float alpha;     //БСВ

    n=2;

    a=new float*[n];
    for(int k=0;k<n;k++)
        a[k]=new float[n];
    f=new float[n];
```

```

h=new float[n];
pi=new float[n];
p=new float*[n];
for(k=0;k<n;k++)
    p[k]=new float[n];

```

```

N=1000;

```

```

i=new int[N+1];
Q=new float[N+1];

```

```

m=10000;

```

```

ksi=new float[m];
for(int j=0;j<m;j++)
    ksi[j]=0;
a[0][0]=-0.1; a[0][1]=0.8;
a[1][0]=0.4; a[1][1]=-0.1;
f[0]=0.1;
f[1]=-0.2;
h[1]=1;
h[0]=0;
pi[1]=0.5;
pi[0]=0.5;
p[0][0]=0.5; p[0][1]=0.5;
p[1][0]=0.5; p[1][1]=0.5;

```

*//Моделируем m цепей Маркова длины N*

```

for(j=0;j<m;j++)
{
    alpha=rand()/float(RAND_MAX);
    if(alpha<pi[0]) i[0]=0;           //реализуется 1-е состояние
    else i[0]=1;                     //реализуется 2-е состояние
    for(k=1;k<=N;k++)

```



```

    {
        alpha=rand()/float(RAND_MAX);
        if(alpha<0.5) i[k]=0;
        else i[k]=1;
    }

    //Вычисляем веса цепи Маркова
    if(pi[i[0]]>0) Q[0]=h[i[0]]/pi[i[0]];
    else Q[0]=0;
    for(k=1;k<=N;k++)
    {
        if(p[i[k-1]][i[k]]>0)
            Q[k]=Q[k-1]*a[i[k-1]][i[k]]/p[i[k-1]][i[k]];
        else Q[k]=0;
    }
    for(k=0;k<=N;k++)
        ksi[j]=ksi[j]+Q[k]*f[i[k]];
}

for(k=0;k<m;k++)
    x=x+ksi[k];

x=x/m;

cout << x;
}

```

Результат работы программы:

x=-0.0559199

y=-0.199699

Точное решение системы:

x=-5/89≈0.056179775

$$y = -18/89 \approx -0.202247191$$

## 2.4. Задания

Вычислить интегралы:

$$1. \int_0^1 e^x dx,$$

$$2. \int_0^{\frac{\pi}{2}} \cos x dx,$$

$$3. \int_0^{\infty} e^{-2x} \sqrt{1+x} dx,$$

$$4. \int_{-\infty}^{\infty} e^{-x^2} \sqrt{1+x^2} dx,$$

$$5. \int_0^1 \frac{\sin x}{x} dx,$$

$$6. \int_0^{\infty} e^{-x} \cos^2 x dx,$$

$$7. \int_0^2 \frac{\operatorname{tg} x}{1+x^2} dx,$$

$$8. \int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} e^{-\frac{x^2+y^2}{2}} \ln \left( 1 + \left( \frac{x}{y} \right)^2 \right) dx dy,$$

$$9. \int_{-\infty}^{\infty} x^3 \exp \left\{ -\frac{x^2}{2} \right\} dx$$

$$10. \int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} e^{-\frac{x^2+y^2}{2}} \sqrt{1+\sin^2(x+y)} dx dy,$$

$$11. \int_0^1 (e^x - 1) \ln \frac{1}{x} dx,$$

$$12. \int_0^{\infty} e^{-x^2} dx,$$

$$13. \int_0^{\infty} \sin \frac{1}{x} \cdot \frac{dx}{x^2}$$

$$14. \int_0^{\infty} \frac{x \ln x}{1+x^2} dx,$$

$$15. \iint_A \frac{dx dy}{x^2 + y^2}, A = \{(x, y) : |y| \leq x^2, x^2 + y^2 \leq 1\}$$

$$16. \iint_{y \geq x^2 + 1} \frac{dx dy}{x^4 + y^4}$$

$$17. \iint_{x^2+y^2} \ln \left( \frac{1}{\sqrt{x^2+y^2}} \right) dx dy,$$

$$18. \int_0^{\infty} \frac{\cos x}{\sqrt{x}} dx,$$

$$19. \int_0^1 x^{15} \sqrt{1+3x^8} dx,$$

$$20. \int_0^{\pi} (x \sin x)^2 dx.$$

# Глава 3. Система моделирования GPSS

Цель работы: научиться использовать язык GPSS (General Purpose Simulation System) для исследования процедур имитационного моделирования сложных технических объектов, представленных как системы массового обслуживания.

## 3.1. Краткие сведения из теории

В математических моделях (ММ) сложных объектов, представленных в виде систем массового обслуживания (СМО), фигурируют средства обслуживания, называемые обслуживающими аппаратами (ОА), и обслуживаемые заявки, называемые транзактами. Так, в модели производственной линии ОА отображают рабочие места, а транзакты - поступающие на обработку детали, материалы, инструмент.

Состояние СМО характеризуется состояниями ОА, транзактов и очередей к ОА. Состояние ОА описывается двоичной переменной, которая может принимать значения "занят" или "свободен". Переменная, характеризующая состояние транзакта, может иметь значения "обслуживания" или "ожидания". Состояние очереди характеризуется количеством находящихся в ней транзактов.

Имитационная модель СМО представляет собой алгоритм, отражающий поведение СМО, т.е. отражающий изменения состояния СМО во времени при заданных потоках заявок, поступающих на входы системы. Параметры входных потоков заявок - внешние параметры СМО. Выходными параметрами являются величины, характеризующие свойства системы - качество ее функционирования. Примеры выходных параметров: производительность СМО - среднее число заявок, обслуживаемых в единицу времени; коэффициенты загрузки оборудования - отношение времен обслуживания к общему времени в каждом ОА; среднее время обслуживания одной заявки. Основное свойство ОА, учитываемое в модели СМО, - это затраты времени на обслуживание, поэтому внутренними параметрами в модели СМО являются величины, характеризующие это свойство ОА. Обычно время обслуживания рассматривается как случайная величина и в качестве внутренних параметров фигурируют параметры законов распределения этой величины.

Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и интенсивностях поступления заявок на входы, при вариациях параметров ОА, при различных дисциплинах обслуживания

заявок. Дисциплина обслуживания - правило, по которому заявки поступают из очередей на обслуживание. Величина, характеризующее право на первоочередное обслуживание, называется приоритетом. В моделях СМО заявки, приходящие на вход занятого ОА, образуют очереди, отдельные для заявок каждого приоритета. При освобождении ОА на обслуживание принимается заявка из непустой очереди с наиболее высоким приоритетом.

Основной тип ОА - устройства, именно в них происходит обработка транзактов с затратами времени. К ОА относятся также накопители (памяти), отображающие средства хранения обрабатываемых деталей в производственных линиях или обрабатываемых данных в вычислительных системах. Накопители характеризуются не временами обслуживания заявок, а емкостью - максимально возможным количеством одновременно находящихся в накопителе заявок.

К элементам имитационных моделей СМО кроме ОА относят также узлы и источники заявок. Связи ОА между собой реализуют узлы, т.е. характеризуют правила, по которым заявки направляются к тому или иному ОА.

Для описания моделей СМО при их исследовании на ЭВМ разработаны специальные языки имитационного моделирования. Существуют общецелевые языки, ориентированные на описание широкого класса СМО в различных предметных областях, и специализированные языки, предназначенные для анализа систем определенного типа. Примером общецелевых языков служит широко распространенный язык GPSS, примером специализированного языка - язык МПЛ/ВС моделирования вычислительных систем.

## **3.2. Основные правила и операторы языка GPSS**

Для описания имитационной модели на языке GPSS полезно представить ее в виде схемы, на которой отображаются элементы СМО - устройства, накопители, узлы и источники. Описание на языке GPSS есть совокупность операторов (блоков), характеризующих процессы обработки заявок. Имеются операторы и для отображения возникновения заявок, задержки их в ОА, занятия памяти, выхода из СМО, изменения параметров заявок (например, приоритетов), вывода на печать накопленной информации, характеризующей загрузку устройств, заполненность очередей и т.п.

Каждый транзакт, присутствующий в модели, может иметь до 12 параметров. Существуют операторы, с помощью которых можно изменять значения любых параметров транзактов, и операторы, характер исполнения которых зависит от значений того или иного параметра обслуживаемого транзакта.

Пути продвижения заявок между ОА отображаются последовательностью операторов в описании модели на языке GPSS специальными операторами передачи управления (перехода). Для моделирования используется событийный метод. Соблюдение правильной временной последовательности имитации событий в СМО обеспечивается интерпретатором GPSSPC - программной системой, реализующий алгоритмы имитационного моделирования.

### 3.2.1. Структура операторов GPSS

В записи оператора выделяют три части: метку, название, поле переменных.

Пример оператора:

```
110 L1 GENERATE 30,5          Первый сегмент модели
2....6.8.....18..19.....70
    <метка><название> <поле переменных>          <комментарии>
```

В поле переменных выделяют подполя, разделяемые при записи запятыми и служащие для указания чисел, стандартных числовых атрибутов (СЧА), символов, обозначающих метки, идентификаторов, указателей разновидностей операторов и т.п. Подполя могут быть пустыми. Возможна запись комментария после последнего непустого поля через пробел.

Стандартные числовые атрибуты служат для сокращенного указания различных величин, фигурирующих в модели.

Примеры СЧА: K126 - константа, равная 126; V2 - переменная N 2; Q4 - длина очереди N 4; X5 - хранимая величина N 5; FN7 - функция N 7; P4 - значение параметра N 4 транзакта; \*6 - содержимое параметра N 6 транзакта; S\*3 (или FN\*3) - память (или функция), определенная в параметре N 3 транзакта.

### 3.2.2. Основные операторы языка GPSS

Основные операторы языка GPSS приведены в виде примеров с конкретными значениями подполей в поле переменных.

GENERATE 12,4,50,5,1 - генерация транзактов, интервалы времени между появлениями транзактов распределены равномерно в диапазоне [12-4, 12+4], первый транзакт появится с задержкой в 50 единиц модельного времени, всего будет создано 5 транзактов, приоритет транзактов равен единице.

GENERATE 12,4,50,,1 - то же, но количество генерируемых транзактов неограничено.

GENERATE 6, FN\$FFF,50,5,1 - то же, но интервал времени между появлениями транзактов есть целая часть произведения числа 6 на значение функции FFF.

FNK FUNCTION RN1,C4  
0,0/0.1,0.8/0.5,1.6/1.0,1.9

- описание функции FNK, ее аргументом является случайная величина (на это указывает значение RN1), равномерно распределенная в диапазоне [0,1], функция является непрерывной числовой (указатель C), заданной таблично четырьмя точками: (0;0), (0.1; 0.8), (0.5, 1.6), (1.0; 1.9).

FNK FUNCTION \*2,D4  
0,12/1,9/2,8/3,6

- то же, но аргументом является значение второго параметра транзакта, для которого вычисляется значение дискретной величины (D) числовой функции FNK, заданной таблично четырьмя узловыми точками. Это текущее значение округляется до ближайшего большего значения аргумента в узловой точке.

SEIZE PLOT - занятие устройства PLOT приходящим на его вход транзактом; если устройство занято, то транзакт задерживается в очереди к этому устройству.

RELEASE PLOT - освобождение устройства PLOT обслуженным транзактом.

ENTER MEM,12 - занятие транзактом 12 единиц емкости в накопителе MEM.

LEAVE MEM,\*2 - освобождение k единиц памяти в накопителе MEM, гдк k - значение 2-го параметра транзакта.

STR STORAGE 4096 - описание накопителя STR емкостью 4096 единиц.

TERMINATE 3 - удаление транзакта из системы, при этом содержимое итогового счетчика уменьшается на 3 единицы, моделирование заканчивается, если содержимое счетчика станет равным или меньше нуля.

ADVANCE A,B - задержка транзакта на время, определенное содержимым полей A и B, смысл величин, записываемых в этих подполях, такой же, как и в операторе GENERATE.

**SPLIT** 3,LLL,6 - копирование транзактов, в данном случае создаются три копии исходного транзакта, исходный транзакт направляется в следующий по порядку блок, а созданные копии - в блок с меткой LLL, при этом параметр 6 основного транзакта увеличивается на единицу, а транзактов - копий - на 2, 3, 4 соответственно.

**ASSEMBLE** 5 - объединение транзактов, первый из вошедших в блок транзактов продолжит движение в системе после того, как в блок придут еще четыре транзакта.

**ASSIGN** 2,NAP - изменение параметров транзактов, в данном случае второй параметр транзакта получит значение NAP.

**ASSIGN** 3+,V4 - изменится значение третьего параметра транзакта - к нему прибавится значение V4.

**TRANSFER** ,MET - безусловная передача управления оператору с меткой (номером) MET.

**TRANSFER** BOTH,LAB1,UNN - переход к оператору с меткой LAB1, если он невозможен, то к оператору с меткой UNN, если и он невозможен, то транзакт задерживается до следующего момента дискретного модельного времени, в который повторяются указанные попытки перехода.

**TRANSFER** .4,AAA,LAB - транзакт с вероятностью 0.4 переходит к оператору с меткой LAB и с вероятностью 0.6 к оператору с меткой AAA.

**TRANSFER** PICK,STK7,STK21 - равновероятный переход к операторам с номерами STK7, STK7+1, STK7+2, . . . , STK21.

**TRANSFER** FN,AAA,5 - переход к оператору, метка которого равна сумме значения функции AAA и числа 5.

**TRANSFER** P,4,41 - переход к оператору, метка которого равна сумме значения параметра N 4 транзакта и числа 41.

**TRANSFER** SBR,PRC,7 - переход к оператору PRC с записью в параметр N 7 транзакта метки данного оператора.

**LOOP** 6,MET - организация цикла - переход 5 раз к оператору с меткой MET и на шестой раз - к следующему по порядку оператору.

**TEST** E V7,K256,LAB - переход по условию (условная передача управления): в позициях 13-18 записывается знак отношения, в первых двух подполях поля переменных записываются сравниваемые величины, если



условие выполняется, то перехода нет, иначе переход есть к оператору с меткой LAB. Символы отношений: G - больше, L - меньше, E - равно, NE - не равно, LE - меньше или равно, GE - больше или равно. В данном примере перехода нет, если  $V7 = 256$ , иначе переход к оператору с номером LAB.

QUEUE SQV - оператор организации очереди, длина очереди SQV увеличивается на единицу.

DEPART SQV - то же, но длина очереди уменьшается на единицу.

PRIORITY 2 - транзакту присваивается приоритет 2.

SIMULATE - начальная карта программы, если разработчик намерен выполнить прогон модели. Если эта карта отсутствует, то интерпретатор проверяет правильность записи модели на языке GPSS, но прогона модели не выполняет.

START 100,,25 - занесение значения 100 в итоговый счетчик, вывод накопленных статистических данных производится с интервалом изменения содержимого итогового счетчика в 25 единиц.

TABULATE MAT7 - в соответствующий интервал гистограммы с именем MAT добавляется единица.

MAT7 TABLE P3,8,1,5 - описание таблицы (гистграммы) MAT7, предназначенной для табулирования величины, значения которой находятся в третьем параметре транзакта, входящего в оператор TABULATE MAT7, верхний предел левого интервала гистограммы равен 8, ширина каждого последующего интервала равна 1, всего интервалов 5.

5 VARIABLE X2 - K25 - вычислительный оператор, в данном случае из хранимой величины N 2 вычитается число 25 и результат присваивается переменной N 5.

SAVEVALUE 5,\*3 - хранимая величина N 5 (X5) получает значение третьего параметра транзакта.

Основные команды интерпретатора GPSSPC

@<имя файла> - загрузить исходный текст модели

SAVE <имя файла> - сохранить текст модели

DISPLAY[<строка N 1>],[< строка N 2>] - вывести на экран текст модели, начиная со строки N1 , до строки N 2.

DELETE [<строка N 1>],[<строка N 2>] - удалить из программы текст, начиная со строки N1 , до строки N 2.

EDIT <номер строки> - отредактировать строку текста

CLEAR - сбросить всю статистику модели в ноль (включая таймеры абсолютного и относительного модельного времени), вернуть все транзакты в пассивный буфер .

END - завершить работу интерпретатора.

### **3.3. Основные блоки GPSS/PC и связанные с ними объекты**

#### **3.3.1. Блоки, связанные с транзактами**

С транзактами связаны блоки создания, уничтожения, задержки транзактов, изменения их атрибутов и создания копий транзактов.

Для создания транзактов, входящих в модель, служит блок GENERATE (генерировать), имеющий следующий формат:

*имя* GENERATE A,B,C,D,E

В поле A задается среднее значение интервала времени между моментами поступления в модель двух последовательных транзактов. Если этот интервал постоянен, то поле B не используется. Если же интервал поступления является случайной величиной, то в поле B указывается модификатор среднего значения, который может быть задан в виде модификатора-интервала или модификатора-функции.

Модификатор-интервал используется, когда интервал поступления транзактов является случайной величиной с равномерным законом распределения вероятностей. В этом случае в поле B может быть задан любой СЧА, кроме ссылки на функцию, а диапазон изменения интервала поступления имеет границы A-B, A+B.

Например, блок

## GENERATE 100,40

создает транзакты через случайные интервалы времени, равномерно распределенные на отрезке [60;140].

Модификатор-функция используется, если закон распределения интервала поступления отличен от равномерного. В этом случае в поле В должна быть записана ссылка на функцию (ее СЧА), описывающую это т закон, и случайный интервал поступления определяется, как целая часть произведения поля А (среднего значения) на вычисленное значение функции.

В поле С задается момент поступления в модель первого транзакта. Если это поле пусто или равно 0, то момент появления первого транзакта определяется операндами А и В.

Поле D задает общее число транзактов, которое должно быть создано блоком GENERATE. Если это поле пусто, то блок генерирует неограниченное число транзактов до завершения моделирования.

В поле Е задается *приоритет*, присваиваемый генерируемым транзактам. Число уровней приоритетов неограничено, причем самый низкий приоритет — нулевой. Если поле Е пусто, то генерируемые транзакты имеют нулевой приоритет.< /P>

Транзакты имеют ряд стандартных числовых атрибутов. Например, СЧА с названием PR позволяет ссылаться на приоритет транзакта. СЧА с названием M1 содержит так называемое *резидентное время* транзакта, т.е. время, прошедшее с момента входа транзакта в модель через блок GENERATE. СЧА с названием XN1 содержит внутренний *номер транзакта*, который является уникальным и позволяет всегда отличить один транзакт от другого. В отличие от СЧА других объектов, СЧА транзактов не содержат ссылки на имя или номер транзакта. Ссылка на СЧА транзакта всегда относится к активному транзакту, т.е. транзакту, обрабатываемому в данный момент симулятором.

Важными стандартными числовыми атрибутами транзактов являются значения их параметров. Любой транзакт может иметь неограниченное число параметров, содержащих те или иные числовые значения. Ссылка на этот СЧА транзактов всегда относится к активному транзакту и имеет вид Pj или P\$имя, где j и имя - номер и имя параметра соответственно. Такая ссылка возможна только в том случае, если параметр с указанным номером или именем существует, т.е. в него занесено какое-либо значение.

Для присваивания параметрам начальных значений или изменения этих значений служит блок ASSIGN (присваивать), имеющий следующий формат:

имя ASSIGN A,B,C

В поле А указывается номер или имя параметра, в который заносится значение операнда В. Если в поле А после имени (номера) параметра стоит знак + или - , то значение операнда В добавляется или вычитается из текущего содержимого параметра. В поле С может быть указано имя или номер функции-модификатора, действующей аналогично функции-модификатору в поле В блока GENERATE.

Например, блок

ASSIGN 5,0

записывает в параметр с номером 5 значение 0, а блок

ASSIGN COUNT+,1

добавляет 1 к текущему значению параметра с именем COUNT.

Для записи текущего модельного времени в заданный параметр транзакта служит блок MARK (отметить), имеющий следующий формат:

*имя* MARK А

В поле А указывается номер или имя параметра транзакта, в который заносится текущее модельное время при входе этого транзакта в блок MARK. Содержимое этого параметра может быть позднее использовано для определения *транзитного времени* пребывания транзакта в какой-то части модели с помощью СЧА с названием MP.

Например, если на входе участка модели поместить блок

MARK MARKER

то на выходе этого участка СЧА MP\$MARKER будет содержать разность между текущим модельным временем и временем, занесенным в параметр MARKER блоком MARK.

Если поле А блока MARK пусто, то текущее время заносится на место отметки времени входа транзакта в модель, используемой при определении резидентного времени транзакта с помощью СЧА M1.

Для изменения приоритета транзакта служит блок PRIORITY (приоритет), имеющий следующий формат:

*имя* PRIORITY A,B

В поле А записывается новый приоритет транзакта. В поле В может содержаться ключевое слово BU, при наличии которого транзакт, вошедший в блок, помещается в списке текущих событий после всех остальных транзактов новой приоритетной группы, и список текущих событий просматривается с начала. Использование такой возможности будет рассмотрено ниже.

Для удаления транзактов из модели служит блок TERMINATE (завершить), имеющий следующий формат:

*имя* TERMINATE A

Значение поля А указывает, на сколько единиц уменьшается содержимое так называемого счетчика завершений при входе транзакта в данный блок TERMINATE. Если поле А не определено, то оно считается равным 0, и транзакты, проходящие через такой блок, не уменьшают содержимого счетчика завершений.

Начальное значение счетчика завершений устанавливается управляющим оператором START (начать), предназначенным для запуска прогона модели. Поле А этого оператора содержит начальное значение счетчика завершений (см. разд. 3). Прогон модели заканчивается, когда содержимое счетчика завершений обращается в 0. Таким образом, в модели должен быть хотя бы один блок TERMINATE с непустым полем А, иначе процесс моделирования никогда не завершится.

Текущее значение счетчика завершений доступно программисту через системный СЧА TG1.

Участок блок-схемы модели, связанный с парой блоков GENERATE-TERMINATE, называется сегментом. Простые модели могут состоять из одного сегмента, в сложных моделях может быть несколько сегментов.

Например, простейший сегмент модели, состоящий всего из двух блоков GENERATE и TERMINATE и приведенный на рис. 1, в совокупности с управляющим оператором START моделирует процесс создания случайного потока транзактов, поступающих в модель со средним интервалом в 100 единиц модельного времени, и уничтожения этих транзактов. Начальное значение счетчика завершений равно 1000. Каждый транзакт, проходящий через блок TERMINATE, вычитает из счетчика единицу, и таким образом моделирование завершится, когда тысячный по счету транзакт войдет в блок TERMINATE. При этом точное значение

таймера в момент завершения прогона непредсказуемо. Следовательно, в приведенном примере продолжительность прогона устанавливается не по модельному времени, а по количеству транзактов, прошедших через модель.

GENERATE	100,40
TERMINATE	1
START	1000

Рис. 1

Если необходимо управлять продолжительностью прогона по модельному времени, то в модели используется специальный сегмент, называемый сегментом таймера.

GENERATE	100,40
TERMINATE	
GENERATE	100000
TERMINATE	1
START	1

Рис. 2

Например, в модели из двух сегментов, приведенной на рис. 2, первый (основной) сегмент выполняет те же функции, что и в предыдущем примере. Заметим, однако, что поле А блока TERMINATE в первом сегменте пусто, т.е. уничтожаемые транзакты не уменьшают содержимого счетчика завершений. Во втором сегменте блок GENERATE создаст первый транзакт в момент модельного времени, равный 100000. Но этот транзакт окажется и последним в данном сегменте, так как, войдя в блок TERMINATE, он обратит в 0 содержимое счетчика завершений, установленное оператором START равным 1. Таким образом, в этой модели гарантируется завершение прогона в определенный момент модельного времени, а точное количество транзактов, прошедших через модель, непредсказуемо.

В приведенных примерах транзакты, входящие в модель через блок GENERATE, в тот же момент модельного времени уничтожались в блоке TERMINATE. В моделях систем массового обслуживания заявки обслуживаются приборами (каналами) СМО в течение некоторого промежутка времени прежде, чем покинуть СМО. Для моделирования такого обслуживания, т.е. для задержки транзактов на определенный отрезок модельного времени, служит блок ADVANCE (задержать), имеющий следующий формат:

*имя* ADVANCE A,B

Операнды в полях А и В имеют тот же смысл, что и в соответствующих полях блока GENERATE. Следует отметить, что транзакты, входящие в блок ADVANCE, переводятся из списка текущих событий в список будущих событий, а по истечении вычисленного времени задержки возвращаются назад, в список текущих событий, и их продвижение по блок-схеме продолжается. Если вычисленное время задержки равно 0, то транзакт в тот же момент модельного времени переходит в следующий блок, оставаясь в списке текущих событий.

Например, в сегменте, приведенном на рис. 3, транзакты, поступающие в модель из блока GENERATE через случайные интервалы времени, имеющие равномерное распределение на отрезке [60;140], попадают в блок ADVANCE. Здесь определяется случайное время задержки транзакта, имеющее равномерное распределение на отрезке [30;130], и транзакт переводится в список будущих событий. По истечении времени задержки транзакт возвращается в список текущих событий и входит в блок TERMINATE, где уничтожается. Заметим, что в списке будущих событий, а значит и в блоке ADVANCE может одновременно находиться произвольное количество транзактов.

GENERATE	100,40
ADVANCE	80,50
TERMINATE	1

Рис. 3

В рассмотренных выше примерах случайные интервалы времени подчинялись равномерному закону распределения вероятностей. Для получения случайных величин с другими распределениями в GPSS/PC используются вычислительные объекты: переменные и функции.

Как известно, произвольная случайная величина связана со случайной величиной R, имеющей равномерное распределение на отрезке [0;1], через свою обратную функцию распределения. Для некоторых случайных величин уравнение связи имеет явное решение, и значение случайной величины с заданным распределением вероятностей может быть вычислено через R по формуле. Так, например, значение случайной величины E с показательным (экспоненциальным) распределением с параметром d вычисляется по формуле:

$$E = -(1/d) * \ln(R)$$

Напомним, что параметр d имеет смысл величины, обратной математическому ожиданию E, а, следовательно, 1/d —математическое ожидание (среднее значение) случайной величины E.

Для получения случайной величины  $R$  с равномерным распределением на отрезке  $[0;1]$  в GPSS/PC имеются встроенные генераторы случайных чисел. Для получения случайного числа путем обращения к такому генератору достаточно записать системный СЧА RN с номером генератора, например RN1. Правда, встроенные генераторы случайных чисел GPSS/PC дают числа не на отрезке  $[0;1]$ , а целые случайные числа, равномерно распределенные от 0 до 999, но их нетрудно привести к указанному отрезку делением на 1000.

Проще всего описанные вычисления в GPSS/PC выполняются с использованием *арифметических переменных*. Они могут быть целыми и действительными. *Целые переменные* определяются перед началом моделирования с помощью оператора определения VARIABLE (переменная), имеющего следующий формат:

*имя VARIABLE выражение*

Здесь *имя* - имя переменной, используемое для ссылок на нее, а *выражение* - арифметическое выражение, определяющее переменную. Арифметическое выражение представляет собой комбинацию операндов, в качестве которых могут выступать константы, СЧА и функции, знаков арифметических операций и круглых скобок. Следует заметить, что знаком операции умножения в GPSS/PC является символ # (номер). Результат каждой промежуточной операции в целых переменных преобразуется к целому типу путем отбрасывания дробной части, и, таким образом, результатом операции деления является целая часть частного.

*Действительные переменные* определяются перед началом моделирования с помощью оператора определения FVARIABLE, имеющего тот же формат, что и оператор VARIABLE. Отличие действительных переменных от целых заключается в том, что в действительных переменных все промежуточные операции выполняются с сохранением дробной части чисел, и лишь окончательный результат приводится к целому типу отбрасыванием дробной части.

Арифметические переменные обоих типов имеют единственный СЧА с названием V, значением которого является результат вычисления арифметического выражения, определяющего переменную. Вычисление выражения производится при входе транзакта в блок, содержащий ссылку на СЧА V с именем переменной.

Действительные переменные могут быть использованы для получения случайных интервалов времени с показательным законом распределения. Пусть в модели из примера на рис. 3 распределения времени поступления транзактов и времени задержки должны иметь показательный закон. Это может быть сделано так, как показано на рис. 4.



```

TARR      FVARIABLE -100#LOG((1+RN1)/1000)
TSRV      FVARIABLE -80#LOG((1+RN1)/1000)
GENERATE   V$TARR
ADVANCE    V$TSRV
TERMINATE  1

```

Рис. 4

Переменная с именем TARR задает выражение для вычисления интервала поступления со средним значением 100, вторая переменная с именем TSRV — для вычисления времени задержки со средним значением 80. Блоки GENERATE и ADVANCE содержат в поле A ссылки на соответствующие переменные, при этом поле B не используется, так как в поле A содержится случайная величина, не нуждающаяся в модификации. Большинство случайных величин не может быть получено через случайную величину R с помощью арифметического выражения. Кроме того, такой способ является достаточно трудоемким, так как требует обращения к математическим функциям, вычисление которых требует десятков машинных операций. Другим возможным способом является использование вычислительных объектов GPSS/PC типа *функция*.

Функции используются для вычисления величин, заданных табличными зависимостями. Каждая функция определяется перед началом моделирования с помощью оператора определения FUNCTION (функция), имеющего следующий формат:

*имя* FUNCTION A,B

Здесь *имя* — имя функции, используемое для ссылок на нее; A — стандартный числовой атрибут, являющийся аргументом функции; B — тип функции и число точек таблицы, определяющей функцию.

Существует пять типов функций. Рассмотрим вначале *непрерывные числовые функции*, тип которых кодируется буквой C. Так, например, в определении непрерывной числовой функции, таблица которой содержит 24 точки, поле B должно иметь значение C24.

При использовании непрерывной функции для генерирования случайных чисел ее аргументом должен быть один из генераторов случайных чисел RNj. Так, оператор для определения функции показательного распределения может иметь следующий вид:

EXP FUNCTION RN1,C24

Особенностью использования встроенных генераторов случайных чисел RNj в качестве аргументов функций является то, что их значения в этом контексте интерпретируются как дробные числа от 0 до 0,999999.

Таблица с координатами точек функции располагается в строках, следующих непосредственно за оператором FUNCTION. Эти строки не должны иметь поля нумерации. Каждая точка таблицы задается парой Xi (значение аргумента) и Yi (значение функции), отделяемых друг от друга запятой. Пары координат отделяются друг от друга символом “/” и располагаются на произвольном количестве строк. Последовательность значений аргумента Xi должна быть строго возрастающей.

Как уже отмечалось, при использовании функции в поле В блоков GENERATE и ADVANCE вычисление интервала поступления или времени задержки производится путем умножения операнда А на вычисленное значение функции. Отсюда следует, что функция, используемая для генерирования случайных чисел с показательным распределением, должна описывать зависимость  $y = -\ln(x)$ , представленную в табличном виде. Оператор FUNCTION с такой таблицей, содержащей 24 точки для обеспечения достаточной точности аппроксимации, имеет следующий вид:

```
EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
```

Вычисление непрерывной функции производится следующим образом. Сначала определяется интервал (Xi;Xi+1), на котором находится текущее значение СЧА-аргумента (в нашем примере — сгенерированное значение RN1). Затем на этом интервале выполняется линейная интерполяция с использованием соответствующих значений Yi и Yi+1. Результат интерполяции усекается (отбрасыванием дробной части) и используется в качестве значения функции. Если функция служит операндом В блоков GENERATE или ADVANCE, то усечение результата производится только после его умножения на значение операнда А.

Использование функций для получения случайных чисел с заданным распределением дает хотя и менее точный результат за счет погрешностей аппроксимации, но зато с меньшими вычислительными затратами (несколько машинных операций на выполнение линейной интерполяции). Чтобы к погрешности аппроксимации не добавлять слишком большую погрешность усечения, среднее значение при использовании показательных распределений должно быть достаточно большим (не менее 50). Эта рекомендация относится и к использованию переменных.

Функции всех типов имеют единственный СЧА с названием FN, значением которого является вычисленное значение функции. Вычисление функции производится при входе транзакта в блок, содержащий ссылку на СЧА FN с именем функции.

Заменим в примере на рис. 4 переменные TARR и TSRV на функцию EXP (рис. 5).

Поскольку в обеих моделях используется один и тот же генератор RN1, интервалы поступления и задержки, вычисляемые в блоках GENERATE и ADVANCE, должны получиться весьма близкими, а может быть и идентичными. При большом количестве транзактов, пропускаемых через модель (десятки и сотни тысяч), разница в скорости вычислений должна стать заметной.

```
EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
ADVANCE              80,FN$EXP
TERMINATE      1
Рис. 5
```

Особенностью непрерывных функций является то, что они принимают “непрерывные” (но только целочисленные) значения в диапазоне от  $Y_1$  до  $Y_n$ , где  $n$  — количество точек таблицы. В отличие от них *дискретные числовые функции*, тип которых кодируется буквой D в операнде B оператора определения функции, принимают только отдельные (дискретные) значения, заданные координатами  $Y_i$  в строках, следующих за оператором определения FUNCTION. При вычислении дискретной функции текущее значение СЧА-аргумента, указанного в поле A оператора FUNCTION, сравнивается по условию  $\leq$  последовательно со всеми значениями упорядоченных по возрастанию координат  $X_i$  до выполнения

этого условия при некотором  $i$ . Значением функции становится целая часть соответствующего значения  $Y_i$ .

Если последовательность значений аргумента таблицы с координатами точек функции представляет числа натурального ряда  $(1, 2, 3, \dots, n)$ , то такую дискретную функцию с целью экономии памяти и машинного времени удобно определить как *списковую числовую функцию* (тип L).

Пусть в модели на рис. 5 заявки, моделируемые транзактами, с равной вероятностью  $1/3$  должны относиться к одному из трех классов (типов) 1, 2 и 3, а среднее время задержки обслуживания заявок каждого типа должно составлять соответственно 70, 80 и 90 единиц модельного времени. Это может быть обеспечено способом, показанным на рис. 6.

В блоке ASSIGN в параметр TYPE каждого сгенерированного транзакта заносится тип заявки, получаемый с помощью дискретной функции CLASS. Аргументом функции является генератор случайных чисел RN1, а координаты ее таблицы представляют собой обратную функцию распределения дискретной случайной величины “класс заявки” с одинаковыми вероятностями каждого из трех значений случайной величины.

Поле A блока ADVANCE содержит ссылку на списковую функцию MEAN, аргументом которой служит параметр TYPE входящих в блок транзактов. В зависимости от значений этого параметра (типа заявки) среднее время задержки принимает одно из трех возможных значений функции MEAN: 70, 80 или 90 единиц.

```
EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
CLASS    FUNCTION      RN1,D3
.333,1/.667,2/1,3
MEAN     FUNCTION      P$TYPE,L3
1,70/2,80/3,90
GENERATE  100,FN$EXP
ASSIGN    TYPE,FN$CLASS
ADVANCE   FN$MEAN,FN$EXP
TERMINATE 1
```

Рис. 6

Следует отметить, что в данном примере можно было бы не использовать параметр TYPE и обойтись одной дискретной функцией, возвращающей с равной вероятностью одно из трех возможных значений среднего времени задержки. Однако использование параметров дает некоторые дополнительные возможности, которые будут рассмотрены позже.

Транзакты могут входить в модель не только через блок GENERATE, но и путем создания копий уже существующих транзактов в блоке SPLIT (расщепить), имеющем следующий формат:

*имя* SPLIT A,B,C

В поле A задается число создаваемых копий исходного транзакта (родителя), входящего в блок SPLIT. После выхода из блока SPLIT транзакт-родитель направляется в следующий блок, а все транзакты-потомки поступают в блок, указанный в поле B. Если поле B пусто, то все копии поступают в следующий блок.

Транзакт-родитель и его потомки, выходящие из блока SPLIT, могут быть пронумерованы в параметре, имя или номер которого указаны в поле C. Если у транзакта-родителя значение этого параметра при входе в блок SPLIT было равно k, то при выходе из блока оно станет равным k+1, а значения этого параметра у транзактов-потомков окажутся равными k+2, k+3 и т.д.

Например, блок

SPLIT      5,MET1,NUM

создает пять копий исходного транзакта и направляет их в блок с именем MET1. Транзакт-родитель и потомки нумеруются в параметре с именем NUM. Если, например, перед входом в блок значение этого параметра у транзакта-родителя было равно 0, то при выходе из блока оно станет равным 1, а у транзактов-потомков значения параметра NUM будут равны 2, 3, 4, 5 и 6.

### 3.3.2.      Блоки, связанные с аппаратными объектами

Все примеры моделей, рассматривавшиеся выше, пока еще не являются моделями систем массового обслуживания, так как в них не учтена основная особенность СМО: конкуренция заявок на использование некоторых ограниченных ресурсов системы. Все транзакты, входящие в эти модели через блок GENERATE, немедленно получают возможность “обслуживания” в блоке ADVANCE, который никогда не “отказывает” транзактам во входе, сколько бы транзактов в нем не находилось.

Для моделирования ограниченных ресурсов СМО в модели должны присутствовать аппаратные объекты: одноканальные или многоканальные устройства. *Одноканальные устройства* создаются в текущей модели при использовании блоков SEIZE (занять) и RELEASE (освободить), имеющих следующий формат:

*имя* SEIZE    A

*имя* RELEASE A

В поле A указывается номер или имя устройства. Если транзакт входит в блок SEIZE, то устройство, указанное в поле A, становится занятым и остаётся в этом состоянии до тех пор, пока этот же транзакт не пройдёт соответствующий блок RELEASE, освобождая устройство. Если устройство, указанное в поле A блока SEIZE, уже занято каким-либо транзактом, то

никакой другой транзакт не может войти в этот блок и остаётся в предыдущем блоке. Транзакты, задержанные (заблокированные) перед блоком SEIZE, остаются в списке текущих событий и при освобождении устройства обрабатываются с учетом приоритетов и очередности поступления.

Каждое устройство имеет следующие СЧА: F —состояние устройства (0 —свободно,1 —занято); FR —коэффициент использования в долях 1000; FC —число занятий устройства; FT —целая часть среднего времени занятия устройства.

Воспользуемся блоками SEIZE и RELEASE для моделирования одноканальной СМО с ожиданием (рис. 7). Теперь блок ADVANCE находится между блоками SEIZE и RELEASE, моделирующими занятие и освобождение устройства с именем SYSTEM, и поэтому в нем может находиться только один транзакт. Транзакты, выходящие из блока GENERATE в моменты занятости устройства, не смогут войти в блок SEIZE и будут оставаться в блоке GENERATE, образуя очередь в списке текущих событий.

EXP	FUNCTION	RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915		
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3		
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9		
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8		
GENERATE	100,FN\$EXP	
SEIZE	SYSTEM	
ADVANCE	80,FN\$EXP	
RELEASE	SYSTEM	
TERMINATE	1	

Рис. 7

Для моделирования *захвата (прерывания)* одноканального устройства вместо блоков SEIZE и RELEASE используются соответственно блоки PREEMPT (захватить) и RETURN (вернуть). Блок PREEMPT имеет следующий формат:

*имя* PREEMPT A,B,C,D,E

В поле А указывается имя или номер устройства, подлежащего захвату. В поле В кодируется условие захвата. Если это поле пусто, то захват возникает, если обслуживаемый транзакт сам не является захватчиком. Если же в поле В записан операнд PR, то захват возникает, если приоритет транзакта-захватчика выше, чем приоритет обслуживаемого транзакта.

Поля С, D и E определяют поведение транзактов, обслуживание которых было прервано. Поле С указывает имя блока, в который будет направлен прерванный транзакт. В поле D может быть указан номер или имя параметра прерванного транзакта, в который записывается время, оставшееся этому транзакту до завершения обслуживания на устройстве. При отсутствии операнда в поле E прерванный транзакт сохраняет право на автоматическое восстановление на устройстве по окончании захвата. Если же в поле E указан операнд RE, то транзакт теряет такое право.

Блок RETURN имеет единственный операнд А, содержащий имя или номер устройства, подлежащего освобождению от захвата.

Блоки PREEMPT и RETURN могут быть использованы для моделирования СМО с абсолютными приоритетами. В простейших случаях, при одном уровне захвата, в блоке PREEMPT используется единственный операнд А. При этом прерванный транзакт переводится симулятором из списка будущих событий в так называемый *список прерываний* устройства, а по окончании захвата устройства возвращается в список будущих событий с предварительно вычисленным временем занятия устройства для продолжения обслуживания.

Для создания в модели *многоканальных устройств (МКУ)* они должны быть предварительно определены с помощью операторов определения STORAGE (память), имеющих следующий формат:

*имя* STORAGE A

Здесь *имя* — имя МКУ, используемое для ссылок на него; А —емкость (количество каналов обслуживания) МКУ, задаваемая константой.

Для занятия и освобождения каналов обслуживания МКУ используется пара блоков ENTER (войти) и LEAVE (покинуть), имеющих следующий формат:

*имя* ENTER A,B

имя LEAVE A,B

В поле А указывается номер или имя МКУ, в поле В - число каналов МКУ, занимаемых при входе в блок ENTER или освобождаемых при входе в блок LEAVE. Обычно поле В пусто, и в этом случае по умолчанию занимается или освобождается один канал.

При входе транзакта в блок ENTER текущее содержимое МКУ увеличивается на число единиц, указанное в поле В. Если свободная емкость МКУ меньше значения поля В, то транзакт не может войти в блок ENTER и остается в предыдущем блоке, образуя очередь в списке текущих событий.

При входе транзакта в блок LEAVE текущее содержимое МКУ уменьшается на число единиц, указанное в поле В. Не обязательно освобождается такое же число каналов МКУ, какое занималось при входе данного транзакта в блок ENTER, однако текущее содержимое МКУ не должно становиться отрицательным.

Многоканальные устройства имеют следующие СЧА: S —текущее содержимое МКУ; R —свободная емкость МКУ; SR —коэффициент использования в долях 1000; SA —целая часть среднего содержимого МКУ; SM —максимальное содержимое МКУ; SC —число занятий МКУ; ST —целая часть среднего времени занятия МКУ.

Воспользуемся блоками ENTER-LEAVE и оператором STORAGE для моделирования двухканальной СМО с ожиданием (рис. 8). Если текущее содержимое МКУ с именем STO2 меньше 2, т.е. в блоке ADVANCE находится один или ни одного транзакта, то очередной транзакт, поступающий в модель через блок GENERATE, может войти в блок ENTER и затем в блок ADVANCE. Если же текущее содержимое МКУ равно 2, то очередной транзакт остается в блоке GENERATE, образуя очередь в списке текущих событий. По истечении задержки одного из двух обслуживаемых транзактов в блоке ADVANCE и после входа его в блок LEAVE первый из заблокированных транзактов сможет войти в блок ENTER.

STO2        STORAGE 2

EXP        FUNCTION        RN1,C24

0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915

.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3

.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9

.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8



GENERATE	100, FN\$EXP
ENTER	STO2
ADVANCE	160, FN\$EXP
LEAVE	STO2
TERMINATE	1

Рис. 8

К аппаратным объектам относятся также *логические переключатели (ЛП)*, которые могут находиться в двух состояниях: “включено” и “выключено”. В начале моделирования все ЛП находятся в состоянии “выключено”. Отдельные переключатели могут быть установлены в начальное состояние “включено” с помощью оператора INITIAL (инициализировать), имеющего следующий формат:

INITIAL LS\$ *имя*

INITIAL LS*j*

Здесь *имя* и *j* - соответственно имя и номер ЛП, устанавливаемого в начальное состояние “включено”.

Для включения, выключения и инвертирования логических переключателей в процессе моделирования служит блок LOGIC (установить ЛП), имеющий следующий формат:

*имя* LOGIC X A

В поле A указывается имя или номер ЛП. Вспомогательный операнд X указывает вид операции, которая производится с логическим переключателем при входе транзакта в блок: S — включение, R — выключение, I — инвертирование. Например:

LOGIC S 9

## LOGIC R FLAG

Логические переключатели имеют единственный СЧА с названием LS. Значение СЧА равно 1, если ЛП включен, и 0, если он выключен.

### 3.3.3. Блоки для сбора статистических данных

Два последних примера в предыдущем параграфе представляют собой законченные модели одноканальной и многоканальной СМО с ожиданием. Однако такие модели разрабатываются обычно для исследования различных характеристик, связанных с ожиданием заявок в очереди: длины очереди, времени ожидания и т.п., а в приведенных примерах очередь транзактов образуется в списке текущих событий и недоступна исследователю. Для регистрации статистической информации о процессе ожидания транзактов в модели должны присутствовать статистические объекты: очереди или таблицы.

Объекты типа *очередь* создаются в модели путем использования блоков — регистраторов очередей: QUEUE (стать в очередь) и DEPART (уйти из очереди), имеющих следующий формат:

*имя* QUEUE A,B

*имя* DEPART A,B

В поле A указывается номер или имя очереди, а в поле B — число единиц, на которое текущая длина очереди увеличивается при входе транзакта в блок QUEUE или уменьшается при входе транзакта в блок DEPART. Обычно поле B пусто, и в этом случае его значение по умолчанию принимается равным 1.

Для сбора статистики о транзактах, заблокированных перед каким-либо блоком модели, блоки QUEUE и DEPART помещаются перед и после этого блока соответственно. При прохождении транзактов через блоки QUEUE и DEPART соответствующим образом изменяются следующие СЧА очередей: Q — текущая длина очереди; QM — максимальная длина очереди; QA — целая часть средней длины очереди; QC — общее число транзактов, вошедших в очередь; QZ — число транзактов, прошедших через очередь без ожидания (число “нулевых” входов); QT — целая часть среднего времени ожидания с учетом “нулевых” входов; QX — целая часть среднего времени ожидания без учета “нулевых” входов.

Дополним приведенную на рис. 7 модель одноканальной СМО блоками QUEUE и DEPART (рис. 9). Теперь транзакты, заблокированные перед блоком SEIZE из-за занятости устройства SYSTEM, находятся в блоке QUEUE, внося свой вклад в статистику о времени ожидания, накапливаемую в статистическом объекте типа “очередь” с именем LINE. При освобождении устройства первый из заблокированных транзактов войдет в блок SEIZE и

одновременно в блок DEPART, прекращая накопление статистики об ожидании этого транзакта.

```
EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
QUEUE LINE
SEIZE         SYSTEM
DEPART       LINE
ADVANCE              80,FN$EXP
RELEASE      SYSTEM
TERMINATE     1
Рис. 9
```

Очень часто исследователя интересует не только среднее значение времени ожидания в очереди, но и дисперсия этого времени, а также статистическое распределение выборки времени ожидания, представляемое обычно графически в виде гистограммы. Имея такое распределение, можно оценить вероятность того, что время ожидания превысит или не превысит некоторое заданное значение. Для сбора и обработки данных о выборочном распределении времени ожидания в очереди служат статистические объекты типа *Q-таблица*.

Для создания в модели такой таблицы она должна быть предварительно определена с помощью оператора определения QTABLE (*Q-таблица*), имеющего следующий формат:

*имя* QTABLE A,B,C,D

Здесь *имя* - имя таблицы, используемое для ссылок на нее; A - номер или имя очереди, распределение времени ожидания в которой необходимо получить; B - верхняя граница первого частотного интервала таблицы; C — ширина частотных интервалов; D - количество частотных интервалов.

Диапазон всевозможных значений времени ожидания в очереди, указанной в поле A, разбивается на ряд частотных интервалов, количество которых указано в поле D. Первый из этих интервалов имеет ширину от минус бесконечности до величины, указанной в поле B, включительно. Второй интервал включает значения, большие, чем величина первой границы в поле B, но меньшие или равные B+C, и т.д. Все промежуточные интервалы имеют одинаковую ширину, указанную в поле C. Наконец, последний интервал включает все значения, большие, чем последняя граница. Значения

операндов В, С и D должны задаваться целыми константами. Операнд В может быть неположительным, хотя для Q-таблицы это не имеет смысла, так как время не может быть отрицательным. Операнды С и D должны быть строго положительными.

При прохождении транзакта через блоки QUEUE и DEPART его время ожидания фиксируется, и к счетчику частотного интервала таблицы, в который попало это время, добавляется 1. Одновременно в таблице накапливается информация для вычисления среднего значения и среднеквадратического отклонения (корня из дисперсии) времени ожидания. По окончании моделирования среднее значение и среднеквадратическое отклонение времени ожидания, а также счетчики попаданий в различные частотные интервалы выводятся в стандартный отчет GPSS/PC.

Таблицы, как и другие объекты GPSS/PC, имеют СЧА: TC — общее число транзактов, вошедших в очередь, связанную с таблицей; TB — целая часть среднего времени ожидания в очереди; TD — целая часть среднеквадратического отклонения времени ожидания в очереди.

Дополним модель из примера на рис. 9 оператором QTABLE для получения распределения времени ожидания в очереди с именем LINE (рис. 10).

```

WTIME    QTABLE        LINE,50,50,10
EXP       FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE   100,FN$EXP
QUEUE      LINE
SEIZE      SYSTEM
DEPART     LINE
ADVANCE    80,FN$EXP
RELEASE    SYSTEM
TERMINATE  1
Рис. 10

```

Оператор определения таблицы с именем WTIME разбивает ось времени на 10 частотных интервалов. Первый интервал включает значения от 0 до 50, второй — от 50 до 100, третий — от 100 до 150 и т.д. Последний, десятый, интервал включает значения, превышающие 450. Если, например, время ожидания некоторого транзакта в очереди составило 145 единиц модельного времени, то к счетчику третьего частотного интервала будет добавлена 1. Следует заметить, что информация в таблицу с именем WTIME заносится автоматически, при входе транзактов в блоки QUEUE и DEPART, и никаких специальных мер для этого принимать не требуется.

Таблицы в GPSS/PC могут использоваться в более общем случае не только для табулирования времени ожидания в очереди, но и для получения

выборочных распределений произвольных СЧА любых объектов модели. Для определения таблиц служит оператор TABLE (таблица), формат которого совпадает с форматом оператора QTABLE. Отличие состоит лишь в том, что в поле А оператора TABLE записывается стандартный числовой атрибут, выборочное распределение которого необходимо получить, а операнды В, С и D определяют разбиение на частотные интервалы диапазона всевозможных значений этого СЧА.

Занесение информации в таблицу, определяемую оператором TABLE, уже не может быть выполнено симулятором автоматически, как в случае Q-таблиц. Для этого используется специальный блок TABULATE (табулировать), имеющий следующий формат:

*имя* TABULATE А

В поле А указывается номер или имя таблицы, определенной соответствующим оператором TABLE.

При входе транзакта в блок TABULATE текущее значение табулируемого аргумента таблицы, указанного в поле А оператора TABLE, заносится в нее в соответствии с заданным в операторе TABLE разбиением области значений аргумента на частотные интервалы. Одновременно корректируются текущие значения СЧА таблицы: счетчик входов в таблицу ТС, среднее время ожидания ТВ и среднеквадратическое отклонение времени ожидания TD.

Пусть, например, в модели многоканальной СМО, приведенной на рис. 8, надо получить распределение времени пребывания заявок в системе, включающего время ожидания в очереди и время обслуживания. Это может быть обеспечено способом, показанным на рис. 11.

Оператор TABLE определяет таблицу с именем TTIME, аргументом которой служит СЧА M1 — время пребывания транзакта в модели. В рассматриваемой модели значение СЧА M1 одновременно будет являться временем пребывания транзакта в СМО в том случае, если занесение информации в таблицу производить перед выходом транзакта из модели. Поэтому блок TABULATE, заносащий информацию о времени пребывания каждого транзакта в модели в таблицу TTIME, располагается перед блоком TERMINATE. Диапазон возможных значений времени пребывания транзакта в модели разбит в операторе TABLE на 12 частотных интервалов, ширина которых (кроме последнего) равна 100 единицам модельного времени.

TTIME	TABLE	M1,100,100,12
STO2	STORAGE	2
EXP	FUNCTION	RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915		

```
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
ENTER         STO2
ADVANCE       160,FN$EXP
LEAVE         STO2
TABULATE      TTIME
TERMINATE     1
Рис. 11
```

### 3.3.4. Блоки, изменяющие маршруты транзактов

В приведенных выше примерах транзакты, выходящие из любого блока, всегда поступали в следующий блок. В более сложных моделях возникает необходимость направления транзактов к другим блокам в зависимости от некоторых условий. Эту возможность обеспечивают блоки изменения маршрутов транзактов.

Блок TRANSFER (передать) служит для передачи входящих в него транзактов в блоки, отличные от следующего. Блок имеет девять режимов работы, из которых рассмотрим здесь лишь три наиболее часто используемых. В этих трех режимах блок имеет следующий формат:

*имя* TRANSFER A,B,C

Смысл операндов в полях A, B и C зависит от режима работы блока.

В режиме *безусловной передачи* поля A и C пусты, а в поле B указывается имя блока, к которому безусловным образом направляется транзакт, вошедший в блок TRANSFER. Например:

TRANSFER ,FINAL

В режиме *статистической передачи* операнд A определяет вероятность, с которой транзакт направляется в блок, указанный в поле C. С вероятностью 1-A транзакт направляется в блок, указанный в поле B (в следующий, если поле B пусто).

Вероятность в поле A может быть задана непосредственно десятичной дробью, начинающейся с точки. Например, блок

TRANSFER .75,THIS,THAT

с вероятностью 0,75 направляет транзакты в блок с именем THAT, а с вероятностью 0,25 — в блок с именем THIS.

Если же поле А начинается не с десятичной точки и не содержит одного из ключевых слов —признаков других режимов работы блока, то его значение рассматривается как количество тысячных долей в вероятности передачи. Например, предыдущий блок TRANSFER можно записать также в следующем виде:

TRANSFER 750,THIS,THAT

В режиме *логической передачи* в поле А записывается ключевое слово BOTH (оба). Транзакт, поступающий в блок TRANSFER, сначала пытается войти в блок, указанный в поле В (или в следующий блок, если поле В пусто), а если это не удастся, т.е. блок В отказывает транзакту во входе, то в блок, указанный в поле С. Если и эта попытка неудачна, то транзакт задерживается в блоке TRANSFER до изменения условий в модели, делающего возможным вход в один из блоков В или С, причем при одновременно возникшей возможности предпочтение отдается блоку В. Например:

TRANSFER BOTH,MET1,MET2

Блок TEST (проверить) служит для задержки или изменения маршрутов транзактов в зависимости от соотношения двух СЧА. Он имеет следующий формат:

*имя* TEST X A,B,C

Вспомогательный операнд X содержит условие проверки соотношения между СЧА и может принимать следующие значения: L (меньше); LE (меньше или равно); E (равно); NE (не равно); GE (больше или равно); G (больше). Поле А содержит первый, а поле В —второй из сравниваемых СЧА. Если проверяемое условие А X В выполняется, то блок TEST пропускает транзакт в следующий блок. Если же это условие не выполняется,

то транзакт переходит к блоку, указанному в поле C, а если оно пусто, то задерживается перед блоком TEST.

Например, блок

```
TEST      LE    P$TIME,C1
```

не пропускает транзакты, у которых значение параметра с именем TIME больше текущего модельного времени. Блок

```
TEST L     Q$LINE,5,OUT
```

направляет транзакты в блок с именем OUT, если текущая длина очереди LINE больше либо равна 5.

Для задержки или изменения маршрута транзактов в зависимости от состояния аппаратных объектов модели служит блок GATE (впустить), имеющий следующий формат:

*имя* GATE X A,B

Вспомогательный операнд X содержит код состояния проверяемого аппаратного объекта, а в поле A указывается имя или номер этого объекта. Если проверяемый объект находится в заданном состоянии, то блок GATE пропускает транзакт к следующему у блоку. Если же заданное в блоке условие не выполняется, то транзакт переходит к блоку, указанному в поле B, а если это поле пусто, то задерживается перед блоком GATE.

Операнд X может принимать следующие значения: U (устройство занято); NU (устройство свободно); I (устройство захвачено); NI (устройство не захвачено); SE (МКУ пусто); SNE (МКУ не пусто); SF (МКУ заполнено); SNF (МКУ не заполнено); LS ( ЛП включен), LR (ЛП выключен).

Например, блок

```
GATE SNE BUF3
```



отказывает во входе транзактам, поступающим в моменты, когда в МКУ с именем BUF3 все каналы обслуживания свободны. Блок

GATE LR 4,BLOK2

направляет транзакты в блок с именем BLOK2, если в момент их поступления ЛП с номером 4 включен.

Блоки рассматриваемой группы используются при моделировании различных СМО с потерями заявок. Воспользуемся, например, блоками TRANSFER для моделирования двухканальной СМО с отказами и повторными попытками (рис. 12).

```
STO2      STORAGE 2
EXP       FUNCTION   RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE   100,FN$EXP
ENT1      TRANSFER   BOTH,,REFUS
ENTER      STO2
ADVANCE    160,FN$EXP
LEAVE      STO2
TERMINATE  1
REFUS     TRANSFER   .1,,OUT
ADVANCE    250,FN$EXP
TRANSFER   ,ENT1
OUT        TERMINATE 1
```

Рис. 12

Транзакты, поступающие в модель, попадают в блок TRANSFER с именем ENT1, работающий в логическом режиме. Если в момент поступления транзакта в МКУ STO2 хотя бы один канал свободен, то блок TRANSFER направит транзакт в следующий блок, т.е. в блок ENTER. Если же в момент поступления оба канала МКУ заняты, и поэтому блок ENTER отказывает во входе, то транзакт будет направлен в блок TRANSFER с именем REFUS, работающий в статистическом режиме. С вероятностью 0,9 транзакты из этого блока передаются в следующий блок, задерживаются в нем на случайное время и с помощью блока TRANSFER, работающего в безусловном режиме, передаются вновь на вход модели в блок с именем ENT1. С вероятностью 0,1 транзакты из блока с именем REFUS передаются в блок TERMINATE с именем OUT для уничтожения.

Следует заметить, что для уничтожения транзактов, получивших отказ в обслуживании, понадобился отдельный блок TERMINATE для фиксации в стандартном отчете количества потерянных транзактов с помощью счетчика блока с именем OUT (СЧА N\$OUT ).

Для моделирования той же СМО может быть использован также блок TEST (рис. 13). В этом варианте модели транзакт проходит в блок ENTER, если текущее число занятых каналов (СЧА S\$STO2) меньше 2.

```

STO2      STORAGE 2

EXP       FUNCTION    RN1,C24

0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915

.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3

.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9

.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8

GENERATE   100,FN$EXP

ENT1      TEST L     S$STO2,2,REFUS

ENTER     STO2

ADVANCE   160,FN$EXP

LEAVE     STO2

TERMINATE 1

REFUS     TRANSFER   .1,,OUT

ADVANCE   250,FN$EXP

```

```

TRANSFER      ,ENT1
OUT           TERMINATE      1

```

Рис. 13

При использовании блока GATE модель принимает вид, показанный на рис. 14. В этом варианте транзакт проходит в блок ENTER, если условие “МКУ STO2 не заполнено” истинно.

```

STO2          STORAGE 2
EXP           FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
ENT1          GATE SNF      STO2,REFUS
ENTER         STO2
ADVANCE              160,FN$EXP
LEAVE         STO2
TERMINATE     1
REFUS         TRANSFER      .1,,OUT
ADVANCE              250,FN$EXP
TRANSFER      ,ENT1
OUT           TERMINATE     1

```

Рис. 14

### 3.3.5. Блоки, работающие с памятью

Для хранения в памяти отдельных числовых значений и массивов таких значений используются сохраняемые величины и матрицы сохраняемых величин.

*Сохраняемые величины* могут использоваться в модели для хранения исходных данных, которые надо изменять при различных прогонах модели, промежуточных значений и результатов моделирования. В начале моделирования все сохраняемые величины устанавливаются равными 0. Для установки отличных от 0 начальных значений сохраняемых величин используется оператор INITIAL, имеющий следующий формат:

INITIAL X\$ *имя* , *значение*

INITIAL X<sub>j</sub>,*значение*

Здесь *имя* и *j* - соответственно имя и номер сохраняемой величины, а *значение* - присваиваемое ей начальное значение (константа).

Для изменения сохраняемых величин в процессе моделирования служит блок SAVEVALUE (сохранить величину), имеющий следующий формат:

*имя* SAVEVALUE A,B

В поле A указывается номер или имя сохраняемой величины, в которую записывается значение операнда B. Если в поле A после имени (номера) сохраняемой величины стоит знак + или - , то значение операнда B добавляется или вычитается из текущего содержимого сохраняемой величины. Например:

SAVEVALUE 5,Q\$LINE

SAVEVALUE NREF+,1

Сохраняемые величины имеют единственный СЧА с названием X, значением которого является текущее значение соответствующей сохраняемой величины.

Изменим пример на рис. 14 таким образом, чтобы исходные данные модели (средний интервал поступления транзактов и среднее время обслуживания) были заданы сохраняемыми величинами, а результат

моделирования (количество потерянных транзактов) фиксировался также в сохраняемой величине. Такая модель будет иметь вид, показанный на рис. 15.

*Матрицы* сохраняемых величин дают возможность упорядочить сохраняемые значения в виде матриц  $m \times n$ , где  $m$  — число строк,  $n$  — число столбцов матрицы. Каждая матрица должна быть перед началом моделирования определена с помощью оператора `MATRIX` (определить матрицу), имеющего следующий формат:

*имя* MATRIX A,B,C

Поле `A` оператора не используется и сохранено в GPSS/PC для совместимости со старыми версиями GPSS. В полях `B` и `C` указываются соответственно число строк и столбцов матрицы, задаваемые константами, причем общее число элементов, равное произведению `B` на `C`, не должно превышать 8191. Например, оператор

MTAB MATRIX ,10,2

определяет матрицу с именем `MTAB`, содержащую десять строк и два столбца.

```
INITIAL      X$TARR,100
INITIAL      X$TSRV,160
STO2         STORAGE      2
EXP          FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE     X$TARR,FN$EXP
ENT1         GATE SNF      STO2,REFUS
ENTER        STO2
ADVANCE      X$TSRV,FN$EXP
```

```

LEAVE          STO2
OUT            TERMINATE    1
REFUS         TRANSFER     .1,,COUT
ADVANCE                250,FN$EXP
TRANSFER      ,ENT1
COUT          SAVEVALUE    NREF+,1
TRANSFER      ,OUT

```

Рис. 15

В начале моделирования элементы всех определенных матриц устанавливаются равными 0. Для установки отличных от 0 начальных значений отдельных элементов матриц используется оператор INITIAL, имеющий следующий формат:

INITIAL MX\$ *имя* ( *a*,*b* ), *значение*

INITIAL MX $j(a,b)$ , *значение*

Здесь *имя* и *j* - соответственно имя и номер матрицы; *a* и *b* - номера соответственно строки и столбца, задаваемые константами; *значение* присваиваемое элементу матрицы начальное значение, задаваемое также константой.

Для изменения значений элементов матриц в процессе моделирования служит блок MSAVEVALUE (сохранить значение элемента матрицы), имеющий следующий формат:

*имя* MSAVEVALUE A,B,C,D

В поле A указывается имя или номер матрицы, после которого, как и в блоке SAVEVALUE, может стоять знак + или -. В полях B и C указываются номера соответственно строки и столбца, определяющие изменяемый элемент матрицы. В поле D указывается величина, используемая для изменения заданного элемента матрицы. Например:

MSAVEVALUE 5,3,2,X1

MSAVEVALUE MTAB+,P\$ROW,P\$COL,1

Матрицы имеют единственный СЧА с названием MX, ссылка на который записывается в следующем виде:

$MX\$ имя ( a, b )$

$MXj(a, b)$

Здесь *имя* и *j* - соответственно имя и номер матрицы; *a* и *b* - номера соответственно строки и столбца, задаваемые константами или ссылками на СЧА параметров транзактов. Например:

$MX5(2,1)$

$MX\$MTAB(P$ROW,P$COL)$

### 3.3.6. Блоки для работы со списками пользователя

Так как заблокированные транзакты находятся в списке текущих событий, то при большом количестве таких транзактов симулятор расходует слишком много времени на просмотр этого списка с целью выбора очередного транзакта для продвижения. Для экономии машинного времени заблокированные транзакты целесообразно помещать в так называемые *списки пользователя* и оставлять их там до тех пор, пока не выполнятся условия, позволяющие дальнейшее продвижение этих транзактов. Кроме того, размещение ожидающих транзактов в списках пользователя позволяет организовать различные дисциплины очередей, отличные от дисциплины “раньше пришел — раньше обслужен”, реализованной в списке текущих событий.

Списки пользователя представляют собой некоторые буферы, куда могут временно помещаться транзакты, выведенные из списка текущих событий. В отличие от списков текущих и будущих событий транзакты вводятся в списки пользователя и выводятся из них не автоматически, а в соответствии с логикой модели с помощью специальных блоков.

Для ввода транзактов в список пользователя служит блок LINK (ввести в список), который может быть использован в двух режимах: условном и безусловном. Ограничимся рассмотрением лишь безусловного режима, в котором блок LINK имеет следующий формат:

*имя LINK A,B*

В поле А задается имя или номер списка пользователя, в который безусловным образом помещается транзакт, вошедший в блок. Поле В определяет, в какое место списка пользователя следует поместить этот транзакт. Если в поле В записано ключевое слово FIFO, то транзакт помещается в конец списка, если LIFO — в начало списка. В других случаях транзакты упорядочиваются в соответствии с вычисленным значением поля В, где обычно записывается один из СЧА транзактов, таких как PR, M1 и ли Р. Если поле В содержит СЧА PR, то транзакты упорядочиваются по убыванию приоритета. В остальных случаях производится упорядочение по возрастанию указанного СЧА.

Например, блок

LINK 5,FIFO

помещает транзакты в список пользователя с номером 5 в порядке их поступления в блок. Блок

LINK BUFER,P\$ORDER

помещает транзакты в список пользователя с именем BUFER, упорядочивая их по возрастанию параметра с именем ORDER.

Условия, при которых транзакт помещается в список пользователя, в безусловном режиме проверяются средствами, предусмотренными разработчиком модели. Например, направить транзакт в список пользователя в случае занятости устройства можно так, как показано на рис. 16. Если устройство с именем FAC4 занято, то блок GATE не впускает транзакт в блок SEIZE, а направляет его в блок LINK с именем WAIT, и транзакт вводится в конец списка пользователя с именем BUFER.



```

.....
GATE NU  FAC4, WAIT
SEIZE    FAC4

.....

WAIT     LINK     BUFER, FIFO

.....

```

Рис. 16

Для вывода одного или нескольких транзактов из списка пользователя и помещения их обратно в список текущих событий служит блок UNLINK (вывести из списка), имеющий следующий формат:

*имя* UNLINK X A,B,C,D,E,F

В поле А указывается имя или номер списка пользователя. Поле В содержит имя блока, в который переходят выведенные из списка пользователя транзакты. В поле С указывается число выводимых транзактов или ALL для вывода всех находящихся в списке транзактов.

Операнды в полях D и E вместе со вспомогательным операндом X определяют способ и условия вывода транзактов из списка пользователя. Если поля D и E пусты, то и операнд X не используется, а транзакты выводятся с начала списка пользователя. Если поле D содержит ключевое слово BACK, то поле E и вспомогательный операнд X не используются, а транзакты выводятся с конца списка. В остальных случаях значение поля D интерпретируется как номер параметра транзактов, находящихся в списке пользователя, а из списка выводится заданное число тех транзактов, у которых значение этого параметра по отношению к значению операнда в поле E удовлетворяет условию, заданному вспомогательным операндом X. Операнд X принимает те же значения, что и в блоке TEST.

В поле F указывается имя блока, куда переходит транзакт, выходящий из блока UNLINK, если из списка пользователя не выведен ни один транзакт. Если это поле пусто, то выводимый транзакт переходит в следующий блок независимо от количества выведенных транзактов.

Например, блок

## UNLINK 5,NEXT,1

выводит из списка пользователя с номером 5 один транзакт с начала списка и направляет его в блок с именем NEXT. Блок

## UNLINK BUFER,ENT1,1,BACK

выводит из списка пользователя с именем BUFER один транзакт с конца списка и направляет его в блок с именем ENT1. Блок

## UNLINK E P\$UCH,MET2,ALL,COND,P\$COND,MET3

выводит из списка пользователя, номер которого записан в параметре UCH выводящего транзакта, и направляет в блок с именем MET2 все транзакты, содержимое параметра COND которых равно содержимому одноименного параметра выводящего транзакта. Если таких транзактов в списке не окажется, то выводящий транзакт будет направлен в блок с именем MET3, в противном случае — к следующему блоку.

Следует отметить следующие особенности выполнения блока UNLINK. Во-первых, если поля D и E содержат ссылки на СЧА транзактов, то поле D вычисляется относительно транзактов в списке пользователя, а поле E — относительно активного транзакта. Во-вторых, после вывода транзактов из списка симулятор продолжает или начинает продвижение транзакта с наивысшим приоритетом, а при равенстве приоритетов отдает предпочтение транзакту-инициатору вывода.

Каждый список пользователя имеет следующие СЧА: СН — текущая длина списка; СА — средняя длина списка (целая часть); СМ — максимальная длина списка; СС — общее число транзактов, вошедших в список; СТ — целая часть среднего времени пребывания транзакта в списке.

Воспользуемся рассмотренными блоками для моделирования многоканальной СМО с ожиданием транзактов в списке пользователя (рис. 17). Если МКУ с именем STO2 не заполнено, блок GATE впускает вновь прибывший транзакт в блок ENTER, и в М КУ занимается один канал. Если же МКУ заполнено, то блок GATE направляет транзакт в блок LINK с именем WAIT, помещающий транзакт в конец списка пользователя с именем BUFER, моделирующего очередь к МКУ. Каждый транзакт, покидающий МКУ по завершении обслуживания и освобождающий один канал, проходит

блок UNLINK и выводит один транзакт с начала списка (если список не пуст), направляя его в блок с именем ENT1 на занятие канала в МКУ.

```
STO2    STORAGE 2
EXP      FUNCTION    RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
GATE  SNF              STO2,WAIT
ENT1  ENTER            STO2
ADVANCE          160,FN$EXP
LEAVE    STO2
UNLINK          BUFER,ENT1,1
TERMINATE 1
WAIT  LINK              BUFER,FIFO
```

Рис. 17

Заметим, что для изменения дисциплины обслуживания на “позже пришел — раньше обслужен” достаточно или заменить в поле В блока LINK FIFO на LIFO, или записать в поле D блока UNLINK операнд BACK. Следует также обратить внимание на то, что блоки QUEUE-DEPART для сбора статистики об ожидающих транзактах не используются, так как почти все те же данные можно получить из статистики о списке пользователя.

Рассмотрим еще один пример, иллюстрирующий использование списков пользователя для организации нестандартных дисциплин обслуживания. Пусть в одноканальной СМО с ожиданием требуется организовать такую дисциплину, при которой приоритет от дается заявкам с наименьшим временем обслуживания. Такая модель будет иметь вид, показанный на рис. 18.

В параметр TSRV поступающих в модель транзактов в блоке ASSIGN записывается случайное время обслуживания, вычисляемое с использованием функции EXP. Если устройство SYSTEM свободно, то блок GATE впускает транзакт в блок SEIZE, и устройство занимается на время P\$TSRV. Если же в момент поступления транзакта устройство занято, то блок GATE направляет транзакт в блок LINK, который вводит транзакт в список пользователя LINE, упорядочивая транзакты по возрастанию времени обслуживания, записанного в параметре P\$TSRV. Блок UNLINK по освобождении устройства выводит с начала списка транзакт с наименьшим временем обслуживания, обеспечивая тем самым заданную дисциплину.

EXP	FUNCTION	RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915		
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3		
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9		
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8		
GENERATE	100,FN\$EXP	
ASSIGN	TSRV,80,EXP	
GATE NU	SYSTEM,WAIT	
SFAC	SEIZE	SYSTEM
ADVANCE	P\$TSRV	
RELEASE	SYSTEM	
UNLINK	LINE,SFAC,1	
TERMINATE	1	
WAIT	LINK	LINE,P\$TSRV

Рис. 18

### 3.4. Управляющие операторы GPSS/PC

Для управления прогоном модели используются управляющие операторы GPSS/PC. С одним из них — оператором START — мы уже сталкивались при рассмотрении блока TERMINATE. Оператор START (начать) имеет следующий формат:

START A,B,C,D

Поле А содержит константу, задающую начальное значение счетчика завершений. В поле В может быть записано ключевое слово NP — признак подавления формирования стандартного отчета по завершении моделирования. Если поле В пусто, то по окончании прогона модели формируется отчет со стандартной статистической информацией о всех объектах модели (см. разд. 5). Поле С не используется и сохранено для совместимости со старыми версиями GPSS. Поле D может содержать 1 для

включения в отчет списков текущих и будущих событий. Если поле D пусто, то выдача в отчет содержимого этих списков не производится.

Оператор SIMULATE (моделировать) устанавливает предел реального времени, отводимого на прогон модели. Если прогон не завершится до истечения этого времени, то он будет прерван принудительно с выдачей накопленной статистики в отчет.

Оператор SIMULATE имеет единственный операнд A, содержащий предельное время моделирования в минутах, задаваемое константой. Оператор размещается перед оператором START, начинающим лимитированный прогон.

Оператор RMULT (установить значения генераторов) позволяет перед началом прогона установить начальные значения генераторов случайных чисел RN, определяющие генерируемые ими последовательности. Поля A-G оператора могут содержать начальные значения генераторов соответственно RN1-RN7, задаваемые константами. Начальные значения генераторов, не установленные операторами RMULT, совпадают с номерами генераторов.

Оператор RESET (сбросить) сбрасывает всю статистическую информацию, накопленную в процессе прогона модели. При этом состояние аппаратных, динамических и запоминающих объектов, а также генераторов случайных чисел сохраняется, и моделирование может быть возобновлено с повторным сбором статистики. Оператор не имеет операндов.

С оператором RESET связано различие между относительным (СЧА C1) и абсолютным (СЧА AC1) модельным временем. Таймер относительного времени C1 измеряет модельное время, прошедшее после последнего сброса статистики оператором RE SET, а таймер абсолютного времени

AC1 — модельное время, прошедшее после начала первого прогона модели. Если не использовалось ни одного оператора RESET, то значения этих таймеров совпадают. Оператор RESET устанавливает таймер C1 в ноль и не влияет на таймер AC1.

Оператор RESET используется обычно при моделировании нестационарных процессов, когда требуется собрать статистику по отдельным интервалам стационарности или исключить влияние переходного периода на собираемую статистическую информацию.

Пусть, например, в модели, приведенной на рис. 18, необходимо отбросить статистику, собираемую на первой тысяче транзактов. Это может быть сделано способом, показанным на рис. 19.

Первый оператор START начинает прогон модели длиной 1000 транзактов (переходный период). Поскольку статистика, накопленная на этом периоде, не используется, в поле B оператора указан признак подавления

формирования отчета NP. Оператор RESET сбрасывает накопленную статистику, не изменяя состояния модели. Второй оператор START начинает основной прогон модели с формированием отчета по завершении прогона.

```

EXP      FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
ASSIGN        TSRV,80,EXP
GATE NU       SYSTEM,WAIT
SFAC      SEIZE      SYSTEM
ADVANCE              P$TSRV
RELEASE          SYSTEM
UNLINK          LINE,SFAC,1
TERMINATE      1
WAIT      LINK              LINE,P$TSRV
START              1000,NP
RESET
START              10000

```

Рис. 19

Оператор CLEAR (очистить) очищает модель, подготавливая ее к повторному прогону. При этом сбрасывается вся накопленная в предыдущем прогоне статистика, из модели удаляются все транзакты, и она приводится к исходному состоянию, как перед первым прогоном. Устанавливаются в ноль сохраняемые величины и матрицы, что следует учитывать при использовании этих объектов для хранения исходных данных. Исключение составляют генераторы случайных чисел, которые не возвращаются к своим начальным значениям, что позволяет повторить прогон модели на новой последовательности случайных чисел. Оператор не имеет операндов.

Оператор CLEAR используется обычно для организации нескольких независимых прогонов модели на разных последовательностях случайных чисел. Перед повторением прогона можно при необходимости переопределить отдельные объекты модели, например емкости многоканальных устройств.

Пусть, например, требуется повторить прогон модели, приведенной на рис. 17, три раза при емкости МКУ, равной 1, 2 и 3. Это может быть выполнено так, как показано на рис. 20. После каждой очистки модели оператором CLEAR оператор STORAGE устанавливает новое значение емкости МКУ с именем STO2.

Оператор END (закончить) завершает *сеанс* работы с GPSS/PC и возвращает управление в операционную систему. Оператор не имеет операндов.

```

STO2      STORAGE 1
EXP       FUNCTION   RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE   100,FN$EXP
GATE SNF   STO2,WAIT
ENT1       ENTER     STO2
ADVANCE    160,FN$EXP
LEAVE      STO2
UNLINK     BUFER,ENT1,1
TERMINATE  1
WAIT       LINK       BUFER,FIFO
START      10000
CLEAR
STO2       STORAGE 2
START      10000

```

CLEAR		
STO2	STORAGE	3
START	10000	

Рис. 20

Как правило, управляющие операторы не включаются в исходную программу, т.е. не имеют номеров строк, а вводятся пользователем непосредственно с клавиатуры ПК.

## 3.5. Некоторые приемы конструирования GPSS-моделей

### 3.5.1. Косвенная адресация

В рассматривавшихся до сих пор примерах моделей ссылки на различные объекты GPSS/PC производились исключительно по данным им произвольным именам. Такая *адресация* объектов удобна, когда речь идет о небольшом числе объектов каждого типа. Если же число объектов некоторого типа велико, то во избежание пропорционального роста количества блоков в модели используют ссылки на эти объекты по их номерам с использованием так называемой *косвенной адресации*.

Идея косвенной адресации заключается в том, что каждый транзакт в некотором своем параметре содержит номер того или иного объекта, а в полях блоков, адресуемых к объектам, записывается ссылка на этот параметр транзакта. Проиллюстрируем применение косвенной адресации на примере следующей модели.

EXP	FUNCTION	RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915		
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3		
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9		
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8		
CLASS	FUNCTION	RN1,D3



```

.333,1/.667,2/1,3
MEAN      FUNCTION      P$TYPE,L3
1,70/2,80/3,90
PRIOT     VARIABLE      4-P$TYPE
STO2      STORAGE       2
WTIME     QTABLE        LINE,50,50,10
TTIME     TABLE        M1,100,100,12
GENERATE   100,FN$EXP
ASSIGN     TYPE,FN$CLASS
PRIORITY   V$PRIOT
QUEUE      LINE
QUEUE      P$TYPE
ENTER      STO2
DEPART     P$TYPE
DEPART     LINE
ADVANCE    FN$MEAN,FN$EXP
LEAVE      STO2
TABULATE   TTIME
TERMINATE  1

```

Рис. 21

Пусть на вход моделируемой многоканальной СМО с двумя каналами обслуживания поступает пуассоновский поток заявок со средним интервалом поступления 100 единиц модельного времени. Каждая заявка с равной вероятностью  $1/3$  относится к одному из трех классов: 1, 2 или 3, а среднее время обслуживания заявок каждого типа составляет соответственно 70, 80 и 90 единиц модельного времени. Чем меньше среднее время обслуживания заявки, тем выше ее приоритет. Необходимо построить модель, позволяющую оценить средние значения времени ожидания заявок каждого типа, а также распределения общего времени ожидания в очереди и общего времени пребывания в системе. Такая модель имеет вид, показанный на рис. 21.

Переменная PRIOT служит для вычисления приоритета транзакта как функции его класса, содержащегося в параметре с именем TYPE. Транзакты класса 1 (P\$TYPE=1) получают приоритет 3, транзакты класса 2 — приоритет 2 и транзакты класса 3 — приоритет 1.

В блоке ASSIGN в параметр TYPE транзактов записывается класс заявки, разыгрываемый с помощью функции CLASS. В следующем блоке PRIORITY с помощью переменной PRIOT определяется приоритет транзактов, первоначально равный 0 (отсутствует поле E в блоке GENERATE).

Далее каждый транзакт “отмечается” в блоках QUEUE в двух очередях. Очередь с именем LINE является общей для транзактов всех классов. Входя в следующий блок QUEUE, транзакт отмечается в очереди с номером 1, 2 или 3 в зависимости от класса заявки, записанного в параметре TYPE. Аналогичным образом фиксируется уход из очередей в блоках DEPART. Таким образом, в модели создается четыре объекта типа “очередь”: одна очередь с именем LINE и три с номерами 1, 2 и 3. При этом три последние очереди создаются одной парой блоков QUEUE-DEPART! В этом и заключается эффект косвенной адресации.

Как уже отмечалось ранее, каждому имени объекта симулятор сам ставит в соответствие некоторый номер. При ссылках на объекты одного и того же типа одновременно по именам и номерам, как это имеет место в рассматриваемом примере, существует опасность параллельной адресации к одному и тому же объекту вместо двух разных или, наоборот, к двум разным объектам вместо одного. Так, в рассматриваемой модели мы, вообще говоря, не знаем, какой именно номер поставит симулятор в соответствие имени очереди LINE. Если этот номер будет от 1 до 3, то это приведет к ошибке, так как в модели окажется не четыре очереди, а три, причем в одну из них будет заноситься информация как обо всех транзактах, так и дополнительно о транзактах одного из трех классов. Как избежать такой ситуации?

К счастью, в большинстве случаев об этом можно не заботиться, поскольку симулятор ставит в соответствие именам объектов достаточно большие номера, начиная с 10000. При необходимости же можно воспользоваться оператором EQU, о котором уже говорилось выше, и самостоятельно сопоставить имени объекта желаемый номер. Например, в рассматриваемой модели для того, чтобы очередь с именем LINE имела номер 4, достаточно записать оператор:

LINE EQU 4

### 3.5.2. Обработка одновременных событий

Так как модельное время в GPSS целочисленно, то оказывается вполне вероятным одновременное наступление двух или более событий, причем вероятность этого тем больше, чем крупнее выбранная единица модельного времени. В некоторых случаях одновременное наступление нескольких событий, или так называемый *временной узел*, может существенно нарушить логику модели.

Рассмотрим, например, еще раз модель на рис. 14. Здесь может образоваться временной узел между событиями “поступление транзакта на вход модели” и “завершение обслуживания в МКУ”. Если непосредственно перед завершением обслуживания были заняты оба канала МКУ, то обработка временного узла зависит от последовательности транзактов, соответствующих событиям, в списке текущих событий.

Предположим, что первым в списке расположен транзакт, освобождающий канал МКУ. Тогда вначале будет обработан этот транзакт, т.е. событие “завершение обслуживания в МКУ”, причем условие “МКУ STO2 не заполнено”, проверяемое в блоке GATE, станет истинным. Затем будет обработан транзакт, поступивший на вход модели, в блок GATE с именем ENT1, из блока GENERATE или из блока TRANSFER в безусловном режиме. При этом транзакт будет впущен в блок ENTER, и МКУ в тот же момент модельного времени снова окажется заполненным. Такая ситуация при обработке временного узла представляется естественной.

Предположим теперь, что первым в списке текущих событий расположен транзакт, поступающий на вход модели. Так как условие “МКУ STO2 не заполнено” ложно, то блок GATE направит этот транзакт в блок с именем REFUS. Таким образом, в модели будет зафиксирован отказ в обслуживании, хотя в этот же момент модельного времени, после обработки транзакта, освобождающего канал, МКУ станет доступным.

Порядок расположения транзактов, соответствующих рассматриваемым событиям, в списке текущих событий случаен, и в среднем в половине случаев временной узел будет обрабатываться не так, как нужно. В результате статистика, связанная с отказами, окажется искаженной.

Для правильной обработки временного узла надо обеспечить такой порядок расположения транзактов в списке текущих событий, чтобы транзакт, освобождающий МКУ, всегда располагался первым. Этого можно добиться, управляя приоритетами транзактов (рис. 22).

```

EXP          FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE      100,FN$EXP
ENT1          GATE SNF      STO2,REFUS
ENTER         STO2
PRIORITY      1
ADVANCE              160,FN$EXP
LEAVE         STO2
TERMINATE     1
REFUS         TRANSFER      .1,,OUT
ADVANCE              250,FN$EXP
TRANSFER      ,ENT1
OUT           TERMINATE     1

```

Рис. 22

Транзакты, поступающие в модель через блок GENERATE, имеют нулевой приоритет. Такой же приоритет имеют транзакты, получившие отказ в обслуживании, направленные в блок с именем REFUS и затем повторно поступающие в блок с именем ENT1. Те же транзакты, что поступают на обслуживание, повышают приоритет до 1 в блоке PRIORITY, и после выхода из блока ADVANCE возвращаются из списка будущих в список текущих событий, располагаясь в начале списка. Таким образом, нужный порядок транзактов обеспечивается, и временной узел будет обработан правильно.

Опасность неверной обработки временных узлов характерна для моделей со списками пользователя. Рассмотрим, например, еще раз модель на рис. 18. Здесь также возможен временной узел между событиями “приход транзакта” и “завершение обслуживания транзакта”.

Пусть первым в списке текущих событий располагается вновь пришедший транзакт. Так как устройство с именем SYSTEM занято, то блок

GATE направит этот транзакт в блок LINK, и он будет введен в список пользователя с именем LINE. Затем будет обработан транзакт, освобождающий устройство. Проходя через блок UNLINK, он выведет транзакт с начала списка пользователя и направит его в список текущих событий, где тот продвинется в блок SEIZE, занимая устройство SYSTEM.

Если же первым в списке текущих событий располагается транзакт, освобождающий устройство, то он выведет первый из ожидающих транзактов из списка пользователя в список текущих событий, где тот расположится после вновь пришедшего транзакта. Поэтому первым будет обработан вновь пришедший транзакт, который пройдет через блок GATE и займет устройство “без очереди”. Транзакт-очередник, который был выведен из списка пользователя, “застрянет” перед блоком SEIZE и после очередного освобождения устройства займет его, нарушая, в свою очередь, логику работы модели.

Проведенный анализ показывает, что для правильной обработки временного узла необходимо обеспечить такой порядок расположения транзактов в списке текущих событий, чтобы первым всегда располагался вновь пришедший транзакт. В рассматриваемом случае этого можно добиться, используя блок PRIORITY с операндом BU (рис. 23).

Перед освобождением устройства обслуженный транзакт проходит через блок PRIORITY, который, оставляя неизменным приоритет транзакта PR, переводит его в конец списка текущих событий. При новом просмотре списка в случае наличия временного узла начинает обрабатываться вновь поступивший транзакт. Так как устройство еще занято, он направляется блоком GATE в список пользователя. При повторной обработке обслуженного транзакта тот освобождает устройство и выводит очередной транзакт из списка пользователя. Таким образом, правильная обработка временного узла обеспечивается и в этом случае.

EXP	FUNCTION	RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915		
.7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3		
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9		
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8		
GENERATE	100, FN\$EXP	
ASSIGN	TSRV, 80, EXP	
GATE NU	SYSTEM, WAIT	

SFAC	SEIZE	SYSTEM
ADVANCE		P\$TSRV
PRIORITY	PR,BU	
RELEASE	SYSTEM	
UNLINK	LINE,SFAC,1	
TERMINATE	1	
WAIT	LINK	LINE,P\$TSRV

Рис. 23

## 3.6. Команды GPSS/PC и технология работы с пакетом

### 3.6.1. Загрузка интегрированной среды

Пакет GPSS/PC включает в себя два основных модуля: модуль GPSSPC.EXE, представляющий интегрированную среду, в которой производится ввод, редактирование, отладка и выполнение модели, и модуль GPSSREP T.EXE, предназначенный для получения стандартного отчета GPSS/PC. Загрузка обоих модулей производится обычным образом из командной строки MS DOS или из программы-оболочки Norton Commander.

После загрузки интегрированной среды на экране появляется “заставка” с названием пакета: начинается так называемый *сеанс* работы с GPSS/PC. Затем заставка гасится, и появляется экран, разделенный на две части: большая верхняя часть содержит так называемое *окно данных*, меньшая нижняя часть — *окно команд*. Окно данных в начальный момент пусто, в окне команд в верхней *командной строке* высвечен символ “приглашения”>, сигнализирующий о готовности системы принимать команды.

### 3.6.2. Ввод новой модели

Если исходная программа с моделью еще не введена и не записана на диске, то необходимо ввести ее с клавиатуры. Ввод производится в командную строку. Сначала вводится номер строки очередного оператора и нажимается клавиша **Пробел**. Курсор автоматически перемещается к началу

следующего поля — поля имени, и в позиции курсора высвечивается символ **L**, сигнализирующий о том, что вы находитесь в поле имени (LABEL — метка). Если оператор имеет имя, необходимо ввести его и нажать клавишу **Пробел**, в противном случае — сразу нажать клавишу **Пробел**. В любом случае курсор переходит к началу следующего поля — поля операции, о чем сигнализирует символ **V** (VERB — глагол) в позиции курсора. Необходимо ввести название оператора и нажать клавишу **Пробел**. Очень удобным является то, что название оператора не обязательно вводить полностью: как только транслятор распознает оператор по нескольким первым буквам, он после нажатия клавиши **Пробел** сам дополнит его до полного названия.

При синтаксической ошибке в операторе под командной строкой появляется указатель на место ошибки, причем ошибочный символ не вводится. Необходимо в этом случае повторить ввод символа.

Аналогичным образом вводятся поля операндов, при этом в позиции курсора высвечивается обозначение текущего поля (**A**, **B**, ... ,**G**). Для перехода к следующему полю операндов вводится запятая, для перехода к полю комментариев — **Пробел**. При переходе курсора в поле комментариев в позиции курсора высвечивается символ **;**, сигнализирующий о возможности начать ввод комментария.

По окончании ввода последнего поля операндов или комментария следует нажать клавишу **Enter**, при этом введенный оператор транслируется и отображается в окне данных, а командная строка очищается, и в ее первой позиции снова появляется символ “приглашения”.

По мере ввода новых операторов окно данных заполняется, и по окончании ввода в нем находится вся исходная программа в последовательности ввода, необязательно совпадающей с последовательностью нумерации строк. Для отображения в окне данных исходной программы в последовательности нумерации строк необходимо ввести в командную строку команду **DISPLAY** (отобразить). Эта команда, как и все остальные команды GPSS/PC, вводится без номера строки. С помощью команды **DISPLAY** можно также вывести в окно данных отдельную строку, указав ее номер в поле **A** команды, или последовательность строк, указав начальный и конечный номера в полях **A** и **B** соответственно.

### 3.6.3. Редактирование текста модели

Удалить строки из исходной программы можно командой **DELETE** (удалить), указав в полях **A** и **B** начальный и конечный номера удаляемой

последовательности. Для удаления одной строки достаточно ввести лишь поле А.

При необходимости вставить в текст новый оператор, поместив его между уже введенными операторами, достаточно ввести его с промежуточным номером строки. Вы можете перенумеровать строки, введя команду RENUMBER (перенумеровать), в поле А которой указывается номер первой строки, а в поле В — шаг перенумерации.

Отредактировать содержимое строки можно с помощью команды EDIT (редактировать), в поле А которой указывается номер редактируемой строки. При вводе такой команды в командной строке появляется редактируемая строка. Подводя курсор к нужным позициям строки, вы можете внести в нее необходимые изменения. По окончании редактирования следует нажать клавишу **Enter**, и отредактированная строка перенесется в окно данных, заменив в исходной программе первоначальную строку с этим номером. Вы можете убедиться в этом, введя команду DISPLAY.

Если редактируемый оператор короткий, а изменений в нем много, то редактирование удобнее произвести, введя измененный оператор с тем же номером строки.

### 3.6.4. Запись и считывание модели с диска

Если работа с моделью предполагается и по окончании данного сеанса, то после ввода и редактирования исходную программу имеет смысл записать на диск. Для этого необходимо ввести команду SAVE (сохранить), в поле А которой указывается имя файла, в который будет записана модель. Файл должен иметь расширение **.GPS**.

Записав модель в файл, вы сможете в следующем сеансе работы с GPSS/PC не вводить ее заново с клавиатуры, а считать с диска, введя команду @спецификация\_файла, где спецификация\_файла - полное имя файла, которое вы дали исходной программе в команде SAVE, включающее расширение **.GPS**. При выполнении команды @ операторы исходной программы по мере их считывания из файла транслируются и выводятся в окно данных.

### 3.6.5. Прогон модели и наблюдение за моделированием

После того, как исходная программа модели введена с клавиатуры или считана с диска и оттранслирована, в памяти ПК создастся текущая модель, и



теперь можно выполнить ее прогон. Для этого в командную строку необходимо ввести управляющий оператор **START**, указав в поле **A** соответствующее начальное значение счетчика завершений. После нажатия клавиши **Enter** оператор **START** переносится в окно данных, и прогон модели начинается. Об этом сигнализирует сообщение **Simulation in Progress**, появляющееся в нижней строке командного окна — *строке состояния*, а также так называемый *индикатор моделирования*, мигающий в правой стороне нижней части окна данных.

Если прогон модели достаточно длинный, то можно наблюдать за процессом моделирования, открывая те или иные *графические окна*. Это производится путем нажатия клавиши **Alt** одновременно с символьной клавишей с первой буквой на звание окна.

Например, после нажатия клавиш **Alt+B** в верхней части экрана на месте окна данных появляется *окно блоков* (**BLOCKS**), изображающее динамику продвижения транзактов через блок-схему модели. Рядом с каждым блоком выводится текущее число транзактов в нем, которое обновляется в процессе моделирования. Нажав клавиши **Alt+N**, вы можете заменить эту информацию на общее число транзактов, прошедших через каждый блок. Блок, в котором находится активный транзакт, выделен повышенной яркостью (на цветных мониторах — другим цветом).

Нажав клавиши **Alt+F**, вы можете наблюдать *окно устройств* (**FACILITIES**), в котором наглядно отображена информация о текущем состоянии каждого устройства модели: его использовании, занятости, очереди к нему.

Аналогичную информацию о многоканальных устройствах можно получить, нажав **Alt+S** и открыв *окно памяти* (**STORAGES**).

Если в модели используются статистические таблицы, то, нажав клавиши **Alt+T**, вы откроете *окно таблиц* (**TABLES**) с гистограммой распределения соответствующего атрибута модели, обновляющейся в процессе моделирования. Над гистограммой выводятся также текущие значения среднего и среднеквадратического отклонения табулируемого атрибута.

Если в модели используются матрицы, то, нажав клавиши **Alt+M**, вы откроете *окно матриц* (**MATRICES**), в котором можно наблюдать обновляющиеся в процессе моделирования значения элементов матриц.

Находясь в любом из перечисленных окон, вы можете путем нажатия клавиш **Alt+L** включить *трассировку* активного транзакта. При этом в верхней части окна появляется строка, содержащая информацию о текущем модельном времени, но мере активного транзакта и его продвижении через

блок-схему модели. Отключить трассировку можно повторным нажатием этих же клавиш.

Перемещение внутри окна любого типа к тому или иному объекту этого типа осуществляется путем нажатия клавиш управления курсором **PgUp**, **PgDn** и **End**. Возвращение в окно данных производится путем нажатия клавиш **Alt+D**.

Следует заметить, что наблюдение графических окон и особенно строки трассировки существенно замедляет моделирование, и при длинных прогонах моделей этой возможностью не следует злоупотреблять.

Открытие того или иного окна может быть выполнено также с помощью команды **WINDOW** (окно), в поле **A** которой указывается имя окна, однако удобнее это делать так, как описано выше.

Кроме графических окон внутри любого из них, кроме окна данных, может быть открыто до четырех *микроокон*. Микроокна открываются и закрываются командой **MICROWINDOW** (микроокно), имеющей следующий формат:

**MICROWINDOW A,B,C ; комментарий**

В поле **A** указывается номер микроокна — константа 1, 2, 3 или 4. Поле **B** содержит наблюдаемую величину — любой **СЧА** модели. Поле **C** определяет состояние микроокна в результате выполнения команды: **ON** — открыто, **OFF** — закрыто. Если поле **C** пусто, то по умолчанию команда открывает заданное микроокно. В поле комментария может быть задано название микроокна длиной до восьми символов.

При открытии любого окна заданные микроокна с обновляющейся в процессе моделирования информацией появляются в правой части соответствующего окна. Микроокно имеет форму прямоугольника с названием над рамкой, если оно было задано в комментарии к команде **MICROWINDOW**.

В процессе моделирования можно также наблюдать одновременно до двух *графиков зависимостей* любых **СЧА** модели от модельного времени. Для этого необходимо до запуска модели ввести одну или две команды **PLOT** (начертить), имеющие следующий формат:

**PLOT A,B,C,D ; комментарий**

В поле А указывается аргумент зависимости — любой СЧА модели. Поле В должно содержать максимальное значение этого СЧА, определяющее масштаб изображения по оси Y. Операнд В задается константой, значение которой должно быть не менее 13. Поля С и D определяют начальное и конечное значения модельного времени, определяющие масштаб изображения по оси X. Эти операнды также задаются константами. В поле комментария может быть задан заголовок графика длиной до 34 символов.

График обновляется при каждом изменении модельного времени, если оно попадает в диапазон, заданный операндами С и D. Указанный в поле А СЧА-аргумент вычисляется относительно первого транзакта, обрабатываемого после изменения модельного времени.

Процесс моделирования можно прервать, нажав одну из клавиш **Esc** или **Home**. При этом в строке состояния командного окна появляется сообщение о номере активного транзакта, обрабатываемого симулятором в момент прерывания. Вы можете узнать значения интересующих вас стандартных числовых атрибутов модели в момент прерывания, введя команду SHOW (показать), операндом которой служат отдельные СЧА или выражения из них. Значение заданного в команде СЧА или выражения выводится в окно данных или другое активное окно. Введя команду EVENTS (события), можно увидеть в окне данных содержимое списков текущих и будущих событий. Команда USERCHAINS (списки пользователя) позволяет просматривать в окне данных содержимое списков пользователя. Обе последние команды не имеют операндов.

Инициировать прерывание моделирования можно также с помощью команды STOP (остановить), имеющей следующий формат:

STOP A,B,C

В поле А указывается номер транзакта, вызывающего прерывание, задаваемый константой. Если это поле пусто, то прерывание вызывается любым транзактом. В поле В задается имя или номер блока, при входе в который происходит прерывание. Если этот операнд опущен, то прерывание происходит при входе в любой блок. В поле С указывается ON для установки условия прерывания и OFF для снятия этого условия (по умолчанию ON).

Например, команда

STOP 100,MET1

устанавливает условие прерывания моделирования при входе транзакта с номером 100 в блок с именем MET1. Команда

STOP 2

будет вызывать прерывание при каждом продвижении транзакта с номером 2, а команда

STOP ,CHAIR

при каждом входе любого транзакта в блок с именем CHAIR. Наконец, команда

STOP

без операндов будет вызывать прерывание при каждом продвижении любого транзакта, а команда

STOP „OFF

снимает все условия прерывания, установленные ранее другими командами STOP.

Прервав моделирование, можно также воспользоваться командой STEP (выполнить шаг) для пошагового выполнения модели с целью ее *отладки*. Операнд в поле А команды задает количество входов активного транзакта в блоки, которое производится при каждом выполнении команды. Обычно этот операнд равен 1, и каждое выполнение команды STEP приводит к продвижению активного транзакта к следующему блоку. Отладку с использованием команды STEP удобно проводить, находясь в окне блоков.

Для продолжения моделирования после прерывания следует ввести в командную строку команду CONTINUE (продолжить).

Команды STEP и CONTINUE могут не только вводиться в командную строку с клавиатуры, но и выбираться из *меню команд*, появляющегося в командном окне при активизации любого графического окна. Выбор производится подводом крестообразного курсора в прямоугольную область нужной команды и нажатием клавиши **Ins**. В окне блоков меню команд предоставляет также некоторые дополнительные возможности [8].

Команды STEP, CONTINUE, а также любые другие часто используемые команды удобно загрузить на функциональные клавиши **F1-F10**. Для этого после ввода загружаемой команды с клавиатуры необходимо нажать клавиши **Ctrl+Fn**, где n — номер выбранной функциональной клавиши. После загрузки команды на функциональную клавишу для ее выполнения достаточно нажатия этой клавиши.

### 3.6.6. Получение и интерпретация стандартного отчета

По завершении прогона модели раздается звуковой сигнал, и в строке состояния появляются сообщения **Writing REPORT.GPS Simulation Complete Reporting ...**, сигнализирующие о том, что моделирование закончено и в данный момент производится создание отчета о прогоне модели. Затем система переходит в состояние ожидания дальнейших команд.

Отчет, создаваемый по завершении моделирования, записывается в файл со стандартным именем REPORT.GPS. Это имя может быть изменено командой REPORT (создать отчет), имеющей следующий формат:

REPORT A,B

В поле A указывается спецификация файла, в который должен быть выведен отчет. Если поле B содержит ключевое слово NOW, то отчет создается немедленно после ввода команды.

Необходимо иметь ввиду, что отчет, создаваемый автоматически по завершении прогона модели или командой REPORT, является *неформатированным*, т.е. непригодным для непосредственного просмотра. Для форматирования и создания стандартного отчета GPSS/PC необходимо завершить сеанс и выполнить программу *форматирования отчета*. Выход из интегрированной среды (завершение сеанса) производится путем ввода управляющего оператора END (закончить). При этом производится выход в MS DO S или в программу-оболочку Norton Commander.

Для форматирования отчета необходимо загрузить модуль форматирования GPSSREPT.EXE. После его загрузки на экране появляется “заставка” с названием модуля, двумя окнами в нижней части экрана и сообщениями-подсказками. В левом окне вы ведено имя файла, в котором находится неформатированный отчет (по умолчанию это файл REPORT.GPS). В правом окне выведено обозначение устройства, куда должен быть выведен форматированный отчет (по умолчанию это экран дисплея **SCRN:**). Форматированный отчет может быть также выведен на печать или на диск. Для этого в правое окно надо ввести обозначение **PRN:** или имя файла на диске соответственно. Для переключения окон используется клавиша **Enter**. Для создания отчета на выбранном устройстве следует нажать клавишу **Пробел**, для выхода из программы — клавишу **Esc**.

Если содержимое окон по умолчанию оставлено без изменения, то после нажатия клавиши **Пробел** на экране появляется отчет о последнем прогоне модели, выполненном перед завершением сеанса работы с модулем GPSSPC.EXE. Отчет содержит следующую информацию:

1) общие сведения о модели и ее прогоне, включающие модельное время начала (**START\_TIME**) и конца (**END\_TIME**) прогона, количество блоков в модели (**BLOCKS**), количество устройств (**FACILITIES**), количество многоканальных устройств (**STORAGES**), объем памяти, оставшейся свободной при прогоне модели (**FREE\_MEMORY**);

2) сведения об именах объектов модели, включающие для каждого имени идентификатор (**NAME**), присвоенное ему числовое значение (**VALUE**) и тип имени: 0, если числовое значение имени присвоено пользователем с помощью оператора EQU; 1, если числовое значение имени присвоено системой; 2, если имя является именем блока;

3) сведения о блоках модели, включающие для каждого блока номер строки исходной программы (**LINE**), номер или имя блока (**LOC**), название блока (**BLOCK\_TYPE**), количество транзактов, прошедших через блок (**ENTRY\_COUNT**), текущее количество транзактов в блоке в момент завершения моделирования (**CURRENT\_COUNT**), количество транзактов, заблокированных перед блоком в момент завершения моделирования (**RETRY**);

4) сведения об устройствах модели, включающие для каждого устройства его имя или номер (**FACILITY**), количество занятий устройства (**ENTRIES**), коэффициент использования (**UTIL.**), среднее время на одно занятие (**AVE.\_TIME**) и ряд других данных;

5) сведения о многоканальных устройствах модели, включающие для каждого МКУ его имя или номер (**STORAGE**), емкость (**CAP.**), количество свободных каналов в момент завершения моделирования (**REMAIN.**), наименьшее (**MIN.**) и наибольшее (**MAX.**) количество занятых каналов в

процессе моделирования, количество занятий МКУ (**ENTRIES**), среднее количество занятых каналов (**AVE.C.**), коэффициент использования (**UTIL.**) и ряд других данных;

6) сведения об очередях модели, включающие для каждой очереди ее имя или номер (**QUEUE**), максимальную длину очереди в процессе моделирования (**MAX.**), текущую длину очереди в момент завершения моделирования (**CONT.**), общее количество транзактов, вошедших в очередь в процессе моделирования (**ENTRIES**), и количество “нулевых” входов в очередь (**ENTRIES(0)**), среднюю длину очереди (**AVE.CONT.**), среднее время ожидания в очереди с учетом всех транзактов (**AVE.TIME**) и без учета “нулевых” входов (**AVE.(-0)**);

7) сведения о статистических таблицах модели, включающие для каждой таблицы ее имя или номер (**TABLE**), среднее значение (**MEAN**) и среднеквадратическое отклонение (**STD.DEV.**) табулируемой величины, границы частотных интервалов (**RANGE**), частоты (**FREQUENCY**) и накопленные частоты в процентах (**CUM.%**) попадания наблюдений в эти интервалы;

8) сведения о списках пользователя модели, включающие для каждого списка его имя или номер (**USER\_CHAIN**), количество транзактов в списке в момент завершения моделирования (**CHAIN\_SIZE**), среднее количество транзактов в списке (**AVE.CONT**), общее количество транзактов, вошедших в список в процессе моделирования (**ENTRIES**), максимальное количество транзактов, находившихся в списке (**MAX**), среднее время пребывания транзакта в списке (**AVE.TIME**);

9) сведения о логических переключателях модели, включающие для каждого ЛП его имя или номер (**LOGICSWITCH**) и состояние ЛП в момент завершения моделирования: 1 — “включен”, 0 — “выключен”;

10) сведения о сохраняемых величинах модели, включающие для каждой сохраняемой величины ее имя или номер (**SAVEVALUE**) и значение в момент завершения моделирования (**VALUE**);

11) сведения о матрицах модели, включающие для каждой матрицы ее имя или номер (**MATRIX**), а также список всех элементов матрицы в формате: “строка”(**ROW**), “столбец”(**COLUMN**), “значение”(**VALUE**).

Если в операторе **START** задан вывод в отчет списков текущих и будущих событий, то отчет включает в себя также сведения о транзактах, находившихся в момент завершения моделирования в этих списках. Сведения о транзактах размещаются в отчете в соответствии с размещением транзактов в каждом списке.

Информация о списке текущих событий включает в себя для каждого транзакта его номер (**XACT\_NUMBER**), приоритет (**PRI**), резидентное время

транзакта (**M1**), номер текущего блока (**CURRENT**), номер следующего блока (< B>NEXT), а также перечень всех параметров транзакта в формате: “параметр”(PARAMETER), “значение”(VALUE).

Информация о списке будущих событий включает для каждого транзакта те же данные, однако вместо резидентного времени транзакта (**M1**) выводится запланированное время выхода транзакта из списка будущих событий (**BDT**).

Разумеется, сведения об объектах того или иного типа появляются в отчете только в том случае, если в модели присутствует хотя бы один объект данного типа. Кроме того, включением в отчет сведений об объектах разных типов можно управлять с помощью так называемого установочного файла SETTINGS.GPS [8]. В отчетах о прогоне моделей, включающих в себя другие, не рассматривавшиеся здесь объекты GPSS/PC, появляется соответствующая информация и об этих объектах.

На рис. 24 приведен отчет о прогоне модели примера на рис. 21.

START_TI ME	END_TI ME	BLOC KS	FACILITI ES	STORA GES	FREE_MEMO RY
0	14617	12	0	1	274320

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT	CURRENT_COUNT	RETRY
80	1	GENERATE	150	0	0
90	2	ASSIGN	150	0	0
100	3	PRIORITY	150	0	0
110	4	QUEUE	150	0	0
120	5	QUEUE	150	0	0
130	6	ENTER	150	0	0
140	7	DEPART	150	0	0
150	8	DEPART	150	0	0
160	9	ADVANCE	150	0	0
170	10	LEAVE	150	0	0
180	11	TABULATE	150	0	0
190	12	TERMINATE	150	0	0

QUE UE	M AX	CO NT.	ENTRI ES	ENTRIES (0)	AVE.CON T.	AVE.TI ME	AVE.(- 0)
1	1	0	54	48	0.02	6.07	54.67
2	1	0	42	35	0.01	4.14	24.86
							67.20



3	1	0	54	49	0.02	6.22	46.56
LIN	2	0	150	132	0.06	5.59	

E

STORA	C	REMA	M	M	ENTRI	A	AV	UTIL
GE	AP.	IN.	IN.	AX.	ES	VL.	E.C.	.
STO2	2	2	0	2	150	1	0.66	0.328

TAB	ME	STD.D	RETR	RAN	FREQUEN	CUM
LE	AN	EV.	Y	GE	CY	.%
WTI	5.59	25.23	0			
ME						
				50	144	96.00
			—			
			50	100	3	98.00
			—			
			100	150	1	98.67
			—			
			150	200	2	100.0
			—			0
TTIME	69.48	70.88	0			
			—	100	117	78.00
			100	—	200	23
			200	—	300	8
			300	—	400	2
						100.00

Рис. 24

Отчет выводится на экран постранично. Для вывода очередной страницы необходимо нажать клавишу **Пробел**, для прекращения вывода отчета — клавишу **Esc**. По окончании вывода отчета на экране появляется сообщение **[SPACE] for another report Any other key to end.**

Для создания отчета на другом устройстве или другого отчета надо нажать клавишу **Пробел**, для выхода из программы GPSSREPT —любую другую клавишу.

Помимо отчета отдельные результаты моделирования могут быть также выведены в *базу данных* GPSS/PC [8] с помощью команд RESULT. Однофакторный *дисперсионный анализ* и получение *доверительных интервалов* для выведенных в базу данных характеристик модели могут быть выполнены с помощью команды ANOVA. Рассмотрение этих команд выходит за рамки данного издания.

# Глава 4. Моделирование сложных систем на основе GPSS

## 4.1. Моделирование работы производственного участка

### Задание

По исходным данным (Задание 13) смоделировать работу производственного участка с двумя обрабатывающими центрами.

### 4.1.1. Алгоритм решения

1. Генерируем транзакты деталей.
2. По заданным правилам присваиваем деталям соответствующий тип.
3. Отправляем детали типа 3 на ОБРЦ-II, детали типов 1 и 2 – на ОБРЦ-I, а в случае его занятости – на ОБРЦ-II.
4. Обрабатываем детали.

### 4.1.2. Иммитационная модель системы

```
;GPSS/PC Program File VN0114.GPS. (V 2)

1          CLEAR

2          RESET

10 XP      FUNCTION      RN1,C24
;Exponential Distribution Function

0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7
,1.2/.75,1.38

.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95
,2.99/.96,3.2
```

.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.999  
8,8

20            INITIAL            X\$VT1 0

30            INITIAL            X\$VT2 0

40            INITIAL            X\$VT3 0

50 ;

60 ; generating

70 ;

80            GENERATE           6 FN\$XP

90 ; set detail type

100 ; type 3 with probability = 50%

110           TRANSFER           .5 T3

120 ; type 2 with probability = 25%

130           ASSIGN            2 8

140           TRANSFER           .5 T2

150           ASSIGN            1 1

160           SAVEVALUE        VT1+ 1

170           TRANSFER           ,MAIN

180 T2        ASSIGN            1 2

190           SAVEVALUE        VT2+ 1

200           TRANSFER           ,MAIN

210 T3        ASSIGN            1 3

220           SAVEVALUE        VT3+ 1

230           ASSIGN            2 10

240 ;

250 ; main part

260 ;

270 ; if type = 3 => OBRC-II

```
280 MAIN    TEST NE          P1 3 OBRC2
290 ; if OBRC-I engaged => OBSC-II
300          GATE NU          OBRC1 OBRC2
310 ; OBRC-I
320 OBRC1 SEIZE              OBRC1
330          ADVANCE          P2
340          RELEASE          OBRC1
350          TRANSFER          ,QUIT
360 ; OBRC-II
370 OBRC2 QUEUE              OBRC2
375          SEIZE              OBRC2
377          DEPART            OBRC2
380          ADVANCE          P2
390          RELEASE          OBRC2
400 ;
410 QUIT    TERMINATE
420 ;
430 ; 16 hours
440 ;
450          GENERATE          , , 960 , 1
460          REPORT            VN0114.REP , NOW
470          TERMINATE        1
480          START              1
```

### 4.1.3. Анализ результатов иммитационного моделирования

Полные результаты одного запуска для базовых данных (время моделирования - 2 смены – 16ч, время обработки деталей типа 1 и 2 – 8 мин, типа 3 - 10 мин).

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT	CURRENT_COUNT		RETRY
80	1	GENERATE	171	0	0	
110	2	TRANSFER	171	0	0	
130	3	ASSIGN	83	0	0	
140	4	TRANSFER	83	0	0	
150	5	ASSIGN	46	0	0	
160	6	SAVEVALUE	46	0	0	
170	7	TRANSFER	46	0	0	
180	T2	ASSIGN	37	0	0	
190	9	SAVEVALUE	37	0	0	
200	10	TRANSFER	37	0	0	
210	T3	ASSIGN	88	0	0	
220	12	SAVEVALUE	88	0	0	
230	13	ASSIGN	88	0	0	
280	MAIN	TEST	171	0	0	
300	15	GATE	83	0	0	
320	OBRC1	SEIZE	47	0	0	
330	17	ADVANCE	47	0	0	
340	18	RELEASE	47	0	0	
350	19	TRANSFER	47	0	0	
370	OBRC2	QUEUE	124	23	0	
375	21	SEIZE	101	0	0	

377	22	DEPART	101	0	0
380	23	ADVANCE	101	1	0
390	24	RELEASE	100	0	0
410	QUIT	TERMINATE	147	0	0
450	26	GENERATE	1	0	0
470	27	TERMINATE	1	0	0
500	28	TERMINATE	0	0	0

FACILITY ENTRIES UTIL. AVE.\_TIME AVAILABLE OWNER PEND  
INTER RETRY DELAY

OBRC1	47	0.391	8.00	1	0	0	0	0	0
OBRC2	101	0.976	9.28	1	135	0	0	0	23

QUEUE MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME  
AVE.(-0) RETRY

OBRC2	26	23	124	2	16.71	129.37	131.49	0
-------	----	----	-----	---	-------	--------	--------	---

SAVEVALUE	VALUE	RETRY
VT1	+46	0
VT2	+37	0
VT3	+88	0

Мы видим (выделено жирным курсивом в результатах), что всего поступило 171 деталь. Среди них было 46 деталей типа 1, 37 - типа 2 и 88 – типа 3. На ОБРЦ-I 1 за 2 смены поступило и было обработано 47 деталей, на ОБРЦ-II – поступило 124, обработано было 100 (одна деталь к концу смены находилась в процессе обработки). Всего завершило процесс 147 деталей. Как видим, очередь образуется перед ОБРЦ-II.

Попробуем управлять процессом разными способами:

1. Изменяя время обработки деталей разных типов.

## 2. Изменяя скорость работы станков.

1.1 Время обработки деталей типа 1 и 2 – 6 мин, типа 3 - 10 мин.

LINE	LOC	BLOCK_TYPE		ENTRY_COUNT		
CURRENT_COUNT		RETRY				
340	18	RELEASE		55	0	0
390	24	RELEASE		102	0	0
410	QUIT	TERMINATE		157	0	0
QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME
AVE.(-0)	RETRY					
OBRC2	12	6	109	6	5.67	49.94
					52.84	0

1.2 Время обработки деталей типа 1 и 2 – 8 мин, типа 3 - 7 мин.

LINE	LOC	BLOCK_TYPE		ENTRY_COUNT					
CURRENT_COUNT	RETRY								
340	18	RELEASE		44	0	0			
390	24	RELEASE		125	0	0			
410	QUIT	TERMINATE		169	0	0			
QUEUE	MAX	CONT. ENTRIES		ENTRIES(0)	AVE.CONT.	AVE.TIME			
AVE.(-0)	RETRY								
OBRC2	10	2	128	12	3.89	29.21	32.23	0	

1.3 Время обработки деталей типа 1 и 2 – 6 мин, типа 3 - 8 мин.

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT		
CURRENT_COUNT		RETRY			
340	18	RELEASE	53	0	0
390	24	RELEASE	121	0	0
410	QUIT	TERMINATE	174	0	0
QUEUE	MAX	CONT. ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME
AVE.(-0)	RETRY				



OBRC2	9	0	122	9	3.25	25.55	27.58	0
-------	---	---	-----	---	------	-------	-------	---

2.1 При обработке на ОБРЦ-I время обработки уменьшается на 3 мин, на ОБРЦ-II время обработки неизменно.

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT
CURRENT_COUNT		RETRY	

340	19	RELEASE	59 0 0
-----	----	---------	--------

390	25	RELEASE	97 0 0
-----	----	---------	--------

410	QUIT	TERMINATE	156 0 0
-----	------	-----------	---------

QUEUE	MAX	CONT. ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME
AVE.(-0)	RETRY				

21	22	21	119	3	6.58 53.09 54.47 0
----	----	----	-----	---	--------------------

2.2 При обработке на ОБРЦ-II время обработки уменьшается на 2 мин, на ОБРЦ-I время обработки неизменно.

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT
CURRENT_COUNT		RETRY	

340	19	RELEASE	45 0 0
-----	----	---------	--------

390	26	RELEASE	120 0 0
-----	----	---------	---------

410	QUIT	TERMINATE	165 0 0
-----	------	-----------	---------

QUEUE	MAX	CONT. ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME
AVE.(-0)	RETRY				

21	12	2	123	14	2.40 18.71 21.11 0
----	----	---	-----	----	--------------------

2.3 При обработке на ОБРЦ-I и ОБРЦ-II время обработки уменьшается на 2 мин.

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT
CURRENT_COUNT		RETRY	

340	19	RELEASE	56 0 0
-----	----	---------	--------

390	26	RELEASE	117 0 0
-----	----	---------	---------

410	QUIT	TERMINATE	173 0 0
-----	------	-----------	---------

QUEUE MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME  
AVE.(-0) RETRY

21 9 3 121 12 2.63 20.83 23.13 0

Анализируя изменение количества обработанных деталей и времени задержки в очереди видим, что особенно перспективно изменение технологического процесса, при котором ускорение работы касаются одновременно всех типов станков и/или деталей. Также существенно влияет ускорение станка 2. Одно ускорение станка 1 или уменьшение времени обработки деталей типа 1 и 2 меняет ситуацию незначительно.

## 4.2. Моделирование работы участка цеха

Необходимо промоделировать работу участка цеха, состоящего из нескольких станков и обрабатывающего два потока деталей различного типа. Маршрут обработки деталей двух типов представлен на рис. 1. В таб.1 представлено распределение выполняемых операций по станкам A1, A2 и A3. Интервалы времени между поступлениями деталей и времена выполнения операций распределены равномерно. Информация о временах поступления и выполнения операций заданы в таб.2 и таб.3.

Определить для рабочего дня (8 часов) и рабочей недели (5 дней при односменном режиме) среднюю загрузку каждого станка, среднее время обработки деталей каждого типа, какова длина очередей на обработку для станков, какой размер склада необходим для данного потока деталей. Предложить способы модификации участка цеха с целью повышения эффективности его работы.

**Таблица 1. Распределение операций по станкам**

Вариант	Операция 1	Операция 2	Операция 3	Операция 4	Операция 5	Операция 6
Пример	A1	A2	A3	A1	A3	A2
1	A1	A2	A3	A3	A2	A1
2	A1	A2	A3	A3	A1	A2
3	A1	A2	A3	A1	A2	A3
4	A1	A2	A3	A2	A1	A3
5	A1	A2	A3	A2	A3	A1

6	A2	A1	A3	A1	A2	A3
7	A2	A1	A3	A1	A3	A2
8	A2	A1	A3	A2	A1	A3
9	A2	A1	A3	A2	A3	A1
10	A2	A1	A3	A3	A1	A2
11	A2	A1	A3	A3	A2	A3
12	A3	A1	A3	A1	A2	A3

**Таблица 2.**

Вариант	Интервалы поступления первого типа (мин.)	времени деталей	Интервалы поступления второго типа (мин.)	времени деталей
Пример		$30 \pm 5$		$20 \pm 5$
1		$25 \pm 4$		$25 \pm 6$
2		$20 \pm 3$		$30 \pm 7$
3		$15 \pm 5$		$35 \pm 8$
4		$10 \pm 4$		$20 \pm 5$
5		$30 \pm 5$		$10 \pm 3$
6		$15 \pm 4$		$15 \pm 6$
7		$30 \pm 10$		$15 \pm 3$
8		$20 \pm 5$		$20 \pm 5$
9		$25 \pm 4$		$10 \pm 3$
10		$45 \pm 5$		$15 \pm 5$
11		$20 \pm 4$		$15 \pm 3$
12		$10 \pm 3$		$15 \pm 5$

**Таблица 3.**

Вариант	Интервал вал	Интервал вал	Интервал вал	Интервал вал	Интервал вал	Интервал вал
	времени выполнения операции 1 (мин.)	времени выполнения операции 2 (мин.)	времени выполнения операции 3 (мин.)	времени выполнения операции 4 (мин.)	времени выполнения операции 5 (мин.)	времени выполнения операции 6 (мин.)
Пример	$5 \pm 2$	$20 \pm 4$	$10 \pm 3$	$7 \pm 3$	$15 \pm 5$	$15 \pm 5$
1	$20 \pm 4$	$5 \pm 2$	$15 \pm 5$	$15 \pm 5$	$7 \pm 3$	$10 \pm 3$
2	$10 \pm 3$	$15 \pm 5$	$5 \pm 2$	$20 \pm 4$	$10 \pm 3$	$7 \pm 3$
3	$18 \pm 3$	$10 \pm 3$	$12 \pm 5$	$20 \pm 4$	$25 \pm 8$	$12 \pm 4$
4	$12 \pm 5$	$15 \pm 5$	$18 \pm 3$	$10 \pm 3$	$5 \pm 2$	$20 \pm 4$
5	$15 \pm 5$	$20 \pm 4$	$10 \pm 3$	$18 \pm 3$	$12 \pm 5$	$20 \pm 4$
6	$10 \pm 3$	$25 \pm 8$	$5 \pm 2$	$15 \pm 5$	$18 \pm 3$	$15 \pm 5$

7	$15 \pm 5$	$12 \pm 5$	$20 \pm 4$	$5 \pm 2$	$10 \pm 3$	$18 \pm 3$
8	$20 \pm 4$	$18 \pm 3$	$10 \pm 3$	$7 \pm 3$	$15 \pm 5$	$25 \pm 8$
9	$10 \pm 3$	$15 \pm 5$	$10 \pm 3$	$12 \pm 5$	$5 \pm 2$	$20 \pm 4$
10	$25 \pm 8$	$5 \pm 2$	$12 \pm 5$	$7 \pm 3$	$10 \pm 3$	$15 \pm 5$
11	$20 \pm 4$	$10 \pm 3$	$15 \pm 5$	$5 \pm 2$	$12 \pm 5$	$25 \pm 8$
12	$12 \pm 5$	$20 \pm 4$	$25 \pm 8$	$15 \pm 5$	$5 \pm 2$	$10 \pm 3$

## Пример

Исходные данные для структуры участка цеха и интервалы времени поступления деталей на станок, выполняющий первую операцию (для деталей первого типа) и на станок, выполняющий четвертую операцию (для деталей второго типа), а также интервалы времени обработки каждым станком заданы в таблицах 1, 2 и 3.

**Таблица 4. Таблица определений для примера**

Элементы GPSS	Назначение
Транзакты:	
1-й сегмент модели	Детали первого типа
2-й сегмент модели	Детали второго типа
3-й сегмент модели	Таймер
Станки:	Выполнение
A1	операции 1 и операции 4
A2	операции 2 и операции 6
A3	операции 3 и операции 5
Очереди:	
AA1	Общая очередь к станку A1
AA2	Общая очередь к станку A2
AA3	Общая очередь к станку A3

Единица времени в модели - 1 мин.

### 4.2.1. Иммитационная модель системы

; GPSS/PC Program File EX1. (V 2, # 39560) 03-01-1999 12:36:18

100 SIMULATE

110 GENERATE 30,5 Первый сегмент модели

120	QUEUE	AA1	
130	SEIZE	A1	
140	DEPART	AA1	
150	ADVANCE	5,2	
160	RELEASE	A1	
170	QUEUE	AA2	
180	SEIZE	A2	
190	DEPART	AA2	
200	ADVANCE	20,4	
210	RELEASE	A2	
220	QUEUE	AA3	
230	SEIZE	A3	
240	DEPART	AA3	
250	ADVANCE	10,3	
260	RELEASE	A3	
270	TERMINATE		
280	GENERATE	20,5	Второй сегмент модели
290	QUEUE	AA1	
300	SEIZE	A1	
310	DEPART	AA1	
320	ADVANCE	7,3	
330	RELEASE	A1	
340	QUEUE	AA3	
350	SEIZE	A3	
360	DEPART	AA3	
370	ADVANCE	15,5	
380	RELEASE	A3	

390	QUEUE	AA2	
400	SEIZE	A2	
410	DEPART	AA2	
420	ADVANCE	15,5	
430	RELEASE	A2	
440	TERMINATE		
450	GENERATE	480	Третий сегмент модели (таймер)
460	TERMINATE	1	

В данном примере таймер настроен на выполнение моделирования в течение 8 часового рабочего дня. Для выполнения моделирования в течение 5 дней таймер должен быть откорректирован.

#### 4.2.2. Анализ результатов имитационного моделирования

Распечатка выходных данных для моделирования работы участка цеха в течение рабочего дня.

GPSS/PC Report file REPORT.GPS. (V 2, # 39560) 03-02-1999 13:45:52  
page 1

START_TIME	END_TIME	BLOCKS	FACILITIES	STORAGES
FREE_MEMORY				

0	480	36	3	0	13264
---	-----	----	---	---	-------

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT
CURRENT_COUNT	RETRY		

110	1	GENERATE	15	0	0
-----	---	----------	----	---	---

120	2	QUEUE	15	0	0
130	3	SEIZE	15	0	0
140	4	DEPART	15	0	0
150	5	ADVANCE	15	0	0
160	6	RELEASE	15	0	0
170	7	QUEUE	15	4	0
180	8	SEIZE	11	0	0
190	9	DEPART	11	0	0
200	10	ADVANCE	11	1	0
210	11	RELEASE	10	0	0
220	12	QUEUE	10	0	0
230	13	SEIZE	10	0	0
240	14	DEPART	10	0	0
250	15	ADVANCE	10	0	0
260	16	RELEASE	10	0	0
270	17	TERMINATE	10	0	0
280	18	GENERATE	24	0	0
290	19	QUEUE	24	0	0
300	20	SEIZE	24	0	0
310	21	DEPART	24	0	0
320	22	ADVANCE	24	0	0
330	23	RELEASE	24	0	0
340	24	QUEUE	24	0	0
350	25	SEIZE	24	0	0
360	26	DEPART	24	0	0
370	27	ADVANCE	24	1	0
380	28	RELEASE	23	0	0
390	29	QUEUE	23	7	0

400	30	SEIZE	16	0	0
410	31	DEPART	16	0	0
420	32	ADVANCE	16	0	0
430	33	RELEASE	16	0	0
440	34	TERMINATE	16	0	0
450	35	GENERATE	1	0	0
460	36	TERMINATE	1	0	0

**FACILITY ENTRIES UTIL. AVE.\_TIME AVAILABLE OWNER PEND  
INTER RETRY DELAY**

A1	39	0.479	5.90	1	0	0	0	0	0
A2	27	0.918	16.33	1	29	0	0	0	11
A3	34	0.877	12.38	1	40	0	0	0	0

**QUEUE MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME  
AVE.(-0) RETRY**

AA1	1	0	39	32	0.03	0.41	2.29	0
AA2	12	11	38	1	5.08	64.16	65.89	0
AA3	2	0	34	10	0.28	3.97	5.63	0

Распечатка выходных данных для моделирования работы участка цеха в течение рабочей недели.

**GPSS/PC Report file REPORT.GPS. (V 2, # 39560) 03-03-1999 11:27:02**  
page 1

**START\_TIME END\_TIME BLOCKS FACILITIES STORAGES  
FREE\_MEMORY**



0	2400	36	3	0	7056
---	------	----	---	---	------

FACILITY	ENTRIES	UTIL.	AVE._TIME	AVAILABLE	OWNER	PEND
INTER	RETRY	DELAY				

A1	199	0.522	6.31	1	0	0	0	0	0
A2	139	0.983	16.99	1	142	0	0	0	59
A3	174	0.955	13.18	1	139	0	0	0	1

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME
AVE.(-0)	RETRY					

AA1	1	0	199	155	0.06	0.74	3.36	0
AA2	59	59	198	1	29.26	354.67	356.47	0
AA3	3	1	175	23	0.88	12.06	13.89	0

## Обсуждение результатов моделирования

Таблица 5. Средняя загрузка станков (в %)

Станок	в течение 8 часов	в течение 5 рабочих дней
A1	48	52
A2	92	98
A3	88	96

Таблица 6. Максимальная длина очередей к станкам

Станок	в течение 8 часов	в течение 5 рабочих дней
A1	1	1

A2	12	59
A3	2	3

**Таблица 7. Среднее время обработки деталей на станках (в мин.)**

Станок	в течение 8 часов	в течение 5 рабочих дней
A1	5.9	6.31
A2	16.33	16.99
A3	12.38	13.18

Общее число обработанных деталей в течение 8 часов равно 40, в течение рабочей недели - 142. Эти данные могут служить основанием для расчета необходимого размера склада готовой продукции. Из результатов моделирования можно сделать вывод, что первый станок A1 загружен на 50 %. Перегружен станок A2 (об этом говорит средний процент использования 98 % и длина очереди - 59).

Станок A3 загружен оптимально.

Для повышения эффективности работы данного участка цеха при данном потоке деталей можно использовать два станка A2. Для проверки данного предположения надо внести в программу модели изменения.

**; GPSS/PC Program File EX2. (V 2, # 39560) 03-03-1999 12:51:15**

**100 SIMULATE**

**105 A2 STORAGE 2** Второй станок моделируется как  
накопитель

**110 GENERATE 30,5**

**120 QUEUE AA1**

**130 SEIZE A1**

**140 DEPART AA1**

**150 ADVANCE 5,2**

**160 RELEASE A1**

**170 QUEUE AA2**

180	ENTER	A2
190	DEPART	AA2
200	ADVANCE	20,4
210	LEAVE	A2
220	QUEUE	AA3
230	SEIZE	A3
240	DEPART	AA3
250	ADVANCE	10,3
260	RELEASE	A3
270	TERMINATE	
280	GENERATE	20,5
290	QUEUE	AA1
300	SEIZE	A1
310	DEPART	AA1
320	ADVANCE	7,3
330	RELEASE	A1
340	QUEUE	AA3
350	SEIZE	A3
360	DEPART	AA3
370	ADVANCE	15,5
380	RELEASE	A3
390	QUEUE	AA2
400	ENTER	A2
410	DEPART	AA2
420	ADVANCE	15,5
430	LEAVE	A2
440	TERMINATE	

**450      GENERATE    2400**

**460      TERMINATE    1**

Распечатка выходных данных для моделирования работы участка цеха в течение рабочего дня.

**GPSS/PC Report file REPORT.GPS. (V 2, # 39560) 03-03-1999 12:48:55**  
**page 1**

**START\_TIME    END\_TIME    BLOCKS    FACILITIES    STORAGES**  
**FREE\_MEMORY**

**0      480    36      2      1      13888**

**FACILITY    ENTRIES    UTIL.    AVE.\_TIME    AVAILABLE    OWNER    PEND**  
**INTER    RETRY    DELAY**

**A1      39    0.462      5.69    1      40    0    0    0    0**

**A3      35    0.927      12.71    1      38    0    0    0    3**

**QUEUE      MAX    CONT.    ENTRIES    ENTRIES(0)    AVE.CONT.    AVE.TIME**  
**AVE.(-0)    RETRY**

**AA1      1    0    39      27    0.08      0.95    3.08    0**

**AA2      1    0    36      31    0.04      0.58    4.20    0**

**AA3      3    3    38      3    1.38      17.47    18.97    0**

**STORAGE      CAP.    REMAIN.    MIN.    MAX.    ENTRIES    AVL.    AVE.C.**  
**UTIL.    RETRY    DELAY**

**A2      2    1    0    2      36    1    1.30    0.649    0    0**

Распечатка выходных данных для моделирования работы участка цеха в течение рабочей недели.

START_TIME	END_TIME	BLOCKS	FACILITIES	STORAGES	FREE_MEMORY
0	2400	36	2	1	12896

FACILITY	ENTRIES	UTIL.	AVE._TIME	AVAILABLE	OWNER	PEND	INTER	RETRY	DELAY
A1	200	0.509	6.11	1	201	0	0	0	0
A3	184	0.985	12.85	1	186	0	0	0	14

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
AA1	1	0	200	155	0.07	0.78	3.49	0
AA2	1	0	190	170	0.03	0.37	3.50	0
AA3	16	14	198	3	7.49	90.79	92.19	0

STORAGE	CAP.	REMAIN.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
A2	2	1	0	2	190	1	1.34	0.672	0	0

По результатам данного варианта моделирования можно сделать вывод: использование двух станков A2 позволило ликвидировать очередь к данному станку, снизив среднюю загрузку станков A2 до 67 % (чего и следовало ожидать). Но при этом перераспределилась нагрузка на станок A3: при средней нагрузке в 99 % максимальная очередь на обработку станком A3 возросла до 16. Можно продолжить моделирование с целью определения оптимальной структуры участка цеха при заданном потоке деталей.

Если структуру цеха менять нельзя , то , используя язык моделирования GPSS, можно подобрать такой поток деталей, который давал бы возможность оптимально загружать данное оборудование.

### **Задание для домашней подготовки**

1. Изучите основные операторы языка GPSS.
2. Изучите схему модели и программу примера.
3. Ознакомьтесь с результатами решения примера, выясните смысл выводимых на печать данных.
4. Внесите в программу необходимые изменения в соответствии с вашим вариантом .
5. Продумайте меры по улучшению выходных параметров участка цеха (повышение производительности, улучшение использования оборудования).
- 6.

## **4.3. Порядок оформления отчета**

1. Ознакомиться с текстом программы на GPSS для примера.
2. Внести изменения в этот текст в соответствии с вариантом , заданным преподавателем.
3. Войти в директорию GPSS, запустить программу GPSSPC.
4. После загрузки интерпретатора GPSS загрузить модель, используя команду загрузки @EX1.
5. Внести изменения в текст примера в соответствии с заданием.
6. Сохранить измененный текст, используя команду SAVE VAR1
7. Закончить работу интерпретатора, введя команду END.
8. Вновь запустить интерпретатор, при этом загрузить ваш вариант текста модели: @VAR1.
9. Выполнить моделирование, используя команду START 1. При выполнении моделирования можно посмотреть динамическое использование устройств, памяти и проследить прохождение транзактов по блоком . Для этого необходимо нажать следующие клавиши: <Alt>+F - для переключения на "окно устройств", <Alt>+S - для переключения на "окно памяти", <Alt>+B - для переключения на "окно блоков".
10. Для просмотра статистики использовать программу GPSSREPT.
11. Составить отчет .

### **4.3.1. Содержание отчета**

1. Тексты программ с внесенными в них изменениями.
2. Результаты моделирования.
3. Объяснение влияния изменений на результаты моделирования.

## Задания для самопроверки

1. Приведите примеры объектов, которые целесообразно исследовать с помощью имитационного моделирования СМО.
2. Объясните, выполнение каких действий вызывает оператор TRASFER P,1,4.
3. Опишите на языке GPSS событие, связанное с приходом транзакта на вход ОА с именем СОМ.
4. Опишите на языке GPSS входной поток транзактов, интервал времени между появлениями транзактов - случайная величина с экспоненциальным законом распределения с интенсивностью  $\lambda = 0.2$  1/мин.
5. Запишите на языке GPSS оператор перехода к оператору с меткой LAB, если длина очереди  $\geq 5$  превышает 10 транзактов.

### 4.3.2. Создание выходного файла

После завершения процесса моделирования автоматически создается системный файл с именем REPORT.GPS, содержащий стандартную выходную информацию. Для представления результатов моделирования в виде текстового ASCII – файла с именем ASTUD.TXT необходимо после выхода в DOS (с помощью команды END) выполнить команду  
GPSSREPT REPORT.GPS ASTUD.TXT.

Здесь ASTUD.TXT – стандартное имя выходного файла с результатами выполнения заданий лабораторного практикума. Просмотр, редактирование или вывод на печать результатов, содержащихся в файле ASTUD.TXT, может осуществляться с помощью команд MSDOS или NORTON COMMANDER. Для длительного хранения результатов целесообразно сохранять файл ASTUD.TXT с индивидуальным именем в подкаталоге STUD.

### 4.3.3. Описание элементов выходного файла

Отформатированный выходной файл статистики состоит из подразделов, содержащих стандартную статистику об объектах GPSS/PC, используемых в данной модели (FACILITY, QUEUE, SYORAGE и т.д.). Начинается файл статистики с заголовка, который берется из поля комментария, расположенного перед началом программы. Заголовок появляется на каждой странице файла статистики. После заголовка автоматически останавливается подзаголовок, который содержит имя неотформатированного файла

статистики, номер версии GPSS/PC, серийный номер, дату и время моделирования. Например,

GPSS/PC Report file TEST (V2) 01-03-1993 12:00:00

Далее следует строка, содержащая основную информацию о результатах работы модели:

START TIME    END TIME    BLOCKS    FACILITIES    STORAGES    FREE  
MEMORY

Элементы статистики, представленные в этой строке, имеют следующее содержание:

START TIME – абсолютное модельное время в момент начала моделирования (эквивалентное абсолютному модельному времени, после последнего применения оператора RESET или CLEAR);

END TIME – абсолютное время, в момент, когда счетчик завершений принял значение 0;

BLOCKS – количество блоков, использованных в текущей модели к моменту завершения моделирования;

FACILITIES – количество блоков, использованных в модели к моменту завершения моделирования;

STORAGES – количество памятей, использованных в модели к моменту завершения моделирования;

FREE MEMORY – количество байтов памяти, доступной для дальнейшего использования.

Затем в файле статистики следует информация об именах, которые просматривает GPSS/PC в ходе моделирования:

NAME    VALUE    TYPE

Поле NAME содержит имена, используемые в программе модели.

Поле VALUE определяет числовое значение (номер), соответствующее имени. Система устанавливает начальный номер равный 10000.

Поле TYPE равно 0, если значение имени устанавливает пользователь, равно 2, если значение имени устанавливает система; 3 – если имя является именем блока.

Далее описываются блоки текущей модели в виде:

LINE    LOC    BLOCK TYPE    ENTRY COUNT    CURRENT COUNT  
RETRY

Поля в записи имеют следующее значение:

LINE – номер строки в рабочей модели, связанный блоком GPSS/PC;

LOC – имя или номер этого блока;

BLOCK TYPE – тип блока GPSS/PC;



ENTRY COUNT – количество транзактов, вошедших в данный блок, после последнего выполнения блоков RESET или CLEAR, или с начала работы программы модели;

CURRENT COUNT – количество транзактов, ожидающих специальных условий, зависящих от состояния данного блока.

Если в модели используются объекты типа «устройство», то далее в файле статистики идет информация об этих объектах:

FACILITY ENTRIES UTIL. AVE.TIME AVAILABLE OWNER  
REND INTER RETRY DELAY

где: FACILITY – номер или имя объекта типа «устройство»;

ENTRIES – количество транзактов, занявших или прервавших после последнего выполнения операторов RESET ,CLEAR или с начала работы программы;

UTIL. – часть периода моделирования, в течении которого устройство было занято;

AVE.TIME – среднее время занятости устройства одним транзактом в течении периода моделирования после последнего выполнения операторов RESET или CLEAR;

AVAILABLE – состояние готовности устройства в конце периода моделирования;

OWNER – номер последнего транзакта, занимавшего устройство (0 означает, что устройство не занималось);

REND – количество транзактов, ожидающих устройство (находящееся в режиме прерывания);

INTER – количество транзактов, обработка которых прервана на устройстве в данный момент модельного времени;

RETRY – количество транзактов, ожидающих специальных условий, зависящих от состояния предмета типа «устройство»;

DELAY – количество транзактов, ожидающих занятия устройства (включая транзакты, ожидающие выхода устройства на режиме прерывания).

В случае использования в модели объектах типа «очередь», далее следует информация об этих объектах. Например,

QUEUE MAX CONT ENTRIES ENTRIES (0) AVE.CONT  
AVE.TIME AVE. ( – 0) RETRY

где: QUEUE – имя или номер объекта типа «очередь»;

MAX – максимальное содержание объекта типа «очередь» в течении периода моделирования, который начинается с начала работы модели или с последнего оператора RESET или CLEAR;

CONT – текущее содержимое объекта типа «очередь» в момент завершения моделирования;

ENTRIES – общее количество входов в очередь в течении периода моделирования (счетчик входов);

ENTRIES (0) – общее количество входов в очередь с нулевым временем ожидания (счетчик «нулевых» входов);

AVE.CONT – среднее значение содержимого очереди;

AVE.TIME – среднее время, проведенное транзактом в очереди с учетом всех входов в очередь;

AVE. ( – 0) – среднее время проведенное транзактом в очереди без учета «нулевых» входов в очередь;

RETRY – количество транзактов, ожидающих специальных условий, зависящих от состояния объекта типа «очередь»;

Если в модели использовались объекты типа «память», то далее в файле статистики идет информация об этих объектах. Например,

STORAGE CAP. REMAIN MIN MAX ENTRIES AVL. AVE. C.  
UTIL. RETRY DELAY

где: STORAGE – имя или номер объекта типа «память»;

CAP. – объем памяти, заданного оператором STORAGE;

REMAIN – число единиц свободного объема памяти в конце периода моделирования;

MIN – минимальное количество используемых единиц памяти в конце периода моделирования;

MAX – максимальное количество используемых единиц памяти за период моделирования;

ENTRIES – количество входов в память за период моделирования;

AVL. – состояние готовности памяти в конце периода моделирования;

AVE. C. – среднее число занятых единиц памяти за период моделирования;

UTIL. – часть периода моделирования в течение которого память использовалась;

RETRY – количество транзактов, ожидающих специальных условий, зависящих от состояния памяти;

DELAY – количество транзактов, ожидающих возможности входа в блок INTER.

Если в модели используются блоки TABLE или QTABLE в файле статистики будет предоставлена информация о таблицах:

TABLE MEAN STD.DEV. RETRY RANGE FREQUENCY CUM.%

где: TABLE – имя или номер объекта типа «таблица» или «q – таблица»;

MEAN – выборочное среднее значение табулируемого аргумента. Значение преобразовывается в формат двойной точности при выводе в файл статистики;

STD.DEV. – выборочное среднеквадратичное отклонение;

RETRY – количество транзактов, ожидающих выполнения специальных условий, зависящих от состояния объекта типа «таблица»;

RANGE – верхние и нижние границы интервалов разбиения числовой оси;

FREQUENCY – величина  $N$ , пропорциональная частоте ( $n$ ) попаданий значений аргумента таблицы в указанный интервал:  $N = kn$ , где  $k$  – значение поля в соответствующих блоках TABULATE или DEPART (если  $N=0$ , то интервал пропускается);

CUM.% – процент попаданий значений аргумента таблицы в интервал от  $-\infty$  до  $x$  (где  $x$  – верхняя граница рассматриваемого интервала);

Информация об используемых в модели списках пользователей имеет вид:

USER CHAIN CHAIN SIZE RETRY AVE.CONT ENTRIES MAX  
AVE.TIME

где: USER CHAIN – номер (имя) объекта типа «список пользователя»;

CHAIN SIZE – количество транзактов в списке пользователя в конце периода моделирования;

RETRY – количество транзактов, ожидающих наступления специальных условий, связанных с состоянием списка пользователя типа «список пользователя»;

AVE.CONT – среднее содержимое списка пользователя в течение периода моделирования;

ENTRIES – общее количество транзактов, помещенных в список пользователя в течение периода моделирования;

MAX – максимальное количество транзактов в списке пользователя за период моделирования;

AVE.TIME – среднее время пребывания транзакта в списке пользователя.

Информация о ячейках памяти имеет вид:

SAEVALUEVALUE RETRY

где: SAEVALUE – имя или номер ячейки;

VALUE – значение ячейки в конце моделирования;

RETRY – количество транзактов, ожидающих наступления специальных условий, зависящих от состояния ячейки.

# Глава 5. Задания лабораторного практикума по GPSS

## ***ЗАДАНИЕ 1. Моделирование процесса функционирования накопителя на магнитных дисках***

### Исходные данные

1. Накопитель состоит из одного канала и трех дисководов.
2. Запросы поступают к каждому из дисководов равновероятно; длины временных промежутков между запросами случайны, независимы и распределены по экспоненциальному закону  $E(\mu)$ .
3. Обработка запроса включает установку головки (при этом канал не требуется и обмен данными через канал (запись или считывание).
4. Время установки головки - случайная величина с равномерным распределением  $R[0, \tau_+]$ . Время обмена данными равно  $\tau$ .

Цель. Разработать GPSSV - модель для анализа функционирования накопителя на магнитных дисках на временном промежутке  $[0, T]$ , взяв за единицу времени величину  $\tau$

Первоначальный перечень экспериментов:  $\tau=1.7$  мс,  $\tau_+=50$  мс,  $\mu=0.1$  мс<sup>-1</sup>

## ***ЗАДАНИЕ 2. Моделирование процесса продажи авиабилетов***

### Исходные данные

1. В помещении аэропорта работают  $m$  кассиров по продаже билетов, к каждому из которых образуется отдельная очередь. Время между моментами прибытия пассажиров - случайная величина с экспоненциальным распределением  $E(\tau_1)$ .

2. Пассажир выбирает самую короткую очередь. Время обслуживания пассажира кассиром - случайная величина с равномерным распределением  $R[\tau_-, \tau_+]$ .

3. В очереди к кассиру может находиться не более  $n$  пассажиров. Если длины очередей по всем кассирам равны максимальному значению  $n$ , то пассажир следует прямо на посадку и покупает билет у стюардессы.

Цель. Разработав GPSSV - модель для оценки среднего времени  $W_1$ , которое пассажир затрачивает на покупку билета; среднюю загрузку  $W_2$ , билетных касс; среднюю длину очереди по результатам имитационного моделирования процесса на  $[0, T]$ .

Первоначальный перечень экспериментов:  $m=5$ ,  $n=2$ ,  $\tau_1=5$  мин.,  $\tau_-=0.5$  мин.,  $\tau_+=1.5$  мин.,  $T=300$  мин.

## ***ЗАДАНИЕ 3. Моделирование системы массового технического контроля и настройки***

### Исходные данные

1. Система массового технического контроля и настройки телевизоров состоит из склада 1, (хранящего все ожидающие контроля телевизоры), пункта контроля из  $m$  параллельно работающих контролеров, склада 2 (хранящего все ожидающие настройки телевизоры) и пункта настройки из  $n$  параллельно работающих настройщиков.

2. Случайное время между поступлениями телевизоров в склад 1 для заключительной проверки, имеет равномерное распределение  $R[\tau_1, \tau_2]$ .

3. Из склада 1 телевизоры без задержки поступают к контролерам. Случайное время, необходимое на проверку одного телевизора, распределено по равномерному закону  $R[\tau_3, \tau_4]$ . В среднем 35 % телевизоров проходят проверку успешно с первого предъявления и направляются в пункт упаковки. Остальные 15 % не проходят проверку и направляются через склад 2 в пункт настройки.

4. Случайное время настройки имеет равномерное распределение  $R[\tau_5, \tau_6]$ .

Цель. Разработать GPSSV - модель и проимитировать функционирование системы массового технического контроля и настройки на отрезке времени  $[0, T]$  для оценки среднего времени, затрачиваемого, на обслуживание одного телевизора, а также среднего размера очереди в складах 1,2.

Первоначальный перечень экспериментов:  $m=2, n=1, \tau_1=3$  мин.,  $\tau_2=7$  мин.,  $\tau_3=6$  мин,  $\tau_4=12$  мин,  $\tau_5=20$  мин,  $\tau_6=40$  мин.

#### ***ЗАДАНИЕ 4. Моделирование производственного процесса на станке с поломками***

##### Исходные данные

1. Подлежащие обработке детали поступают на станок в среднем  $m$  раз в час; распределение вероятностей случайной длины промежутка между моментами поступления деталей - экспоненциальное  $E(1/m)$ .

2. При нормальном режиме работы задания выполняются в порядке их поступления. Случайное время обработки детали на станке имеет гауссовское распределение  $N(\mu_2, \sigma_2^2)$ . Перед началом обработки детали производится настройка (подготовка) станка, случайная длительность которой имеет равномерное распределение  $R[\tau_-, \tau_+]$ . Деталь, обработанная на станке, покидает данный станок и направляется в другие отделы цеха.

3. Станок подвергается случайным поломкам, при которых обработка детали прерывается. Случайная длительность интервала между поломками имеет гауссовское распределение  $N(\mu_2, \sigma_2^2)$ . При поломке обрабатываемая деталь снимается со станка и помещается в начало имеющейся перед станком очереди. После ремонта станка обработка детали возобновляется с того места, на котором она была прервана.

4. Ремонт станка состоит из трех независимых последовательных фаз. Случайная длительность каждой фазы имеет экспоненциальное распределение  $E(\lambda)$ .

Цель. Разработать GPSSV - модель и осуществить имитацию производственного процесса на отрезке времени  $[0, T]$ , для оценки вероятности простоя станка, средней длины очереди к станку и среднего времени обработки детали.

Первоначальный перечень экспериментов:  $m=1$ ,  $\mu_1=0.5$  час,  $\sigma_1=0.1$  час,  $\tau_-=0.2$  час.,  $\tau_+=0.5$  час.,  $\mu_2=20$  час.,  $\sigma_2=2$  час.,  $\lambda=4/3$  (час.)<sup>-1</sup>,  $T=500$  час.  $\tau_4=12$  мин,  $\tau_5=20$  мин,  $\tau_6=40$  мин.

### ***ЗАДАНИЕ 5. Моделирование процесса обработки деталей***

#### Исходные данные

1. В обрабатывающий цех поступают детали, обработка которых осуществляется двумя станками.
2. Детали, поступающие на первичную обработку, образуют пуассоновский поток со средним интервалом между поступлением  $\tau$  мин.
3. Первый станок обрабатывает деталь в среднем  $t_1$  мин. и имеет  $a_1$  % брака, второй - соответственно  $t_2$  мин. и  $a_2$  % брака. Все бракованные детали возвращаются на повторную обработку на первый станок. Интервалы времени обработки обоих станков имеют экспоненциальное распределение.

Цель. Построить GPSSV - модель процесса обработки для  $n$  деталей. Определить среднюю загрузку первого станка на вторичной обработке и вероятность появления отходов (деталей, попавших в раздел бракованных после вторичной обработки).

Первоначальный перечень экспериментов:  $\tau=50$ ,  $t_1=40$ ,  $t_2=60$ ,  $a_1=4$  %,  $a_2=8$  %,  $n=500$ .

### ***ЗАДАНИЕ 6. Моделирование системы обработки информации***

#### Исходные данные

1. Система обработки сигналов, поступающих с датчиков, оснащена мультиплексным каналом и тремя мини-ЭВМ.
2. Сигналы от датчиков поступают на вход канала через интервалы времени  $\mu_b \pm \Delta\mu_b$ .
3. В канале сигналы буферизуются (образуют очередь) и претерпевают первичную обработку в течение  $\mu_1 \pm \Delta\mu_1$  мкс.
4. Каждая мини-ЭВМ имеет входной накопитель, емкостью  $L$  единиц. После первичной обработки сигналы поступают на обработку в ту мини ЭВМ, где имеется наименьшая по длине очередь и обрабатывается со временем  $\mu_2 \pm \Delta\mu_2$  мкс.

Цель. Разработать GPSSV - модель обработки и сигналов. При этом определить среднее время задержки сигналов в канале и мини ЭВМ и вероятность переполнения входных накопителей.

Первоначальный перечень экспериментов:  $L=10$ ,  $\mu_b=10$ ,  $\Delta\mu_b=5$ ,  $\mu_1=10$ ,  $\Delta\mu_1=3$ ,  $\mu_2=30$ ,  $\Delta\mu_2=3$ ,  $n=250$ .

### ***ЗАДАНИЕ 7. Моделирование работы конвейера сборочного цеха***

#### Исходные данные

1. Сборочный цех, содержащий комплектовочный конвейер, состоящий из секций емкостью по  $L$  изделий каждого из двух типов.
2. На конвейер для сборки поступают каждые  $\mu_1 \pm \Delta\mu_1$  мин.  $a_1$  изделий первого типа и каждые  $\mu_2 \pm \Delta\mu_2$  мин. –  $a_2$  изделий второго типа.

3. Комплектация начинается только при условии наличия деталей обоих типов в требуемом количестве и длится  $t$  мин. При нехватке деталей секция конвейера остается пустой.

Цель. Построить-GPSSV - модель работы сборочного конвейера в течение одной смены ( $\tau$  часов). Определить вероятность пропуска секции, средние очереди по каждому типу изделий. Определить экономическую целесообразность перехода на секции с характеристиками:  $a_2$  изделий каждого типа и временем комплектации  $2t$  мин.

Первоначальный перечень экспериментов:  $L=10$ ,  $\mu_1=5$ ,  $\Delta\mu_1=1$ ,  $\mu_2=2$ ,  $\Delta\mu_2=7$ ,  $a_1=5$ ,  $a_2=20$ ,  $t=10$ ,  $\tau=7$ .

### ***ЗАДАНИЕ 8 Моделирование функционирования вычислительной системы (ВС)***

#### Исходные данные

1. Специализированная ВС, оснащенная тремя процессами (ПП) и общей оперативной памятью (ОП), обеспечивает обработку задач пользователей.

2. Через интервалы времени  $\mu_B \pm \Delta\mu_B$  мин. поступают задания, занимающие объем ОП размером  $L_n$  кб.

3. После трансляции первым процессором - в течение  $\mu_1 \pm \Delta\mu_1$  мин. их объем достигает  $L_2$  кб и они поступают в ОП.

4. Далее задания редактируются вторым процессором в течении  $\mu_2 \pm \Delta\mu_2$  мин. на  $L_2$  кб. каждый, объем задания достигает  $L_r$  кб.

5. После редактирования задания через ОП поступают в третий процессор на выполнение, требующее  $\mu_3 \pm \Delta\mu_3$  мин. времени на каждый  $L_3$ .кб. и покидают систему, минуя ОП.

Цель. Разработать GPSSV - модель ВС в течение  $\tau$  час. Определить статистические характеристики занятия СП по всем видам заданий

Первоначальный перечень экспериментов:  $L_r=30$ ,  $\mu_3=1.5$ ,  $\Delta\mu_3=0.4$ ,  $L_3=10$ ,  $\tau=24$ ,  $\mu_B=5$ ,  $\Delta\mu_B=2$ ,  $L_n=10$ ,  $\mu_1=5$ ,  $\Delta\mu_1=1$ ,  $L_1=20$ ,  $\mu_2=25$ ,  $\Delta\mu_2=0.5$ ,  $L_2=10$ .

### ***ЗАДАНИЕ 9. Моделирование студенческого машинного зала ВЦ***

#### Исходные данные

1. Студенческий машинный зал, оснащенный двумя мини ЭВМ и одним устройством подготовки данных (УПД), обслуживает задания студентов.

2. Студенты приходят с интервалом  $\mu_B \pm \Delta\mu_B$  мин. и треть из них требует использования УПД и ЭВМ, а остальные - только ЭВМ.

3. Допустимая очередь в машинном зале составляет  $q$  человека, включающего работающего на УПД, работа которого занимает  $\mu_1 \pm \Delta\mu_1$  мин. Работа ЭВМ длится  $t$  мин.

4.  $a\%$ , работавших на ЭВМ, возвращаются для повторного использования УПД и ЭВМ.

Цель. Построить-GPSSV - модель работы машинного зала в течение  $t$  часов. Определить среднюю загрузку УПД и ЭВМ и вероятность отказа в обслуживании.

Первоначальный перечень экспериментов:  $\mu_B=10$ ,  $\Delta\mu_B=2$ ,  $a=40$ ,  $\mu_1=5$ ,  $\Delta\mu_1=2$ ,  $t=15$ ,  $a=20$ ,  $\tau=24$ .

### ***ЗАДАНИЕ 10. Моделирование передачи речинформационной цифровой системой***

#### Исходные данные

1. Информационная система, состоящая из 2 транзитных каналов с входным накопителем, осуществляет передачу речи в цифровом виде.
2. Время передачи цифрового пакета по каждому каналу составляет  $t$  мс, интервал поступления составляет  $\mu_B \pm \Delta\mu_B$  мс.
3. Пакеты, длительность передачи которых превышает  $t_n$  мс, подлежат уничтожению, так как сильно искажают качество речи, однако более  $a\%$  уничтожения - недопустимо.

Цель. Смоделировать средствами GPSSV работу описанной системы. Оценить частоту уничтожения пакетов.

Первоначальный перечень экспериментов:  $t=5$ ,  $\mu_B=6$ ,  $\Delta\mu_B=3$ ,  $t_n=10$ ,  $a=30$ .

### ***ЗАДАНИЕ 11. Моделирование обслуживания запросов банком данных***

#### Исходные данные

1. На базе ЭВМ, соединенных дуплексным каналом связи, организован распределенный банк данных, служащий для обработки запросов.
2. Поступающий с интервалом  $\mu_B \pm \Delta\mu_B$  запрос обрабатывается на первой ЭВМ и с вероятностью  $a\%$  необходимая информация будет обнаружена. В противном случае запрос посылается на вторую ЭВМ.
3. Первичная обработка запроса занимает  $t_1$  с., а выдача ответа требует  $\mu_1 \pm \Delta\mu_1$  с., передача по каналу связи требует  $t_c$  времени.

Цель. Осуществить с помощью GPSSV - модели продвижение  $n$  запросов. Определить необходимую емкость накопителей перед ЭВМ для безотказной работы системы и функцию распределения случайного времени обслуживания заявки.

Первоначальный перечень экспериментов:

### ***ЗАДАНИЕ 12. Моделирование процесса функционирования участка механообработки ГАП***

#### Исходные данные

1. Участок механообработки включает входной конвейер, транспортный робот, два однотипных независимо работающих станка с ЧПУ и выходной конвейер.
2. Время поступления деталей на обработку и время обработки детали на станке распределены равномерно на интервалах  $[a \pm \delta]$  сек.,  $[b \pm \delta]$  сек. соответственно.
3. Робот переносит деталь с входного конвейера на свободный станок и затем со станка на выходной конвейер, время каждого переноса распределено равномерно на интервале  $[c \pm v]$  сек.



Цель. Разработать GPSSV - модель для анализа процесса функционирования участка в течение  $k$  смен (продолжительность одной смены 8 часов ).

Первоначальный перечень экспериментов :  $k=1$ ,  $a=160$ ,  $\delta=90$ ,  $b=280$ ,  $\varepsilon=30$ ,  $c=10$ ,  $v=2$

### **ЗАДАНИЕ 13. Моделирование участка механообработки ГАП**

#### Исходные данные

1. Участок механообработки включает рабочее место комплектации пакета, два станка для обработки деталей и транспортный робот (робокар).

2. Имеется два типа деталей. Детали 1-го и 2-го типов поступают на место комплектации пакета соответственно через  $t_1$  сек. и  $t_2$  сек. Каждый пакет вмещает  $m_1$  деталей типа 1 либо  $m_2$  деталей типа 2.

3. Пакеты с деталями типа 1,2 обрабатываются на станках 1,2 соответственно. Время обработки распределено равномерно для станка 1 на интервале  $[a_1 \pm \delta_1]$  сек, для станка 2 на интервале  $[a_2 \pm \delta_2]$ .

4. Робокар транспортирует по две пакеты от места комплектации до станков и обратно в течение времени, равномерно распределенного на интервале  $[b \pm \varepsilon]$ .

5. На месте комплектации пакеты освобождаются от обработанных деталей, загружаются снова и поступают на обработку.

Цель. Разработать GPSSV - модель анализа процесса функционирования участка в течение  $k$  часов.

Первоначальный перечень экспериментов  $k=1$ ,  $t_1=15$ ,  $t_2=10$ ,  $a_1=120$ ,  $\delta_1=10$ ,  $a_2=40$ ,  $\delta_2=5$ ,  $b=16$ ,  $\varepsilon=4$ .

### **ЗАДАНИЕ 14. Моделирование производственного участка с двумя обрабатывающими центрами**

#### Исходные данные

1. Производственный участок имеет два обрабатывающих центра (ОБРЦ-1 и ОБРЦ-2), которые осуществляют механообработку деталей трех типов.

2. Детали поступают на обработку через каждые  $t$  мин, образуя пуассоновский поток. В этом потоке детали типов 1 и 2 появляются с вероятностями 0,25, детали типа 3 - с вероятностью 0,5.

3. Время обработки деталей типов 1 и 2 –  $\tau_1$  мин, деталей типа 3 –  $\tau_2$  мин.

4. ОБРЦ-1 осуществляет обработку деталей типов 1 и 2, а ОБРЦ-2 - деталей всех типов, причем детали типов 1 и 2 поступают в ОБРЦ-2, в случае, если ОБРЦ-1 занят.

5. На участке Фиксируется количество обработанных деталей каждого типа.

Цель работы. Разработать GPSSV - модель для анализа процесса функционирования производственного участка в течение одной смены. т.е.  $\varepsilon$  часов.

Первоначальный перечень экспериментов:  $t=6$ ,  $\tau_1=10$ ,  $\tau_2=8$ .

### **ЗАДАНИЕ 15. Моделирование автоматизированной технологической линии**

#### Исходные данные

1. Технологическая линия включает источник деталей, два взаимосвязанных станка, накопитель, технологический модуль для окончательной обработки деталей, рабочее место комплектации пакет и транспортный робот для их транспортировки на склад.

2. Время поступления деталей из источника распределено равномерно на интервале  $[a \pm \delta]$  сек, причем деталь поступает в минимальную из очередей к станкам.

3. Если деталь поступает на станок I, то затем она поступает на станок 2. Если деталь поступает на станок 2, то затем она поступает на станок I. Время работы деталей на станках 1,2 распределено равномерно на интервалах  $[b_1 \pm \varepsilon_1]$  сек,  $[b_2 \pm \varepsilon_2]$  сек. соответственно.

4. После цикла механообработки деталь попадает в накопитель (на 10 деталей). Из накопителя все детали одновременно передаются в технологический модуль для окончательной обработки  $[b_3 \pm \varepsilon_3]$  сек. Затем осуществляется укладка деталей в пакеты по 10 шт. Транспортом робот отбирает по 2 пакеты и транспортирует их на склад. Время транспортировки распределено равномерно на интервале  $[c \pm v]$  сек.

Цель. Разработать GPSSV - модель для анализа процесса функционирования автоматизированной технологической линии в течение часа.

Первоначальный перечень экспериментов:  $a=10$ ,  $\delta=2$ ,  $b_1=10$ ,  $\varepsilon_1=4$ ,  $b_2=9$ ,  $\varepsilon_2=3$ ,  $b_3=6$ ,  $\varepsilon_3=2$ ,  $c=16$ ,  $v=4$ .

### ***ЗАДАНИЕ 16. Моделирование процесса функционирования централизованной вычислительной системы***

#### Исходные данные

1. Вычислительная система содержит три однотипных процессора с общей памятью, разделенной на  $n$  блоков (каждой задаче при ее решении выделяется один блок).

2. Задачи пользователей образуют пуассоновский поток с интенсивностью  $\lambda$  зад/сек. Задачи остаются в накопителе, если длина очереди на выполнение не превышает  $n$  задач.

3. Задачи пользователей могут быть типа I (25 %, высший приоритет) и типа 2 (75 %, низший приоритет). Время решения задачи в каждом процессоре имеет экспоненциальное распределение с математическими ожиданиями соответственно  $\mu_1$  и  $\mu_2$  сек.

4. С вероятностью 0,6 задача решается и покидает систему. В остальных случаях задача поступает на обслуживание в блоки (время обслуживания распределено равномерно на интервале  $[a \pm \delta]$  сек.) внешней памяти и возвращается в очередь на повторное решение.

Цель. Разработать GPSSV - модель для анализа процесса функционирования вычислительной системы в течение часа. Первоначальный перечень экспериментов:  $n=10$ ,  $\lambda=0.25$ ,  $\mu_1=2$ ,  $\mu_2=5$ ,  $a=5$ ,  $\delta=3$ .

**Задание 17.** Моделирование процесса функционирования распределенной вычислительной системы.

Исходные данные:

1. РВС имеет архитектуру с общей шиной, к которой подключены 10 терминалов пользователей, 3 однотипных процессора, 2 магнитных диска и 1 АЦПУ.
2. Задачи пользователей образуют пуассоновский поток с интенсивностью  $\lambda$  зад/сек, а время обработки задачи в процессоре имеет экспоненциальное распределение с математическим ожиданием  $\mu$  сек.
3. Время обслуживания задачи на магнитном диске  $[85 \pm 25] \cdot 10^{-3}$  сек, в АЦПУ- $[a \pm \delta]$  сек. Время “захвата” шины одной задачей  $25 \cdot 10^{-3}$  сек. Во всех случаях имеет место равномерное распределение.
4. С вероятностью  $p$  задача решается в процессорном блоке и возвращается к пользователю. В остальных случаях задача поступает на обслуживание в блоки внешней памяти и возвращается в очередь на повторное решение. Примерно  $q$  решенных задач проходят обслуживание а АЦПУ.

Цель: разработать GPSSV- модель для анализа процесса функционирования РВС в течение одного часа.

Первоначальный перечень экспериментов:  $\lambda=0$ ,  $\mu=5$ ,  $a=12$ .  $p=0.6$ ,  $q=5$ .

Задание 18. Моделирование вычислительной системы с удаленными терминалами.

Исходные данные:

1. Вычислительная система представляет собой двухпроцессорный комплекс, который обслуживает местных пользователей и три однотипных удаленных терминала.
2. На каждом терминале задача формируется в среднем через  $t$  сек, а время выполнения задачи в процессорном блоке имеет математическое ожидание  $\mu$  сек (экспоненциальное распределение).
3. После выполнения задача возвращается на соответствующий терминал, инициируя тем самым формирование новой задачи.
4. Время передачи данных по каналу связи распределяется равномерно в пределах от  $a$  до  $b$  сек.

Цель. Разработать GPSSV- модель для анализа процесса функционирования вычислительной системы в течение одного часа.

Первоначальный перечень экспериментов:  $t=25$ ,  $\mu=20$ ,  $a=10$ ,  $b=30$ .

Задание 19. Моделирование вычислительной системы коллективного пользования.

Исходные данные:

1. Вычислительная система имеет две разнотипные ЭВМ (ЭВМ-1 и ЭВМ-2) которые обслуживают сеть активных терминалов.
2. Задачи пользователей образуют пуассоновский поток с интенсивностью  $\lambda$  зад/сек, а время выполнения задачи в ЭВМ имеет экспоненциальное распределение с математическими ожиданиями  $\mu_1$  сек (ЭВМ-1) и  $\mu_2$  сек (ЭВМ-2).
3. Задачи пользователей выполняются в мультипрограммном режиме, причем область памяти каждой ЭВМ разделяет на  $n$  блоков.

5. Если поступившая задача застаёт ЭВМ занятой, то она направляется в ЭВМ-2.
6. После выполнения в ЭВМ 25 % всех задач обслуживается в АЦПУ, причём время распечатки одного листинга распределено равномерно на интервале  $[a \pm \varepsilon]$  сек.

Цель: Разработать GPSSV-модель для анализа процесса функционирования вычислительной системы в течение одного часа.

Первоначальный перечень экспериментов:  $\lambda=0.2$ ,  $\mu_1=8$ ,  $\mu_2=12$ ,  $n=10$ ,  $a=12$ ,  $\varepsilon=8$

*Задание 20.* Моделирование процесса функционирования вычислительного центра.

Исходные данные:

1. Вычислительный центр, оснащенный тремя однотипными ЭВМ, обслуживает сеть активных терминалов.
2. Задачи пользователей образуют пуассоновский поток с  $\lambda$  зад/сек, а время выполнения задачи в ЭВМ имеет экспоненциальное распределение с математическим ожиданием  $\mu$  сек.
3. Программа-диспетчер обрабатывает задачу, выбирая для нее свободную ЭВМ. Время обработки равномерно распределено на интервале  $[a \pm \delta]$ . Если все ЭВМ заняты, то задача направляется в очередь, которая на данный момент является минимальной.
4. После выполнения в ЭВМ, задача возвращается на соответствующий терминал, причем 30 % задач обслуживается в АЦПУ  $[b \pm \varepsilon]$  сек.

Цель: Разработать GPSSV- модель для анализа процесса функционирования вычислительного центра в течение одного часа.

Первоначальный перечень экспериментов:  $\lambda=0.2$ ,  $\mu=12$ ,  $a=2$ ,  $\delta=1$ ,  $b=12$ ,  $\varepsilon=8$ .

*Задание 21.* Моделирование роботизированного трехканального обрабатывающего центра.

Исходные данные:

1. Обрабатывающий центр включает промышленный робот, который распределяет поступающие детали в три однотипных канала их механообработки.
2. Детали, поступающие в обрабатывающий центр, образуют поток Эрланга 2-ого порядка с интенсивностью  $\lambda$  дет/сек, а время механообработки детали имеет экспоненциальное распределение с математическим ожиданием  $\mu$  сек.
3. Робот обслуживает деталь  $[a \pm \delta]$  сек (равномерный закон распределения) распределяя ее в свободный канал. Если все каналы заняты, то деталь направляется в очередь, которая на данный момент является минимальной.

Цель: Разработать GPSSV- модель для анализа процесса функционирования обрабатывающего центра в течение одного часа.

Первоначальный перечень экспериментов:  $\lambda=0.1$ ,  $\mu=25$ ,  $a=3$ ,  $\delta=1$ .

**Задание 22.** Моделирование процесса функционирования участка сборки.

Исходные данные:

1. Участок сборки включает источник кассет со сборочным материалом, два сборочных модуля для изготовления изделий 2-ух типов и конвейер для их транспортировки на склад.
2. Кассеты поступают на сборку в среднем через  $t$  мин, образуя пуассоновский поток. А этом потоке 75 % кассет типа 1 и 25 % - типа 2.
3. При сборке изделия сборочный модуль 1 выполняет последовательно  $m_1$  операции, а сборочный модуль 2 –  $m_2$  операций причем каждая операция сборки занимает в среднем  $\tau$  мин. (экспоненциальное распределение).
4. Контроль качества и настройка изделия типа 1 занимают  $[a \pm \delta]$  мин, изделия типа 2 –  $[b \pm \varepsilon]$  мин (равномерное распределение). После контроля и настройки изделия поступают на конвейер, который транспортирует их на склад. Время транспортировки имеет равномерное распределение на интервале  $[c \pm v]$  мин.
5. В модели табулируется время сборки изделий обоих типов (от момента поступления кассеты на участок до момента появления изделия на складе).

Цель. Разработать GPSSV- модель для анализа процесса функционирования участка сборки в течение смены, т. е. 8-ми часов.

Первоначальный перечень экспериментов:  $t=6$ ,  $m_1=2$ ,  $m_2=3$ ,  $\tau=5$ ,  $a=5$ ,  $\delta=2$ ,  $b=9$ ,  $\varepsilon=3$ ,  $c=4$ ,  $v=1$ .

**Задание 23.** Моделирование процесса сборки изделий.

Исходные данные:

1. На участок сборки поступают кассеты со сборочным материалом в среднем через  $t$  сек, образуя пуассоновский поток. Участок реализует последовательную схему сборки изделий круговой компоновкой трех сборочных модулей.
2. Сборочный модуль 1 выполняет  $m_1$  сборочных операций, затрачивая на операцию в среднем  $\tau_1$  сек (экспоненциальное распределение).
3. Сборочный модуль 2 выполняет  $m_2$  сборочных операций, затрачивая на операцию в среднем  $\tau_2$  сек (экспоненциальное распределение).
4. Сборочный модуль 3 выполняет  $m_3$  сборочных операций, затрачивая на операцию в среднем  $\tau_3$  сек (экспоненциальное распределение).
5. Сборка следующего изделия начинается после завершения всех операций в сборочном модуле 1.

Цель. Разработать GPSSV- модель для анализа вероятностных характеристик процессов сборки изделий по результатам имитации одной смены, т. е. 8-ми часов.

Первоначальный перечень экспериментов:  $t=90$ ,  $m_1=5$ ,  $m_2=3$ ,  $m_3=2$ ,  $\tau_1=10$ ,  $\tau_2=20$ ,  $\tau_3=30$ .

**Задание 24.** Моделирование процесса стендовых испытаний.

Исходные данные:

1. Объектом моделирования является процесс контроля качества изделий на 2-ух испытательных стендах.
2. Изделия могут быть типа 1 (40 %) и типа 2 (60 %). Изделия типа 1 проходят испытания на стенде 1, а изделия типа 2- на стенде 1и 2. Изделия типа 2 поступают на стенд 1 в случае, если занят стенд 2.
3. Изделия, поступающие на стендовые испытания, образуют пуассоновский поток с  $\lambda$  изд/мин., а время их обслуживания имеет экспоненциальное распределение.
4. Математическое ожидание времени обслуживания изделий типа 1- $\mu_1$  мин, изделий типа 2 на стенде 2 -  $\mu_2$  мин. И изделий типа 2 на стенде 1- $\mu_3$  мин.

Цель. Разработать GPSSV- модель для анализа процесса стендовых испытаний изделий в течение суток.

Первоначальный перечень экспериментов:  $\lambda=0.067$ ,  $\mu_1=15$ ,  $\mu_2=25$ ,  $\mu_3=20$ .

Задание 25. Моделирование автоматизированной системы контроля.

Исходные данные:

1. В систему контроля поступают готовые изделия, которые транспортируются по конвейеру к трем последовательно расположенным испытательным стендам.
2. Все испытательные стенды выполняют однотипные операции функционального контроля, причем время контроля изделия имеет равномерное распределение на интервале  $[a \pm \delta]$  мин.
3. Изделие поступает на конвейер через каждые  $t$  мин. и транспортируется  $\tau_1$  мин к стенду 1. Если стенд 1 занят, то изделие транспортируется  $\tau_2$  мин. к стенду 2. Если стенд 2 занят, то изделие транспортируется  $\tau_3$  мин. к стенду 3. Если стенд 3 занят, то изделие оказывается не обслуженным.
4. В модели табулируются временные характеристики процесса обслуживания изделия в системе контроля.

Цель. Разработать GPSSV- модель для анализа автоматизированной системы контроля в течение смены, т. е. 8-ми часов.

Первоначальный перечень экспериментов.  $a=12$ ,  $t=5$ ,  $\delta=9$ ,  $\tau_1=\tau_2=\tau_3=2$ .

Задание 26. Моделирование процесса функционирования ЭВМ с сетью АРМ.

Исходные данные:

1. ЭВМ обслуживает сеть из 3-ех АРМ (АРМ-1, АРМ-2, АРМ-3), обеспечивая мультипрограммный режим работы (область памяти разделяется на  $n$  блоков).
2. Задачи, поступающие с АРМ, образуют пуассоновские потоки с интенсивностями  $\lambda_1$  зад/сек. (АРМ-1),  $\lambda_2$  зад/сек. (АРМ-2),  $\lambda_3$  зад/сек. (АРМ-3), а время выполнения задачи в процессоре имеет экспоненциальное распределение с математическим ожиданием  $\mu$  сек.
3. После выполнения в ЭВМ 40 % всех задач обслуживается в АЦПУ, причем распечатка одного листинга занимает  $[a \pm \delta]$  сек.(равномерный закон распределения).

4. В модели фиксируется количество выполненных задач по каждому АРМ в отдельности.

Цель. Разработать GPSSV- модель для анализа процесса функционирования ЭВМ с сетью АРМ в течение одного часа.

Первоначальный перечень экспериментов.  $n = 10$ ,  $\lambda_1 = 0.05$ ,  $\lambda_2 = 0.04$ ,  $\lambda_3 = 0.01$ ,  $\mu = 9$ ,  $a = 12$ ,  $\delta = 8$ .

*Задание 27.* Моделирование процесса функционирования ЭВМ с учетом отказов.

Исходные данные:

1. Объектом моделирования является ЭВМ, которая обслуживает сеть активных терминалов в мультипрограммном режиме (область памяти разделяется на  $n$  блоков).
2. Задачи пользователей поступают на обслуживание в среднем через  $t$  сек, образуя пуассоновский поток. Время выполнения задачи в процессоре имеет экспоненциальное распределение с математическим ожиданием  $t$  сек.
3. Сбои ЭВМ или перегрузка операционной системы (отказы) происходят в среднем через час, а время восстановления имеет равномерное распределение в пределах от  $a$  до  $b$  мин.
4. В модели фиксируется количество отказов и предусматривается вывод результатов имитации в конце первого, второго, 3-его и 4-ого часов работы ЭВМ.

Цель. Разработать GPSSV- модель для анализа процесса функционирования ЭВМ с учетом отказов в течение 4-ех часов.

Первоначальный перечень экспериментов.  $n = 10$ ,  $t = 10$ ,  $\tau = 9$ ,  $a = 2$ ,  $b = 6$ .

*Задание 28.* Моделирование вычислительной системы с разделением времени.

Исходные данные:

1. Вычислительная система включает центральный процессор и  $m$  активных терминалов, обеспечивая режим разделения времени с квантовой дисциплиной обслуживания.
2. В 1-ой группе  $m_1$  терминалов, среднее время обдумывания  $\mu_1$  сек. (экспоненциальное распределение), квант обслуживания -  $\Delta t_1$  сек.
3. В 2-ой группе  $m_2$  терминалов, среднее время обдумывания  $\mu_2$  сек. (экспоненциальное распределение), квант обслуживания -  $\Delta t_2$  сек.
4. После обслуживания в процессоре 60 % задач возвращаются на терминалы, инициируя формирование новых задач. Остальные задачи поступают в очередь к процессору на повторное обслуживание.

Цель. Разработать GPSSV- модель для анализа процесса функционирования вычислительной системы в течение одного часа.

Первоначальный перечень экспериментов.  $m = 15$ ,  $m_1 = 3$ ,  $m_2 = 12$ ,  $\mu_1 = 10$ ,  $\mu_2 = 20$ ,  $\Delta t_1 = 0.9$ ,  $\Delta t_2 = 0.6$

*Задание 29.* Моделирование однопроцессорной вычислительной системы коллективного пользования.

Исходные данные:

1. Вычислительная система содержит центральный процессор и  $m$  активных терминалов, обеспечивая режим разделения времени с квантовой дисциплиной обслуживания.
2. Времена обдумывания и обслуживания имеют экспоненциальное распределение с мат. ожиданием соответственно  $\mu_1$  и  $\mu_2$  сек.
3. Поступающие в систему задачи обслуживаются в порядке очереди, причем если за квант времени задача не выполняется, то она возвращается в очередь на повторное обслуживание.
4. С вероятностью 0.6 задача решается и переходит в терминальный узел. В остальных случаях задача возвращается в очередь к центральному процессу.

Цель. Разработать GPSSV- модель для анализа процесса функционирования однопроцессорной вычислительной системы в течение одного часа.

Первоначальный перечень экспериментов.  $m=12, \mu_1=10, \mu_2=0.5$ .

Задание 30. Моделирование многопроцессорной вычислительной системы коллективного пользования.

Исходные данные:

1. Вычислительная система содержит 3 однотипных процессора с общей оперативной памятью и блоки внешней памяти на магнитных дисках.
2. Задачи пользователей образуют пуассоновский поток с интенсивностью  $\lambda$  зад/сек, а время решения задачи в каждом процессоре имеет экспоненциальное распределение с мат. ожиданием  $\mu$  сек.
3. Поступающие в систему задачи обслуживаются в порядке очереди без приоритетов.
4. С вероятностью 0.6 задача решается и покидает систему. В остальных случаях задача поступает на обслуживание в блоки внешней памяти и возвращается в очередь на повторное решение (время обслуживания распределено равномерно на интервале  $[a \pm \delta]$  сек.)

Цель. Разработать GPSSV- модель для анализа процесса функционирования многопроцессорной вычислительной системы по результатам имитации  $n$  задач.

Первоначальный перечень экспериментов.  $\mu=4, a=5, n=100, \delta=3, \lambda=0.5$ .

Задание 31. Моделирование сети автоматизированных рабочих мест.

Исходные данные:

1. Сеть АРМ организована на базе 3-ех персональных ЭВМ, выполняющих функции интеллектуальных терминалов. АРМ имеют доступ к центральной ЭВМ через коммуникационный процессор.
2. На каждом терминале задача формируется в среднем через  $t$  сек. (экспоненциальное распределение), причем на выполнение в центральную ЭВМ идет 25 % задач. Остальные задачи выполняются в автономном режиме.



3. Время обслуживания задачи в коммуникационном процессоре и центральной ЭВМ имеет экспоненциальное распределение с мат. ожиданием  $\mu_1$  и  $\mu_2$  сек.
4. После выполнения в центральной ЭВМ задача возвращается через коммуникационный процессор на соответствующий терминал, инициируя тем самым формирование новой задачи.

Цель. Разработать GPSSV- модель для анализа процесса функционирования сети АРМ в течение одного часа.

Первоначальный перечень экспериментов.  $t=10$ ,  $\mu_1=1$ ,  $\mu_2=12$ .

**Задание 32.** Моделирование участка по производству печатных плат.

Исходные данные:

1. На участке функционирует 3 технологических модуля, которые производят печатные платы.
2. За смену модуль 1 производит  $[a_1 \pm \delta]$  плат (равномерный закон распределения).
3. Вероятность изготовления качественной платы в модуле 1-0.8, а в модуле 2-0.9 и в модуле 3- 0.95.
4. Производительность участка определяется общим числом качественных плат, изготовленных всеми модулями за смену.

Цель. Разработать GPSSV- модель для анализа вероятностных характеристик производительности участка по результатам  $k$  прогонов модели.

Первоначальный перечень экспериментов:  $k=100$ ,  $a_1=125, \delta_1=25$ ,  $a_2=115, \delta_2=15$ ,  $a_3=110, \delta_3=10$ .

**ЗАДАНИЕ 52.** Моделирование процесса функционирования технологического модуля

Исходные данные:

1. Объектом моделирования является технологический модуль (ТМ), который осуществляет параллельную обработку деталей двух типов.
2. Детали типа 1 поступают на ТМ через каждые  $a_1 \pm \alpha_1$  сек., а детали типа 2 - через каждые  $a_2 \pm \alpha_2$  сек., образуя общую очередь.
3. Детали типа 1 обрабатываются  $b_1 \pm \beta_1$  сек, а детали типа 2 -  $b_2 \pm \beta_2$  сек. Через каждый час осуществляется смена инструмента на обоих станках в течение  $t$  сек.
4. После окончания обработки детали покидают ТМ, причем фиксируется общее число деталей каждого типа.

Цель. Разработать GPSSV – модель для анализа процесса функционирования ТМ по результатам имитации одной смены, т.е. 8 часов.

Первоначальный перечень экспериментов:

$a_1 = 180$ ,  $\alpha_1 = 60$ ,  $a_2 = 360$ ,  $\alpha_2 = 120$

$b_1 = 180$ ,  $\beta_1 = 180$ ,  $b_2 = 360$ ,  $\beta_2 = 60$

$t = 20$

**ЗАДАНИЕ 53. Моделирование процесса функционирования механического цеха**

Исходные данные:

1. Объектом моделирования является механический цех, который оборудован тремя универсальными станками для обработки пяти типов деталей.
2. Поступающие в цех детали образуют пуассоновский поток с  $\lambda$  дет/мин, а тип деталей в этом потоке задается распределением

Тип деталей	1	2	3	4	5
$p_i$	0.1	0.2	0.3	0.35	0.05

3. Время обработки детали на станке имеет экспоненциальное распределение с математическим ожиданием  $\mu$  мин.
4. После окончания обработки детали покидают цех, причем фиксируется общее число деталей каждого типа.

Цель. Построить GPSSV – модель для анализа процесса функционирования механического цеха по результатам имитации одной смены, т.е. 8 часов.

Первоначальный перечень экспериментов:

$$\lambda = 0.8, \quad \mu = 4$$

**ЗАДАНИЕ 54. Моделирование процесса функционирования промышленного робота**

Исходные данные:

1. Поступающие к роботу изделия образуют пуассоновский поток с  $\lambda$  изд./сек, а время обработки изделия составляет  $t$  сек.
2. Вероятность безотказной работы робота за время обработки изделия равна 0,95. Если робот отказывает, то происходит его восстановление по экспоненциальному закону с математическим ожиданием  $\mu$  сек. После восстановления робота обработка изделия продолжается, причем фиксируется общее число отказов робота.

Цель. Построить GPSSV – модель для анализа процесса функционирования промышленного робота по результатам имитации  $n$  изделий.

Первоначальный перечень экспериментов:

$$\lambda = 0.02, \quad t = 45, \quad \mu = 180, \quad n = 100.$$

**ЗАДАНИЕ 55. Моделирование процесса функционирования робототехнического комплекса**

Исходные данные:

1. РТК содержит 3 технологических модуля:  $TM_1$ ,  $TM_2$ ,  $TM_3$ . Причем  $TM_1$  и  $TM_2$  начинают работу одновременно, а  $TM_3$  – после завершения первых двух операций.
2. Длительность технологических операций  $t_1 = t_2 = 10$  мин.,  $t_3 = 20$  мин.
3. Вероятности безотказной работы оборудования при выполнении операций и математические ожидания процессов восстановления

$$p_1 = 0.8, \quad m_1 = 2 \text{ мин}$$

$$p_2 = 0.9, \quad m_2 = 5 \text{ мин}$$

$$p_3 = 0.95, \quad m_3 = 10 \text{ мин}$$

Цель. Разработать GPSSV – модель для анализа временных характеристик РТК по результатам 100 прогонов мебели.

ЗАДАНИЕ 56. Моделирование процесса функционирования участка контроля

Исходные данные:

1. Изделия поступают из цеха на контроль через каждые  $a \pm b$  мин. (здесь и далее равномерный закон).
2. Каждый из 2-х контролеров выполняет свои функции  $b \pm \beta$  мин. После контроля примерно 75% изделий направляется на склад, а остальные – к наладчику для доводки.
3. Наладчик выполняет свои функции  $c \pm \gamma$  мин., и возвращает изделия на повторный контроль.
4. На участке контроля фиксируется число изделий, направленных на склад и прошедших наладчика.
5. После окончания смены изделия из цеха не поступают, а изделия из очереди обслуживаются контролерами.

Цель. Разработать GPSSV – модель для анализа процесса функционирования участка контроля в течение смены, т.е. 8 часов.

Первоначальный перечень экспериментов:

$$a = 5, \quad b = 2, \quad b = 9, \quad \beta = 3, \quad c = 30, \quad \gamma = 10.$$

ЗАДАНИЕ 57. Моделирование процессов обслуживания технологического модуля

Исходные данные:

1. Технологический модуль содержит  $m$  независимо работающих однотипных агрегатов, которые обслуживаются двумя операторами.
2. Агрегат требует обслуживания через каждые  $a \pm \alpha$  мин., причем время его обслуживания составляет  $b \pm \beta$  мин. (равномерный закон).
3. Обслуживание агрегатов осуществляется в порядке очереди без приоритетов. При этом через каждый час один из операторов выполняет наладку оборудования в течение  $c \pm \gamma$  мин. (равномерный закон).

Цель. Разработать GPSSV – модель для анализа процесса обслуживания технологического модуля в течение смены, т.е. 8 часов.

Первоначальный перечень экспериментов:

$$m = 10, \quad a = 25, \quad \alpha = 5, \quad b = 3, \quad \beta = 1, \quad c = 9, \quad \gamma = 3.$$

ЗАДАНИЕ 58. Моделирование процесса обслуживания коллектива пользователей

Исходные данные:

1. Терминальный класс имеет  $m$  активных терминалов, которые обслуживает консультант-программист.
2. Пользователь требует обслуживания через каждые  $30 \pm 10$  мин., причем время его обслуживания составляет  $3 \pm 2$  мин.

3. Обслуживание пользователей осуществляется в порядке очереди без приоритетов.
4. Если в системе возникает сбой, то консультант-программист обслуживает пользователя и после этого устраняет сбой. Сбои возникают в системе через каждые  $a \pm \alpha$  мин., а время их устранения  $b \pm \beta$  мин. (равномерный закон распределения).

Цель. Разработать GPSSV – модель для анализа процесса обслуживания коллектива пользователей в течение 8 часов.

Первоначальный перечень экспериментов:

$$m = 12, \quad a = 90, \quad \alpha = 30, \quad b = 9, \quad \beta = 3.$$

**ЗАДАНИЕ 59.** Моделирование процесса циклического обслуживания ЭВМ трех терминалов

Исходные данные:

1. ЭВМ обслуживает 3 терминала по круговому циклическому алгоритму.
2. Каждому терминалу выделяется  $t$  с. времени, если задание успевает обработаться, то обслуживание завершается.
3. Если предыдущий этап имеет альтернативный результат, то остаток задания становится в специальную очередь, использующую свободные циклы терминалов, т.е. задача обслуживается, если на каком-либо терминале нет заявок.
4. Интервалы поступления заявок на терминалы составляют  $d \pm \Delta d$  и имеют длину  $\mu_e \pm \Delta \mu_e$  единиц. Скорость обработки заданий ЭВМ составляет  $\nu$  ед./с.

Цель. Разработать GPSSV – модель функционирования ЭВМ. Определить среднюю загрузку ЭВМ, величину цикла терминала, позволяющую избежать образования специальной очереди.

Первоначальный перечень экспериментов:

$$t = 30, \quad d = 20, \quad \Delta d = 5, \quad \mu_e = 300, \quad \Delta \mu_e = 50, \quad \nu = 10.$$

**ЗАДАНИЕ 60.** Моделирование работы вычислительной системы (ВС)

Исходные данные:

1. Вычислительная система содержит три ЭВМ, работа которых организована по следующему алгоритму: с интервалом  $\mu_1 \pm \Delta \mu_1$  с. задания поступают в очередь на обработку 1-й ЭВМ, время обработки составляет  $t_1$  с. После задание поступает одновременно на вторую и третью ЭВМ, время обработки которых  $\mu_2 \pm \Delta \mu_2$  с. и  $\mu_3 \pm \Delta \mu_3$  с. соответственно.
2. Окончание обработки задания на любой ЭВМ означает снятие ее с решения.

Цель. Разработать GPSSV – модель работы системы в течение  $\tau$  часов. Определить необходимую емкость накопителей перед всеми ЭВМ, коэффициенты загрузки ЭВМ и функцию распределения случайного времени обслуживания заданий.

Первоначальный перечень экспериментов:

$$\mu_1 = 30, \quad \Delta \mu_1 = 2, \quad t_1 = 30, \quad \mu_2 = 14, \quad \Delta \mu_2 = 5, \quad \mu_3 = 16, \quad \Delta \mu_3 = 1, \quad \tau = 7$$

**ЗАДАНИЕ 61. Моделирование процесса функционирования двухмашинного вычислительного комплекса**

Исходные данные:

1. На вход двухмашинного вычислительного комплекса (ВК) от внешних абонентов А1, А2, А3 поступают сообщения на обработку. Дисциплина обслуживания - в порядке поступления.
2. Одна ЭВМ обрабатывает только поступающие сообщения с постоянным временем обслуживания ( $T_1$  – для сообщений об абоненте А1,  $T_2$  – для сообщений абонента А2,  $T_3$  – для сообщений абонента А3).
3. Вторая ЭВМ подключается к обработке при количестве сообщений в очереди  $q > 5$  и отключается при  $q \leq 3$ .
4. Интервалы между моментами поступления сообщений от абонентов равномерно распределены в интервалах  $[a_1, b_1]$ ,  $[a_2, b_2]$ ,  $[a_3, b_3]$  соответственно.
5. Время обслуживания сообщений второй ЭВМ постоянно и равно  $\tau_1$  – для сообщений первого абонента,  $\tau_2$  – второго,  $\tau_3$  – третьего.

Цель. Разработать GPSSV – модель для анализа процесса функционирования двухмашинного комплекса в течение суток. Провести анализ работы вычислительного комплекса.

Первоначальный перечень экспериментов:

$$a_1 = 7, \quad b_1 = 13, \quad a_2 = 15, \quad b_2 = 25, \quad a_3 = 22, \quad b_3 = 38.$$

**ЗАДАНИЕ 62. Моделирование процесса функционирования моечной станции**

Исходные данные:

1. Моечная станция имеет  $N$  мест на стоянке для автомобилей. Если клиенты подъезжают и не застают свободного места для ожидания, они уезжают.
2. Поток автомобилей является пуассоновским с интенсивностью  $\lambda$ . Время мойки автомобиля распределено экспоненциально с интенсивностью  $\mu$ .

Цель. Разработать GPSSV – модель и использовать ее для выбора оптимального числа мест  $N$  на стоянке так, чтобы число уехавших машин было минимальным.

Первоначальный перечень экспериментов:

$$\lambda = 5 \text{ авт/мин}, \quad \mu = 4 \text{ авт/мин}, \quad 3 \leq N \leq 7$$

**ЗАДАНИЕ 63. Моделирование процессов поломки и восстановления станка**

Исходные данные:

1. В станке используется деталь, которая время от времени выходит из строя. Как только используемая деталь отказывает, станок необходимо выключить. Имеется  $N$  запасных деталей.
2. Отказавшую деталь вынимают и на ее место ставят исправную запасную деталь. Ней исправные детали ремонтируют и после ремонта снова используют.
3. Время работы детали распределено по нормальному закону со средним  $\mu_1$

и стандартным отклонением  $\sigma_1$ .

4. Съём отказавшей детали со станка занимает  $t$  часов. Время, необходимое для того, чтобы установить запасную деталь, равно  $\tau$  часов. Время ремонта неисправной детали распределено по нормальному закону со средним  $\mu_2$  и среднеквадратическим отклонением  $\sigma_2$ .

Цель. Построить GPSSV – модель системы. Проанализировать работу станка в течение 5000 часов.

Первоначальный перечень экспериментов:

$$\mu_1 = 350, \quad \sigma_1 = 70, \quad t = 4, \quad \tau = 6, \quad \mu_2 = 8, \quad \sigma_2 = 0.5.$$

ЗАДАНИЕ 64. Моделирование процесса обслуживания танкеров в порту

Исходные данные:

1. Порт используют для заливки танкеров сырой нефтью. Порт имеет возможность заливать одновременно до  $N$  танкеров, которые прибывают в порт через интервалы времени, распределенные по равномерному закону в интервале  $[a_1, b_1]$ .
2. В порту имеется один буксир. Если буксир свободен, время подхода к стоянке или отхода от нее занимает  $T$  часов. Если буксир занят, танкеры ожидают его.

Цель. Построить GPSSV – модель, имитирующую работу порта в течение недели. Обеспечить в модели возможность определения времени пребывания танкеров в порту.

Первоначальный перечень экспериментов:

$$N = 3, \quad a_1 = 4 \text{ час.}, \quad b_1 = 18 \text{ час.}, \quad T = 1 \text{ час.}$$

ЗАДАНИЕ 65. Моделирование станции технического контроля производственной линии

Исходные данные:

1. Собранные телевизионные приемники проходят серию испытаний на станциях технического контроля. На последней станции проверяют регулировку установки кадров по вертикали. Отбракованный телевизор отправляют в цех наладки.
2. Предположим, что телевизоры попадают на последнюю станцию через интервалы времени, распределенные по равномерному закону в интервале  $[a_1, b_1]$ . На станции находятся два контролера. Каждому из них требуется на проверку телевизора время, равномерно распределенное в интервале  $[a_2, b_2]$ .
3. Примерно 85% телевизоров проходят проверку успешно и попадают в цех упаковки. Остальные 15% попадают в цех наладки, в котором находится один наладчик. Наладка занимает время, равномерно распределенное в интервале  $[a_3, b_3]$ .

Цель. Построить GPSSV – модель для анализа работы указанного подразделения производственной линии.

Первоначальный перечень экспериментов:

$$a_1 = 3 \text{ мин.}, \quad b_1 = 7 \text{ мин.}, \quad a_2 = 6 \text{ мин.}, \quad b_2 = 12 \text{ мин.}, \quad a_3 = 20 \text{ мин.}, \quad b_3 = 40 \text{ мин.}$$

ЗАДАНИЕ 66. Моделирование работы продовольственного магазина

Исходные данные:

1. Небольшой продовольственный магазин состоит из четырех прилавков и одной кассы на выходе из магазина.
2. Покупатели приходят в магазин, образуя пуассоновский поток с  $\lambda$  с.
3. Войдя в магазин, каждый покупатель берет корзинку и может обойти один или несколько прилавков, отбирая продукты, вероятность обхода конкретного прилавка указана в следующей таблице:

Прилавок	Вероятность покупки	Время обхода
1	0.65	$180 \pm 60$ с
2	0.85	$160 \pm 30$ с
3	0.70	$100 \pm 25$ с
4	0.90	$50 \pm 15$ с

4. После того, как товар отобран, покупатель становится в конец очереди к кассе. Время обслуживания покупателя в кассе пропорционально числу сделанных покупок, на одну покупку уходит время  $t$ .

Цель. Построить GPSSV – модель, имитирующую работу магазина в течение 8 часов. Оценить число покупателей, стоявших в очередь к кассе.

Первоначальный перечень экспериментов:

$$\lambda = 95, \quad t = 3 \text{ с.}$$

ЗАДАНИЕ 67. Моделирование процесса обслуживания клиентов в банке

Исходные данные:

1. В банке имеется  $N$  касс. Приход клиентов в банк описывается пуассоновским потоком с интенсивностью  $\lambda$ .
2. К каждому кассиру стоит очередь. Время обслуживания – экспоненциальное со средним  $\mu$ .
3. Если в момент входа клиента в банк хотя бы один кассир свободен, клиент сразу же попадает к этому кассиру. В противном случае клиент присоединяется к любой очереди, которая на текущий момент является кратчайшей.
4. Обслуживание клиентов в очереди осуществляется по принципу "первым пришел – первым обслужен". После обслуживания клиент уходит из банка.

Цель. Построить GPSSV – модель для анализа работы банка в течение 8 часов.

Первоначальный перечень экспериментов:

$$\lambda = 200 \text{ чел/час}, \quad \mu = 45 \text{ чел/час.}$$

ЗАДАНИЕ 68. Моделирование работы парикмахерской с одним креслом

Исходные данные:

1. В парикмахерскую с одним креслом могут приходить клиенты двух типов. Клиенты первого типа желают только стричься. Распределение интервалов их прихода является равномерным  $R[\tau_1, \tau_2]$ . Клиенты второго типа желают постричься и побриться. Распределение интервалов их прихода является равномерным  $R[\tau_3, \tau_4]$ .

2. Время стрижки распределено равномерно  $R[\tau_5, \tau_6]$ .
3. Время, необходимое на бритье, распределено равномерно  $R[\tau_7, \tau_8]$ .
4. Предположим, что парикмахерская открывается в 8 ч. 30 мин., а закрывается в 17 ч. 00 мин. Парикмахер имеет перерыв на обед в 12 ч. 00 мин. Клиенты, которые приходят в парикмахерскую в течение перерыва, ждут его возвращения.

Цель. Разработать GPSSV – модель и проимитировать работу парикмахерской.

Первоначальный перечень экспериментов:

$$\tau_1 = 25 \text{ мин.}, \quad \tau_2 = 45 \text{ мин.}, \quad \tau_3 = 40 \text{ мин.}, \quad \tau_4 = 80 \text{ мин.}, \quad \tau_5 = 12 \text{ мин.}, \quad \tau_6 = 24 \text{ мин.}, \\ \tau_7 = 8 \text{ мин.}, \quad \tau_8 = 12 \text{ мин.}$$

## **ЗАДАНИЕ 85. Моделирование работы моторного цеха**

### **авторемонтной станции Исходите данные**

1. В моторном цехе работает  $m$  слесарей, каждой из которых занят ремонт мотора одного автомобиля. Цех работает 8 часов утра, до 20 часов вечера.
2. Длины временных промежутков между поступлениями автомобилей на ремонт двигателя - случайные величины, которые независимы, одинаково распределены по равномерному закону на интервале  $[\tau_-, \tau_+]$  часов. Время ремонта двигателя - случайная равномерно распределенная величина на интервале  $[\mu_-, \mu_+]$  часов.
3. Если всем слесарей заняты, то автомобиль становится в очередь на ремонт, длина которой практически не ограничена в течение рабочего дня. На следующий день прежняя очередь не сохраняется.
4. В 20 часов все автомобили, которые находятся на рабочих местах слесарей и не отремонтированы до конца, остаются в цехе. На следующий день ремонт начинается с этих автомобилей.

Цель. Составить GPSSV -модель работы цеха в течение  $T$  дней. Предполагая, что в первый день с утра, все рабочие места слесарей были свободны. Оценить среднее время ожидания автомобиля в очереди в первый, второй и третий день работы стоящий, .. среднее время простоя слесарей.

Первоначальный перечень экспериментов:  $m=5, \tau_-=0, \tau_+=2, \mu_-=2, \mu_+=3, T=5$ .

**ЗАДАНИЕ 86. Моделирование работы моторного цеха авторемонтной станции с приоритетами**

Исходные данные:

1. Моторный цех работает с 8.00 до 20.00. В нем работают  $m$  слесарей, каждый из которых занят ремонтом одной машины.
2. Длины временных промежутков между поступлениями автомобилей участников и инвалидов Великой отечественной войны (ВОВ) случайные равномерно распределенные величины на интервале  $[\tau_-, \tau_+]$  часов, рядовых клиентов - случайные равномерно распределенные величины на интервале



$[\mu_-, \mu_+]$  часов. Время ремонта двигателя- случайная величина, имеющая экспоненциальное распределение с параметром  $\lambda$  часов

3. Автомобили участников и инвалидов Великой Отечественной войны имеют приоритетное обслуживание. Если все  $m$  слесарей заняты, то автомобиль становится в очередь на ремонт, длина которой практически не ограничена в течение рабочего дня. На следующий день очередь рядовых клиентов не сохраняется.
4. В 20 часов все автомобили, которые находятся на рабочих местах слесарей и не отремонтированы до конца и автомобили инвалидов и участников Великой Отечественной войны, остаются в цехе. На следующий день ремонт начинается с этих автомобилей.

Цель. Составить GPSSV - модель работы цеха в течение  $T$  дней, предполагая, что в первый день в 8.00 все рабочие места слесарей были свободны. Оценить отдельно для рядовых клиентов и клиентов, пользующихся льготами, среднее время пребывания в очереди в течение первых трех дней, среднее время простоя слесарей в первый и пятый день работы.

Первоначальный перечень экспериментов:  $m=5$ ,  $\tau_-=0$ ,  $\tau_+=2$ ,  $\mu_-=2$ ,  $\mu_+=3$ ,  $T=5$ .

ЗАДАНИЕ 87. Моделирование процесса функционирования - двухпроцессорной ЭВМ

#### Исходные данные

1. Двухпроцессорная ЭВМ имеет общую память, разделенную на  $n$ , блоков. Каждой задаче при ее решении отводится один блок.
2. Длины временных промежутков между поступлениями задач случайны, независимы и одинаково распределены по равномерному закону  $[\tau_-, \tau_+]$
3. Время обработки порции информации в процессоре  $\Pi_j$  случайно и подчинено экспоненциальному закону  $E(\mu_j), j=1,2$
4. Между обработкой порций с вероятностью  $P$  возможно обращение к внешней памяти со случайным временем обращения, подчиненным равномерному закону  $R[t_-, t_+]$ . С вероятностью  $q=1-p$  задача оказывается "решенной" и покидает ЭВМ.

Цель. Разработать GPSSV - модель для анализа, функционирования ЭВМ при решении  $N$  задач.

Первоначальный перечень экспериментов:  $\tau_-=2$ ,  $\tau_+=14$ ,  $\mu_-=0.2$ ,  $\mu_+=0.5$ ,  $p=0.6$ ,  $t_-=2$ ,  $t_+=8$ ,  $N=100$

ЗАДАНИЕ 88. Моделирование процесса статистического Приемочного контроля последовательных партий продукции

Исходные данные:

1. Партии продукции поступают на контроль последовательно, образуя пуассоновский поток с интенсивностью  $\lambda$  партий/мин.
2. Время контроля партии распределено равномерно и зависит от вида контроля:

ослабленный контроль (ОК)	30-40 мин.
нормальный контроль (НК)_____	50-60 мин.
усиленный контроль (УК)	80-100 мин.

3. Для первой партии всегда используется нормальный контроль. Для  $i$ -й партии вид контроля зависит от качества  $i$ -й партии, характеризуемого входным уровнем дефектности  $P_{i-1}$  ( $i=2,3,\dots$ ). При этом для  $i$ -й партии используется:
  4. НК, если  $P_{i-1} \in [p_0 + \delta]$ ,
  5. ОК, если  $P_{i-1} < p_0 + \delta$ ,
  6. УК, если  $p_{1-1} > p_0 + \delta$ ,
  7. где  $p_0$  - приемочный уровень дефектности. В последнем случае  $i$ -я партия бракуема.
8. Величины  $\{p_i\}$  являются случайными равномерно распределенными на интервале  $[p_0 + E]$

Цель. Разработать GPSSV - модель процесса приемочного контроля в течение 25 рабочих смен (8 часов каждая). Подсчитать число бракованных партий при различных соотношениях  $\delta$  и  $E$ .

Первоначальный перечень экспериментов:  $\lambda=0,01$ ,  
 $p_i=0,1, \delta=0,02, E=0,5\delta, \delta, 2\delta$

## ЗАДАНИЕ 89. Моделирование САПР

### Исходные данные

1. Система автоматизации проектирования состоит из ЭВМ и трех терминалов. Каждый проектировщик формирует задание на расчет в интерактивном режиме.
2. Набор строки занимает время, равномерно распределенное на интервале  $[a_1, e_1]$ . Получение ответа на строку требует  $t_1$  секунд работы ЭВМ и  $t_2$  секунд работы терминала.
3. После набора  $N$  строк задание считается сформированным и поступает на решение, при этом ЭВМ в течение времени, равномерно распределенного на интервале  $[a_2, b_2]$ , прекращает выработку ответов на вводимые строки.

4. Вывод результата требует  $t_1$  секунд работы терминала. Анализ результата занимает у проектировщика  $t_2$  секунд, после чего цикл повторяется.

Цель. Построить GPSSV - модель системы и смоделировать ее работу в течение 6 часов. Определить вероятность простоя проектировщика из-за занятости ЭВМ и коэффициент загрузки ЭВМ,

Первоначальный перечень экспериментов  $a_1=5c$ ,  $v_1=15c$ ,  $t_1=3c$ ,  $t_2=5c, N=10$ ,  $\tau_1=8c$ ,  $\tau_2=30c$

#### ЗАДАНИЕ 90. Моделирование системы управления запасами

##### Исходные данные

1. Детали, необходимые для работы цеха, находятся на цеховом и центральных складах. На цеховом складе хранятся комплекты деталей, потребность появляется через интервалы времени, равномерно распределенные в интервале  $[a_1, v_1]$  составляет один комплект.
2. В случае снижения запасов до  $n$ , комплекты формируются в течение  $T$  мин, заявка на пополнение запасов цехового «клада до полного объема в  $N$  комплектов, которая посылается в центральный склад, где в течение времени, равномерно распределенного на интервале  $[a_2, v_2]$ , происходит комплектование.
3. За время, равномерно распределенное на интервале  $[a_3, v_3]$  осуществляется доставка деталей в цех.

Цель. Построить GPSSV -модель системы. Смоделировать работу цеха в течение 400 часов. Определить вероятность простоя цеха из-за отсутствия деталей и среднюю загрузку цехового склада.

Первоначальный перечень экспериментов:  $a_1=50$  мин,  $v_1=70$  мин,  $N=20, n=3, T=60$  мин,  $a_2=40$  мин,  $v_2=80$  мин,  $a_3=55$  мин,  $v_3=65$  мин.

#### ЗАДАНИЕ 91. Моделирование работы вычислительного центра

##### Исходные данные

1. На вычислительный центр через интервалы времени, распределенные равномерно на  $[a_1, v_1]$ , поступают задания длиной, распределенной по равномерному закону на интервале  $[a_2, v_2]$
2. Скорость ввода, вывода и обработки заданий  $N$  байт/мин.
3. Задания проходят последовательно ввод, обработку и вывод, буферируясь перед каждой операцией.
4. После вывода  $M$  процентов заданий оказываются выполненными неправильно вследствие сбоев и возвращаются на ввод.
5. Для ускорения обработки задания в очередях располагаются по возрастанию их длины, т.е. короткие сообщения обслуживаются в первую

очередь. Задания, выполненные неверно, возвращаются на ввод и во всех очередях обслуживаются первыми.

Цель. Построить GPSSV - модель системы. С моделировать работу вычислительного центра в течение 30 часов. Определить необходимую емкость буферов и функцию распределения времени обслуживания заданий.

Первоначальный перечень экспериментов:  $a_1=200$ ,  $b_1=400$ ,  $a_2=300$ ,  $b_2=700$ ,  $M=5$ ,  $N=100$

## ЗАДАНИЕ 92. Моделирование работы транспортного цеха

### Исходные данные

1. Транспортный цех объединения, имеющий  $m$  грузовиков, обслуживает три филиала А, В, С. В момент начала работы грузовики находятся в филиале А. Грузовики перевозят изделия А в В и из В в С.
2. Погрузка в А занимает в среднем  $t_1$  мин, переезд из А в В длится  $t_2$  мин, разгрузка (погрузка.) в В –  $t_3$  мин, переезд в С-  $t_4$  мин, разгрузка в С –  $t_5$  мин, (всюду распределение экспоненциальное). Если к моменту погрузки в А и В изделий нет, грузовики уходят дальше по маршруту.
3. Изделия выпускаются партиями по 1000 шт. в А через 20+3 мин, в В через 20+5 мин.(равномерный закон распределения). Грузоподъемность грузовика 1000 изделий.

Цель. Разработать GPSSV - модель работы цеха в течение рабочей недели. Определить частоту пустых перегонов.

Первоначальный перечень экспериментов:  $t_1=20$ ,  $t_2=30$ ,  $t_3=40$ ,  $t_4=30$ ,  $t_5=20$

## ЗАДАНИЕ 93. Моделирование участка термической обработки шестерен

### Исходные данные

1. На участке термической обработки выполняется цементация и закаливание шестерен.
2. Время поступления деталей на участок, длительность цементации и закаливания распределены равномерно на интервалах  $[a+\alpha]$ ,  $[b+\beta]$ ,  $[c+\gamma]$  мин. соответственно
3. В зависимости от величины суммарного времени обработки  $t$  выполняются следующие действия: если  $t > 25$  мин, то шестерни покидают участок (относятся к 1-му сорту); если  $t < 25$  - передаются на повторную закалку: если  $t < 20$ , то шестерни относятся ко второму сорту.

Цель: Разработать GPSSV - модель процесса обработки на участке шестерен. Определить режим функционирования, обеспечивая выход 90 % изделий 1-го сорта.

Первоначальный перечень экспериментов:  $a=10, \alpha=5, b=10, \beta=7, c=10, \gamma=6, n=400$

#### ЗАДАНИЕ 94. Модель автобусной остановки

##### Исходные данные:

1. По расписанию автобус должен приходить на остановку каждые  $T$  мин, однако возможно его опоздание на время, равномерно распределенное в интервале  $[a_1, b_1]$  мин. Опоздание автобуса никак не зависит от опоздания предыдущего автобуса и не влияет на опоздание следующего.
2. Приход людей на автобусную остановку подчиняется закону Пуассона с интенсивностью  $\lambda$  человек каждые  $t$  мин.
3. Автобус вместимостью  $N$  человек в момент своего прибытия везет  $M+\Delta M$  человек. После того, как от трех до семи пассажиров выйдут (распределение равномерное), в автобус входят столько ожидающих, сколько возможно. Те, кто не смог сесть после того, как автобус заполнился, уходят и больше не возвращаются.
4. Для высадки пассажира требуется  $4+3$  с, а для посадки  $8+4$  с. Пассажиры выходят и входят один за другим. Ожидающие не начинают посадки до тех пор, пока из автобуса не выйдут все желающие. Посадка производится в соответствии с правилом "первым пришел - первым обслужен".

Цель. Построить GPSSV - модель, имитирующую события на автобусной остановке в течение 12 часов. Оценить количество людей, пришедших на остановку и не уехавших в автобусе.

Первоначальный перечень экспериментов:  $T=30$  мин,  $a_1=0$  мин,  $b_1=3$  мин,  $\lambda=12$  человек,  $t=30$  мин,  $N=50$ ,  $M=35$ ,  $\Delta M=15$

#### ЗАДАНИЕ 95\*. Моделирование процесса управления производством

##### Исходные данные

1. На трикотажной фабрике  $N$  швейных машин работают по 8 часов в день и по 5 дней в неделю. В резерве имеется  $M$  машин.
2. Время безотказной работы машины распределено по равномерному закону в интервале  $[a_1, b_1]$ . Если машина выходит из строя, ее заменяют резервной либо сразу в случае, когда она имеется в наличии, либо по мере ее появления. Тем временем сломанную машину отправляют в ремонтную мастерскую, где ее чинят и возвращают в цех, но уже в качестве резервной.
3. На ремонт швейной машины уходит время, равномерно рас-

предельное на интервале  $[a_2, b_2]$

Цель. Разработать GPSSV - модель системы с целью минимизации стоимости производства.

Первоначальный перечень экспериментов:  $N=50, 1 < M < 5$ , оплата ремонта машины 3.75 руб в час. За машину, находящуюся в резерве, нужно платить 30 руб. в день,  $a_1=132, b_1=182, a_2=4, b_2=10$

**ЗАДАНИЕ 96.** Моделирование производственных процессов в ремонтной мастерской Исходные данные

1. Ремонтная мастерская состоит из цеха ремонта, в который поступают и ремонтируются механизмы, и цеха контроля, в котором отремонтированные механизмы тестируются и выпускаются из мастерской или направляются на повторный ремонт. В цехе ремонта имеются  $m$  одинаковых (параллельных) рабочих мест, а в цехе контроля  $n$  параллельно работающих контролеров. Случайная длительность промежутка времени между поступлениями механизмов в мастерскую имеет экспоненциальное распределение  $E(\lambda)$  •
2. Случайное время ремонта механизма имеет распределение Эрланга с математическим ожиданием  $\mu$  и дисперсией  $\sigma^2$ . В цехе ремонта в первую очередь обслуживаются механизмы с наименьшим временем ремонта.
3. Очередь в цехе контроля образуется в порядке времени поступления механизмов в этот цех. Тестирование механизма занимает  $\tau$  единиц времени. После тестирования механизм отправляется повторно в цех ремонта с вероятностью  $p^k$ , где
  1.  $K$  - число раз, которое данный механизм направлялся на повторный ремонт.
4. В начальный момент времени  $t=0$  все цехи свободны и в цех ремонта поступает первый механизм.

Цель. Разработать GPSSV - модель и осуществить компьютерную имитацию производственных процессов в ремонтной мастерской на временном промежутке  $0, T$  для оценки средних длин очередей, среднего числа механизмов в мастерской, распределений вероятностей времени пребывания механизма в системе и времени ожидания в цехе ремонта.

Первоначальный перечень экспериментов:  $m=3, n=1, \lambda=0.1, \mu=22$  час.,  $\sigma=15$  час,  $\tau=6$  час,  $p=0.15, T=2000$  час

**ЗАДАНИЕ 97.\* Моделирование сборочной линии.**

Исходные данные

1. Сборочная линия содержит  $m$  рабочих мест; время обработки изделия на каждом из них—случайная величина с экспоненциальным распределением  $E(\mu)$ .

2. Изделия движутся от первого рабочего места ко второму, затем к третьему и так далее до  $m$ -го, где обработка завершается.
3. Очередное изделие поступает на первое рабочее место через каждые  $\tau_1$  единиц времени. Если обработка изделия на каком-либо рабочем месте не завершена в течение  $\tau_1$  единиц времени, изделие направляется на одно из  $l$  вспомогательных рабочих мест, расположенных на сборочной линии. Здесь осуществляется полная доработка изделия. К оставшемуся времени выполнения незаконченной операции при этом добавляется 1 единица времени. Время выполнения каждой последующей незаконченной операции на вспомогательных рабочих местах – случайная величина, распределённая по экспоненциальному закону  $E(\tau_2)$ .

Цель. Разработать GPSSV-модель для оценки статистических характеристик загрузки всех обслуживающих устройств и доли изделий, снятых с линии на каждой из операций на временном промежутке  $[0, T]$ .

Первоначальный перечень экспериментов:

$m = 3, l = 2, \tau_1 = 15, \tau_2 = 16, T = 1600, \mu = 10.$

### **ЗАДАНИЕ 98.\* Моделирование процесса добычи руды карьерным способом.**

Исходные данные

1. В карьере  $N$  самосвалов доставляют руду от  $m$  экскаваторов к измельчителю, причём после выгрузки руды у измельчителя самосвалы всегда возвращаются к одним и тем же экскаваторам. Используются самосвалы грузоподъемностью 20 т. и 50 т. К  $i$ -му экскаватору приписано  $n_{i_1}$  20 т. и  $n_{i_2}$  50 т. самосвалов ( $i = \overline{1, m}$ ), так что  $\sum_{i=1}^m (n_{i_1} + n_{i_2}) = N$ .
2. От грузоподъемности самосвала зависит время его загрузки, поездка до измельчителя, разгрузки и обратной поездки к своему экскаватору. Для 20 т. самосвалов: случайное время погрузки имеет экспоненциальное распределение  $E(\tau_1)$ ; время поездки постоянно и равно  $\tau_2$ ; случайное время разгрузки имеет экспоненциальное распределение  $E(0.4\tau_1)$ ; время обратной поездки постоянно и равно  $0.6\tau_2$ . Для 50 т. самосвалов соответственно имеем:  $E(2\tau_1)$ ;  $1.2\tau_2$ ;  $E(0.8\tau_1)$ ;  $0.8\tau_2$ .
3. Очереди к каждому экскаватору организованы по принципу «первым пришёл – первым обслужен». В очереди к измельчителю приоритет имеют большегрузные самосвалы.

Цель. Разработать GPSSV-модель и проимитировать функционирование описанной системы добычи руды карьерным способом в течение  $T$  единиц времени для оценки средней загрузки экскаваторов и измельчителя и оценки средних длин очередей в системе.

Первоначальный перечень экспериментов:  $N = 9, m = 3, n_{i_1} = 2, n_{i_2} = 3$ ,  
( $i = 1, 2, 3$ );  $\tau_1 = 5, \tau_2 = 2.5, T = 480$ .

### **ЗАДАНИЕ 99.\* Моделирование процесса функционирования грузового аэропорта.**

#### Исходные данные

1. Грузы прибывают для отправки в аэропорт в контейнерах: случайные длительности промежутков между прибытиями двух контейнеров распределены по равномерному закону  $R[\tau_1, \tau_2]$ . В грузовом аэропорту нет фиксированного расписания, а каждый самолёт отправляется в момент его полной загрузки.
2. Для перевозки грузов из данного аэропорта в аэропорт назначения используются два типа самолётов. Есть  $N_1$  самолётов с грузоподъёмностью  $m$  контейнеров и  $N_2$  самолётов повышенной грузоподъёмности  $2m$  контейнеров. Время полёта каждого самолёта туда и обратно является случайной величиной, имеющей усеченное на  $[\tau_3, \tau_4]$  гауссовское распределение  $N(\mu, \sigma^2)$ .
3. Управляющий аэропортом старается, как можно чаще использовать самолёты меньшей грузоподъёмности. Самолёты, поднимающие  $2m$  контейнеров, используются лишь при отсутствии других.
4. Время погрузки каждого контейнера в самолёт случайно и имеет равномерное распределение  $R[\tau_5, \tau_6]$ .

Цель. Разработать GPSSV-модель и осуществить имитацию процесса функционирования грузового аэропорта на отрезке времени  $[0, T]$  для оценки среднего времени “простоя” самолётов каждого типа.

Первоначальный перечень экспериментов:  $\tau_1 = 0.5$  мин,  $\tau_2 = 1.5$  мин,  
 $N_1 = 3, N_2 = 2, m = 80, \tau_3 = 2$  час,  $\tau_4 = 4$  час,  
 $\mu = 3$  час,  $\sigma = 1$  час,  $\tau_5 = 1$  мин,  $\tau_6 = 2$  мин.

### **ЗАДАНИЕ 100.\* Моделирование статистически регулируемого технологического процесса штамповки.**

#### Исходные данные

1. Время изготовления одного изделия на участке штамповки распределено экспоненциально с интенсивностью  $\lambda$  изделий/сек. На наладку ТП в начале смены уходит  $t$  мин.



2. Через каждые  $\tau$  мин. после начала работы рабочий берёт на контроль выборки изделий, с помощью специальных шаблонов подсчитывает число дефектных изделий  $\nu$  и наносит полученное значение на контрольную карту.

# Глава 6. Статистическое моделирование в финансовом анализе.

В детерминированном финансовом анализе параметры финансовой операции такие как, первоначальная сумма, продолжительность ссуды, ставка процентов, считаются заданными. Теория таких расчетов хорошо развита [ ].

Однако в реальных ситуациях часть из этих параметров, а то и все параметры в целом, могут быть не полностью определены в силу различных причин. Чтобы сделать обоснованные выводы о ходе таких финансовых операций, необходимы дополнительные априорные предположения о поведении таких не вполне определенных параметров. В данном разделе мы будем считать, что не вполне определенные параметры финансовой операции являются случайными величинами с заданными, либо спрогнозированными законами распределения вероятностей. Тогда и результат финансовой операции (наращенная сумма, современная величина) будут также случайными величинами и для их оценки можно успешно применять аппарат статистического моделирования. Исследователя будут интересовать такие числовые характеристики, например, наращенной суммы, как минимальное и максимальное значение, среднее значение, дисперсия и вероятность попадания наращенной суммы в заданный интервал. По результатам статистического моделирования эти числовые характеристики могут быть оценены.

В финансовой математике, в силу ряда причин [ ], наибольшее применение находят два типа распределений случайных величин: дискретное распределение и “трапецеидальное” распределение  $Tr(a,b;c,d)$ ,  $a \leq c \leq d \leq b$ . Распределение  $Tr(a, b; c, d)$ -это абсолютно-непрерывное распределение с плотностью распределения заданной на интервале  $[a, b]$  и принимающей постоянное значение  $2/(b+d-a-c)$  на интервале  $[c, d]$ , линейно возрастающее на интервале  $[a, c]$  и линейно убывающее на интервале  $[d, b]$ . Если  $a = c$ ,  $d = b$ , то “трапецеидальное” распределение обращается в равномерное распределение на отрезке  $[a, b]$ . Если  $c = d$ , то “трапецеидальное” распределение трансформируется в “треугольное” распределение. Наконец, если  $c = d = (a+b)/2$ , то имеем стандартное “треугольное” распределение .

Эти типы распределений, имеющие хорошую экономическую интерпретацию, легко моделируются.

## 6.1. Моделирование наращенных сумм и современных величин платежа.

Приведем формулы для вычисления наращенных сумм и современных величин платежей для ставок простых, сложных процентов и банковской учетной ставки. Все величины, стоящие в правых частях ниже приведенных формул, являются параметрами финансовой операции. Часть из этих параметров может быть не полностью заданной. В этих случаях не полностью определенные параметры будем рассматривать, как случайные величины, и будем моделировать их числовые значения. После того как все значения неизвестных параметров будут смоделированы, можно по приведенным ниже формулам, вычислить результирующее значение финансовой операции. А по совокупности результирующих значений можно оценить все числовые характеристики финансовой операции, используя методы математической статистики.

Формула наращения по ставке простых процентов, изменяющейся кусочно - линейно от интервала к интервалу, имеет вид:

$$S = P(1 + n_1 i_1 + n_2 i_2 + \dots + n_k i_k) , \quad (1)$$

где  $S$  - наращенная сумма,  $P$  - первоначальная сумма,  $i_{s, s=\overline{1, k}}$  - ставка простых процентов, определенная на интервале  $n_s$ ,  $n_1 + n_2 + \dots + n_k = n$ . Каждая из величин, стоящих в правой части (1), может быть случайной величиной.

Если уравнение (1) разрешить относительно  $P$ , то получим современное значение  $P$  наращенной суммы  $S$ :

$$P = S(1 + n_1 i_1 + n_2 i_2 + \dots + n_k i_k)^{-1}. \quad (2)$$

Если наращение производится по ставкам сложных процентов, то формула (1) трансформируется в формулу

$$S = P(1 + i_1)^{n_1} (1 + i_2)^{n_2} \cdot \dots \cdot (1 + i_k)^{n_k}, \quad (3)$$

а формула (2) – в формулу :

$$P = S(1 + i_1)^{-n_1} (1 + i_2)^{-n_2} \cdot \dots \cdot (1 + i_k)^{-n_k}. \quad (4)$$

Учет векселя с суммой  $S$ , поглощаемого за  $t$  дней до его погашения по годовой банковской ставке  $d$ , осуществляется по формуле

$$P = S(1 - \frac{t}{360} \cdot d). \quad (5)$$

### Задание 1.

В начале года на депозит положено 560 тысяч рублей сроком на один год. По условиям финансового соглашения банк имеет право изменить процентную ставку в начале каждого квартала. Первоначальная ставка процентов  $i_1$  известна и равна 70 % годовых, ставки процентов  $i_1, i_2, i_3$ , соответственно, во втором, третьем и четвертом кварталах неизвестны заранее. Есть основания считать, что  $i_2$  будет равномерно распределена на интервале [73 %, 78 %] ;  $i_2$  будет иметь “трапецеидальное” распределение на интервале [73 %, 78 %] с максимальным значением плотности распределения на интервале [74 %, 76 %], а ставка процентов  $i_4$  в четвертом квартале примет значение 79 % с вероятностью 0,4 и 80 % с вероятностью 0,6. причем  $i_1, i_2, i_3$  - независимые случайные величины.

По 10 реализациям ставок процентов  $i_1, i_2, i_3$  оценить числовые характеристики наращенной суммы  $S$ :  $S_{min}$ ,  $S_{max}$ ,  $E\{S\}$ ,  $D\{S\}$ ,  $P\{97600 \leq S \leq 97900\}$ .

*Решение.* Обозначим через  $\mathbf{i} = (i_1, i_2, i_3)$  трехмерный вектор. Пусть в результате моделирования получены 10 реализаций этого вектора:  $\mathbf{i}^{(1)} = (0,723; 0,748; 0,8)$ ,  $\mathbf{i}^{(2)} = (0,746; 0,733; 0,79)$ ,  $\mathbf{i}^{(3)} = (0,758; 0,756; 0,8)$ ,  $\mathbf{i}^{(4)} = (0,734; 0,779; 0,8)$ ,  $\mathbf{i}^{(5)} = (0,734; 0,741; 0,79)$ ,  $\mathbf{i}^{(6)} = (0,729; 0,757; 0,8)$ ,  $\mathbf{i}^{(7)} = (0,755; 0,736; 0,8)$ ,  $\mathbf{i}^{(8)} = (0,739; 0,747; 0,8)$ ,  $\mathbf{i}^{(9)} = (0,753; 0,758; 0,79)$ ,  $\mathbf{i}^{(10)} = (0,742; 0,762; 0,8)$ .

Используя формулу (1) с  $t_1 = 0,7$ ;  $n_1 = n_2 = n_3 = n_4 = 0,25$ , получаем:  $S^{(1)} = 975,94$ ;  $S^{(2)} = 975,66$ ;  $S^{(3)} = 981,96$ ;  $S^{(4)} = 981,82$ ;  $S^{(5)} = 975,1$ ;  $S^{(6)} = 978,04$ ;  $S^{(7)} = 978,74$ ;  $S^{(8)} = 978,04$ ;  $S^{(9)} = 980,14$ ;  $S^{(10)} = 980,56$ .

По выборочным значениям  $S^{(i)}$ ,  $i = \overline{1,10}$ , оценим следующие числовые характеристики  $S$ :

$$S_{\min} = 975,1 \text{ тыс.}; S_{\max} = 980,56 \text{ тыс.}; E\{S\} = \frac{1}{10} \sum_{i=1}^{10} S^{(i)} = 978,6 \text{ тыс.}; D\{S\} = 6,276 \text{ тыс.}; P\{97600 \leq S \leq 97900\} = 0,5.$$

## Задание2.

По 1000 реализаций  $S$  вычисленных по формуле (1), оценить числовые характеристики  $S$ :  $S_{\min}$ ,  $S_{\max}$ ,  $E\{S\}$ ,  $D\{S\}$ ,  $P\{S_1 \leq S \leq S_2\}$ , для следующих значений параметров, входящих в правую часть формулы (1):

- 1)  $\kappa = 4$ ,  $n_1 = n_2 = n_3 = n_4 = 0,25$  года,  $P = 1230$ ,  $S_1 = 1335$ ,  $S_2 = 1340$ ;  $t_1, t_2, t_3, \square$  – независимые случайные величины,  $t_1 \in R(8,2\%; 9\%)$ ,  $t_2 \in R(8,4\%; 9,1\%)$ ,

$t_3$	8,6 %	8,7 %	8,8 %	9,2 %
P	0,1	0,3	0,5	0,1

$t_4$	8,7 %	8,9 %	9,3 %
P	0,1	0,6	0,3

- 2)  $\kappa = 3$ ,  $P = 15800$ ;  $n = 1$  год,  $S_1 = 16800$ ,  $S_2 = 17000$ ,  $n_1, n_2$  – независимые дискретные случайные величины:

$n_1$ , лет	1/12	1/4
P	0,7	0,3

$n_2,$ лет	1/4	1/3	1/2
P	0,5	0,4	0,1

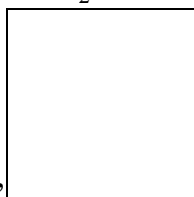
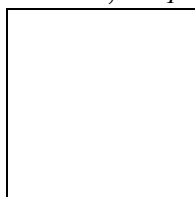
Если  $n_1 = 1/12$ , то  $t_1 \in R$  (3 %; 4,2 %), если  $n_1 = 1/4$ , то  $t_1 \in R$  (5 %; 6,2 %),  
если  $n_2 = 1/4$ , то  $t_2 \in R$  (5 %; 6,2 %);  $n_2 = 1/3$ , то  $t_2 \in R$  (6 %; 7 %);

если  $n_2 = 1/2$ , то  $t_2 \in R$  (6,5 %; 7,8 %). Если  $n_3 \leq 0,5$ , то  $t_3 \in R$  (5 %; 7,8 %);  
если  $n_3 > 0,5$ , то  $t_3 \in R$  (6 %; 8 %).

### Задание3.

По 1000 реализаций  $S$  вычисленных по формуле (3), оценить числовые характеристики  $S$ :  $S_{min}$ ,  $S_{max}$ ,  $E\{S\}$ ,  $D\{S\}$ ,  $P\{S_1 \leq S \leq S_2\}$ , для следующих значений параметров, входящих в правую часть формулы (3):

1)  $\kappa = 2$ ,  $n_1 = n_2 = 2$  года,  $P = 10000$ ,  $S_1 = 13900$ ,  $S_2 = 14100$ ;



— независимые случайные величины,  $t_1 \in R$ (9 %; 10,2 %),  $t_2 \in Tr$  (8 %,12 %; 9 %,10 %);

2)  $\kappa = 3$ ,  $P = 20800$ ;  $n = 6$  лет,  $n_1 = 2$  года,  $t_1 = 9$  %,  $n_2$  - дискретная случайная величина:

$n_2,$ годы	1	2
P	0,8	0,2

если  $n_2 = 1$ , то

$t_2$	9 %	9,1 %	9,3 %
P	0,7	0,2	0,1

если  $n_2 = 2$ , то

$t_2$	9,2 %	9,3 %	9,4 %
P	0,6	0,3	0,1

если  $n_3 = 3$ , то  $t_3 = 10$  %, если  $n_3 = 2$ , то  $t_3 = 9,2$  %.

#### Задание 4.

По 10000 реализаций  $P$ , вычисляемых по формуле (2), оценить числовые характеристики современной величины  $P$  наращенной суммы  $S$ :  $P_{min}$ ,  $P_{max}$ ,  $E\{P\}$ ,  $D\{P\}$ ,  $P\{P_1 \leq P \leq P_2\}$ , для следующих значений параметров, входящих в правую часть формулы (2):

1)  $\kappa = 4$ ,  $n_1 = n_2 = n_3 = n_4 = 0,25$  года, ставки процентов, измеренные в долях, измеряются в начале каждого квартала по цепной зависимости Маркова с трендом:  $i_k = f(k) + \xi_k$ ,  $k = 1, 2, 3, 4$ , где  $f(k) = 0,05 + 0,001(k - 1)$  - тренд, характеризующий общую тенденцию изменения ставок процентов,  $\xi_k$  - маржа, измеряющаяся по цепной зависимости Маркова, с двумя состояниями. В первом состоянии  $\xi_k$  принимает значение 0,005, во втором - значение 0,01. Вектор вероятностей начальных состояний  $\pi = (1,0)$ , матрица одношаговых переходов имеет следующие компоненты:  $p_{11} = 0,7$ ;  $p_{12} = 0,3$ ;  $p_{21} = 0,2$ ;  $p_{22} = 0,8$ . Наращенная сумма  $S = 10000$ ,  $P_1 = 9600$ ,  $P_2 = 9700$ ;

2) Оцените числовые характеристики начальной суммы  $P$ , вычисляемой по формуле (4) с теми же значениями параметров, что и в пункте 1).

#### Задание 5.

Вексель на сумму  $S = 245000$  может быть учтен в банке за 120 дней до его погашения с вероятностью 0,6, но тогда банковская учетная ставка  $d$  будет описываться дискретной случайной величиной

$d$	6 %	6,2 %	6,4 %
P	0,7	0,2	0,1

за 80 дней до его погашения с вероятностью 0,2 и в этом случае

$d$	6,5 %	6,8 %	7 %
$P$	0,6	0,2	0,2

за 40 дней до его погашения с вероятностью 0,2 и в этом случае  $d \in R(7,5\%; 9\%)$ .

. По 1000 реализаций  $P$ , вычисленных по формуле (3), оценить числовые характеристики  $P$ :  $P_{min}$ ,  $P_{max}$ ,  $E\{P\}$ ,  $D\{P\}$ ,  $P\{200000 \leq P \leq 220000\}$ .

## 6.2. Моделирование величины реальной наращенной суммы платежей с учетом инфляции.

Реальная наращенная сумма ссуды или депозита с учетом инфляции определяется по формуле:

$$C = \frac{S}{j_p} \quad (1)$$

и определяет покупательскую способность наращенной суммы  $S$ , измеренную по номиналу (смотри формулы (1), (3) предыдущего пункта 7.1, где  $j_p$  – индекс цен. Так как инфляция является цепным процессом, т.е. цены в периоде  $k$  повышаются на  $h_k$  процентов относительно уровня, сложившегося в периоде  $k-1$ , то индекс за  $m$  таких периодов равен произведению цепных индексов цен

$$j_p = \prod_{k=1}^m \left(1 + \frac{h_k}{100}\right), \quad (2)$$

где типы инфляции  $h_k$  измеряются в процентах. В формуле (1) наращенная сумма  $S$  и индекс цен  $h_k$  должны вычисляться за один и тот же промежуток времени.



Если речь идет об оценке в начальный момент реальной покупательской способности наращенной суммы депозита  $P$  размещенного в банке на  $n$  лет, и срок  $n$  содержит в себе  $m$  месяцев, то месячные темпы инфляции  $h_k$ ,  $k = \overline{1, m}$  заранее неизвестны. В этом случае непосредственно нельзя воспользоваться формулой (1). В таких случаях для оценки будущего значения реальной наращенной суммы  $S$  моделируют значения последующих месячных темпов инфляции на основе той или иной модели изменения этих темпов. В этом разделе мы будем предполагать, что темпы инфляции в будущие месяцы являются случайными величинами с заданными или спрогнозированными законами распределения вероятностей.

## Задание 1

Пусть условия депозита такие же как в задании 1 предыдущего пункта 7.1. Дополнительно будем предполагать, что поквартальные темпы инфляции  $h_1$ ,  $h_2$ ,  $h_3$  и  $h_4$  являются независимыми случайными величинами с прогнозируемыми законами распределения вероятностей:  $h_l \in R(3\%, 5\%)$ ,  $l = 1$ ,

2, 3;  $h_4 \in Tr(2\%, 6\%; 3\%, 4\%)$ . Ставки процентов



и темпы инфляции

$h_1$ ,  $h_2$ ,  $h_3$  и  $h_4$  – независимые в совокупности случайные величины.

По 10 реализациям случайных параметров данной финансовой операции оцените числовые характеристики  $C$ :  $C_{min}$ ,  $C_{max}$ ,  $E\{C\}$ ,  $D\{C\}$ ,  $P\{840000 \leq C \leq 850000\}$ .

*Решение.* Для простоты рассуждений предположим, что результаты моделирования значений наращенной суммы  $S^{(1)}$ ,

$l = \overline{1, 10}$  такие же как и в решении задания 1 пункта 7.1. Далее, обозначим через  $h = (h_1, h_2, h_3, h_4)$  четырехмерный вектор, результаты моделирования 10 реализаций этого вектора, допустим, следующие:

$h^{(3)} = (3,96; 4,02; 4,33; 3,52)$ ,  $h^{(4)} = (4,02; 3,33; 4,17; 3,03)$ ,  $h^{(5)} = (3,49; 3,24; 4,17; 3,84)$ ,  $h^{(6)} = (3,31; 4,28; 3,40; 2,83)$ ,  $h^{(7)} = (4,09; 4,31; 3,09; 3,82)$ ,  $h^{(8)} = (4,23; 3,43; 3,82; 3,73)$ ,  $h^{(9)} = (3,14; 3,97; 4,03; 3,57)$ ,  $h^{(10)} = (4,39; 4,02; 3,03; 3,06)$ .

Тогда, в соответствии с формулой (2),

$$j_p^{(1)} = 1,1583027; j_p^{(2)} = 1,2020007; j_p^{(3)} = 1,1679293; j_p^{(4)} = 1,1535849; j_p^{(5)} = 1,1557227; j_p^{(6)} = 1,1454699; j_p^{(7)} = 1,1620704; j_p^{(8)} = 1,1609796; j_p^{(9)} = 1,1553875; j_p^{(10)} = 1,1530006.$$

Согласно формуле (1)

$$C^{(1)} = 975,94 / 1,1583027 = 842,56; C^{(2)} = 811,697; C^{(3)} = 840,77; C^{(4)} = 851,103; C^{(5)} = 843,715; C^{(6)} = 853,833; C^{(7)} = 842,238; C^{(9)} = 842,427; C^{(10)} = 850,442.$$

По выборочным значениям  $C^{(i)}, i = \overline{1,10}$ , оценим числовые характеристики  $C$ :

$$C_{min} = 811,697 \text{ тыс.}, C_{max} = 853,833 \text{ тыс.}, E\{C\} = 842,711 \text{ тыс.}, D\{C\} = 124,69 \text{ тыс.}, P\{840000 \leq C \leq 850000\} = 0,6.$$

## Задание 2.

По 1000 реализациям оценить числовые характеристики наращенной суммы  $C$ :  $C_{min}$ ,  $C_{max}$ ,  $E\{C\}$ ,  $D\{C\}$ ,  $P\{1200 \leq C \leq 1250\}$ . Условия для вычисления номинальной наращенной суммы  $S$ , без учета инфляции, сформулированы в задании 2, пункт 1), предыдущего раздела 7.1, а распределение вероятностей поквартальных темпов инфляции точно такое же, как и в задании 1 данного раздела.

## Задание 3.

По 1000 реализациям оценить числовые характеристики наращенной суммы  $C$ :  $C_{min}$ ,  $C_{max}$ ,  $E\{C\}$ ,  $D\{C\}$ ,  $P\{13000 \leq C \leq 13200\}$ . Условия для вычисления номинальной наращенной суммы  $S$ , без учета инфляции, сформулированы в задании 3, пункт 1), предыдущего раздела 7.1. Поквартальные темпы инфляции, выраженные в процентах, описываются процессом случайного блуждания с трендом:  $h_I = 2,6 + 0,4 \cdot i + \varepsilon_i$ , где  $\varepsilon_i$  - случайный процесс с дискретным фазовым пространством состояний;  $\varepsilon_i \in X = \{x : x = l \cdot 0,1; l = 0, \pm 1, \pm 2, \dots\}$ ,  $i = \overline{1,16}$ ,  $\varepsilon_1 = 0$ . Если в квартале с номером  $i$  процесс находится в состоянии  $x \in X$ , то в следующем квартале  $\varepsilon_{i+1} = x + \eta_i$ , где  $\eta_i \in \{-0,1; 0; 0,1\}$  - дискретная случайная величина с распределением вероятностей :  $P\{\eta_i = 0,1\} = 0,2$ ;  $P\{\eta_i = -0,1\} = 0,1$ ;  $P\{\eta_i = 0\} = 0,7$ ;

### 6.3. Моделирование обменного курса валют при двойной конверсии валют.

Рассмотрим задачу о вычислении наращенных сумм депозитов с использованием двойной конверсии валюты [ ].

Если имеются свободные денежные средства, то их можно нарастить, поместив на депозит. Для определенности рассуждений будем предполагать, что свободные денежные средства  $P_v$  - это свободно конвертируемая валюта, которую можно конвертировать в рубли. Эти свободные денежные средства можно нарастить двумя способами: 1) используя прямое наращение капитала на валютный депозит под  $i$  годовых процентов на срок  $n$  лет; 2) используя двойную конверсию валюты, обменяв вначале по курсу  $K_0$  валюту на рубли, затем поместив рубли на депозит под  $j$  годовых процентов, и, наконец, обменяв наращенную рублевую массу на валюту по обменному курсу  $K_1$ . Для определенности рассуждений будем полагать, что указанные депозиты краткосрочные,  $n \leq 1$  и ставки процентов  $i, j$  – это годовые ставки простых процентов. В данной ситуации возникает вопрос: что выгоднее – прямое помещение капитала или двойная конверсия? Ответить на него помогают два барьерных значения  $K_1$ :  $K_1^* = K_0(1 + ni)$ ,  $\bar{K}_1 = \frac{K_1^*}{(1 + nj)}$ . Эти значения можно вычислить в самом начале операции депонирования. Если 1)  $K_1^* < K_1$ , то двойная конверсия убыточна; 2)  $K_1^* > K_1$ , то двойная конверсия прибыльна. В ситуации 2) надо решать вопрос: а может прямое помещение капитала более выгодно, чем использование двойной конверсии валют? В данной ситуации используется значение  $\bar{K}_1$ , если  $K_1 < \bar{K}_1$ , то выгоднее применить двойную конверсию валют, если  $K_1^* > K_1 > \bar{K}_1$ , то прямое помещение капитала на депозит выгоднее, чем использование двойной конверсии.

Однако использовать этот алгоритм принятия решения не представляется возможным, так как значение обменного курса  $K_1$  в конце операции неизвестно. Для того, чтобы оценить  $K_1$ , используют различные методы прогноза  $K_1$ .

Опишем подход к принятию решений в данной ситуации, который основан на том, что неизвестное значение  $K_1$  мы будем рассматривать, как случайную величину с заданным либо спрогнозированным законом распределения вероятностей. В данном случае мы не только сможем выбрать лучшую из трех альтернатив принятия решения, но и количественно оценить в виде вероятности степень обоснованности выбора той или иной альтернативы: 1)  $P_1 = P(K_1^* < K_1)$  – вероятность того, что двойная конверсия убыточна, 2)  $P_2 = P(K_1 < \bar{K}_1)$  – вероятность того, что двойная конверсия дает большую выгоду, чем прямое помещение капитала, 3)  $P_3 = 1 - P_1 - P_2$  –

вероятность того, что прямое помещение капитала более выгодно, чем использование двойной конверсии валют. Оценивая значения  $P_1$ ,  $P_2$  и  $P_3$ , можно сделать обоснованный вектор в пользу той или иной альтернативы.

## Задание 1

Решено поместить 1200\$ на трехмесячный депозит. Курс обмена валюты на данный момент  $K_0 = 1820$  руб. На валютном вкладе ставка 10 % годовых, на рублевом вкладе – 50 % годовых. Есть основания считать, что обменный курс  $K_1$  через 3 месяца будет описываться "трапецеидальным" законом распределения вероятностей,  $K_1 \in Tr(.1830, 2100; 1840, 1850)$

По 10 реализациям  $K_1$  оценить вероятности  $P_1$ ,  $P_2$  и  $P_3$  трех альтернатив принятия решения.

Решение. В данном случае  $K_1^* = 2048$ ,  $\bar{K}_1 = 1998$ . Пусть в результате моделирования получены следующие значения обменных курсов  $K_1$ : 1838, 1842, 2010, 1843, 1999, 1849, 1881, 1837, 1847, 1844. По этим значениям оценим вероятности  $P_1$ ,  $P_2$  и  $P_3$ :  $P_1 = 0$ ;  $P_2 = 0,8$ ;  $P_3 = 0,2$ .

*Вывод:* данная конверсия выгодна и двойная конверсия даст большую выгоду, чем прямое поглощение капитала.

## Задание 2

По 1000 реализациям обменного курса  $K_1$  оцените вероятности  $P_1$ ,  $P_2$  и  $P_3$  принятия альтернативных решений для условий депонирования вклада из задания 1 и следующих распределений  $K_1$ :

1) дискретный тип распределения  $K_1$ : 1840 с вероятностью 0,13; 1850 с вероятностью 0,47; 1970 с вероятностью 0,12;

2) равномерное распределение,  $K_1, K_1 \in R(1840;2000)$ ;

3) нормальное распределение,  $K_1, K_1 \in N(1965;0,1)$ ;

4) марковская зависимость с трендом. Значения обменного курса  $K_{1j}$  в день с номером  $j$  от начала действия депозита складывается из значения тренда  $f(j)$  и маржи  $\varepsilon_{\xi_j}$ , т.е.  $K_{1j} = f(j) + \varepsilon_{\xi_j}$ , где тренд  $f(j) = 1830 + j$ ,  $\varepsilon_1 = -1, \varepsilon_2 = 0, \varepsilon_3 = 2$  - заданные значения маржи;  $\xi_j$  - номер состояния цепи Маркова с тремя состояниями и матрицей одношаговых переходов с элементами :  $p_{11} = 0,2$ ;  $p_{21} = 0,7$ ;  $p_{13} = 0,1$ ;  $p_{21} = 0,2$ ;  $p_{22} = 0,5$ ;  $p_{23} = 0,3$ ;  $p_{31} = 0,2$ ;  $p_{32} = 0,6$ ;  $p_{33} = 0,2$ . Цепь выходит из второго состояния, т.е.  $\varepsilon_2 = 0$ .

# Приложение

**Таблица П.1**  
**Системные числовые атрибуты GPSS/PC**

Символическое обозначение	Описание системных числовых атрибутов
RNj	Реализация БСВ для j-ого датчика ( $1 \leq j \leq 999$ ); $0 \leq RNj \leq 999$ во всех случаях, кроме случаев использования RNj в качестве аргумента функции и переменной, когда $0 \leq RNj \leq 0.999999$
C1	Текущее значение относительного момента времени (устанавливается в 0 операторами CLEAR и RESET)
AC1	Текущее значение абсолютного времени (устанавливается в 0 только оператором CLEAR)
TG1	Текущее значение счетчика числа завершений
XN1	Номер активного транзакта (обрабатываемого в данный момент)
Z1	Размер свободной оперативной памяти в байтах
M1	Время прерывания в модели активного транзакта

**Таблица П.2**  
**Стандартные числовые атрибуты GPSS/PC**

Тип объектов	Символическое обозначение СЧА	Описание СЧА
Транзакты	Pj P*<имя>  MPj MBj   PR	Значение параметра j транзакта Значение параметра с именем <имя> Значение времени, равное AC1-Pj Флаг синхронизации: 1, если транзакт в блоке j из одного семейства с активным транзактом; 0 – в противном случае Приоритет активного транзакта
Блоки	Nj	Общее число транзактов,

	Wj	вошедших в блок Текущее число транзактов в блоке
Устройства	Fj  FIj  FRj  FCj FVj  FTj	Текущее состояние устройства (0 – устройство свободно, 1 – устройство занято или прервано) Признак прерывания устройства (1 – состояние прерывания, 0 – в противном случае) Коэффициент использования устройства в долях от 1000 Общее число входов в устройство Признак доступности устройства (1 – устройство доступно, 0 – устройство не доступно) Среднее время использования устройства одним транзактом
Логические ключи	LSj	Состояние логического ключа (1 – установлен, 0 – не установлен)
Памяти (многоканальные устройства)	Sj Rj SRj  SMj SCj STj  SEj  SFj  SVj	Текущее содержание памяти Число свободных единиц памяти Коэффициент использования памяти j в долях от 1000 Максимальное содержание памяти Общее число входов в память Среднее время пребывания транзактов в памяти Признак пустоты памяти (1 – память пуста, 0 – память не пуста) Признак заполненности памяти (1 – память заполнена, 0 – память не заполнена) Признак доступности памяти (1 – память доступна, 0 – память не доступна)
Очереди	Qj QAj QMj QCj QZj	Текущая длина очереди Средняя длина очереди Максимальная длина очереди Общее число входов в очередь Число нулевых входов (без

	$QT_j$  $QX_j$	<p>ожидания) в очередь</p> <p>Среднее время пребывания транзакта в очереди (включая нулевые входы)</p> <p>Среднее время пребывания транзакта в очереди (без нулевых входов)</p>
Переменные и функции	$V_j$  $BV_j$  $FN_j$	<p>Значение арифметической переменной с фиксированной или плавающей точкой</p> <p>Значения булевой переменной (1 – истина, 0 – ложь)</p> <p>Значение функции (дробная часть отбрасывается за исключением использования в качестве функции-модификатора или аргумента другой функции)</p>
Таблицы	$TB_j$ $TD_j$  $TC_j$	<p>Среднее значение аргумента</p> <p>Общее число включений в таблицу (число значений аргумента таблицы)</p> <p>Среднеквадратическое отклонение аргумента таблицы</p>
Ячейки	$X_j$	Содержимое ячейки
Матрицы ячеек	$MX_j(a,b)$	Содержимое элемента матрицы ячеек, расположенного в строке a и столбце b
Списки пользователя	$CH_j$ $CA_j$ $CM_j$  $CS_j$ $CT_j$	<p>Текущее число транзактов в списке</p> <p>Среднее число транзактов в списке</p> <p>Максимальное число транзактов в списке</p> <p>Общее число транзактов в списке</p> <p>Среднее время пребывания транзакта в списке</p>

**Таблица П.3**  
**Операторы блоков GPSS/PC**

Код оператора	Операнд	Содержание операнда
Назначение		
ADVANCE	A	Среднее время

Задержка транзакта	[B]	Модификатор-интервал Модификатор-функция: FNj,FNS<имя>,FN*j
ASSEMBLE Объединение транзактов	A	Число транзактов из одного семейства, объединяемых в один транзакт
ASSIGN Изменение значений параметров	A B C	Номер или диапазон изменения номеров параметров и указатель режима (накопления "+", вычитания – "–", замещения) Значение, на которое изменяется параметр (параметры) Номер модификатора функции
COUNT XXX Определение количества объектов из заданного множества, удовлетворяющих заданному условию	A B C [D] [E]	Номер параметра – счетчика объектов Нижняя граница диапазона номеров объектов Верхняя граница диапазона номеров объектов Значение, сравниваемое со значением СЧА в поле E Код СЧА анализируемых объектов (без номера и имени)
DEPART Уменьшение очереди	A [B]	Номер (имя) очереди Число единиц, на которое уменьшается длина очереди (не превосходит текущей очереди)
ENTER Занятие памяти	A [B]	Номер (имя) памяти Число занимаемых транзактом единиц памяти
GATE XXX Выбор направления в зависимости от состояния объекта	A [B]	Номер (имя) устройства, памяти, ключа Номер (метка) альтернативного блока в случае невыполнения проверяемого условия
GENERATE Создание и запуск транзактов в модель с заданным законом времени поступления	[A] [B] [C] [D] [E]	Среднее время поступления транзактов Модификатор-интервал или модификатор -функция (CM.ADVANCE) Время задержки первого транзакта Количество создаваемых блоком транзактов Приоритет транзактов



<b>LEAVE</b> Освобождение памяти	A [B]	Номер (имя) памяти Число освобождаемых транзактом единиц памяти
<b>LINK</b> Перевод транзакта из списка текущих событий в список пользователя	A B  [E]	Номер (имя) списка пользователя Режим упорядочивания списка: FIFO – "первым пришел – первым обслужился", LIFO – "первым пришел – последним обслужился", номер параметра Номер (метка) альтернативного блока
<b>LOGIK X</b> Изменение состояния ключа $X \in \{S, R, I\}$	A	Номер (имя) логического ключа, состояние которого изменяется: S – "включается", R – "выключается", I – "инвертируется"
<b>LOOP</b> Организация циклов	A  [B]	Номер (имя) параметра транзакта, используемого в качестве счетчика цикла Номер (метка) первого блока в цикле
<b>MATCH</b> Синхронизация транзактов	A	Номер (метка) сопряженного блока MATCH. Метка исходного блока MATCH указывается в поле A сопряженного блока
<b>PREEMPT</b> Прерывание обслуживания уст- ройством одного транзакта с целью захвата устройства прерывающим транзактом	A [B]  [C]  [D]  [E]	Номер (имя) прерываемого устройства Задание режима прерывания: по приоритету – PR, обычный – операнд опущен Номер (метка) блока для прерванного транзакта Номер (имя) параметра прерванного транзакта для записи остатка времени обслуживания RE – режим удаления прерванного транзакта с направлением по адресу в поле C
<b>PRIORITY</b> Изменение приоритета транзакта	A	Новое значение приоритета транзакта
<b>QUEUE</b> Увеличение очереди	A [B]	Номер (имя) очереди Число единиц, на которое увеличивается длина очереди
<b>RELEASE</b>	A	Номер (имя) освобождаемого

Освобождение устройства		устройства транзактом, который его занимал
RETURN Снятие прерывания с устройства	A	Номер (имя) устройства, на котором снимается прерывание транзактом, осуществившим прерывание
SAVEVALUE Изменение значений ячеек	A B	Номер (имя) ячейки и указатель режима: накопления (+), вычитания ( – ), замещения (по умолчанию) Значение, на которое изменяется содержимое ячейки
SEIZE Занятие устройства	A	Номер (имя) занимаемого устройства
SELECT XXX Выбор первого из множества объектов, удовлетворяющего заданному условию	A – E [F]	Аналогичны блоку COUNT Номер (метка) альтернативного блока в случае невыполнения проверяемого условия
SELECT MAX(MIN)	A – F	Аналогичны предыдущему случаю (поиск объектов с MAX (MIN) значением указанного СЧА)
SPLIT Создание копий исходного транзакта	A [B] [C]	Число создаваемых копий Номер (метка) блока, к которому направляются копии Номер (имя) параметра для присвоения копиям последовательных номеров
TABULATE Табуляция текущего значения аргумента таблицы	A [B]	Номер (имя) таблицы Число добавляемых в соответствующий интервал таблицы единиц
TERMINATE Удаление транзактов из модели	[A]	Число единиц, на которое уменьшается счетчик завершений (поле A оператора START)
TEST YY Выбор направления движения транзакта в за-	A,B [C]	Величины, сравниваемые с помощью оператора отношения $YY \in \{L, LE, G, GE, E, NE\}$ Номер (имя) альтернативного блока при невыполнении проверяемого

зависимости от значений СЧА		условия
TRANSFER Выбор направления движения транзактов из множества возможных направлений	[A]  [B],[C] [D]	Режим выбора направления: "," – режим безусловного перехода; "BOTH" – выбор из двух направлений; "ALL" – выбор из более чем двух направлений; ".K" – выбор с вероятностью Значения номеров (меток) блоков Константа, используемая для вычисления возможных адресов в режиме "ALL"
UNLINK Удаление транзактов из списка пользователя	A B  [C]	Номер (имя) списка пользователя Номер (метка) блока следования удаляемых транзактов Счетчик числа удаляемых транзактов ("ALL" – удаление всех транзактов)

### Примечание к таблице П.3

Всюду в таблице, где особо не оговаривается, операнды могут иметь одну из следующих форм представления: положительное целое число (K), <имя>, СЧАj, СЧА\*<параметр>. В таблице используются обозначения: [] – признак того, что соответствующий операнд может быть опущен; xxx – логический указатель (табл.П.4) или оператор отношения ('L', 'LE', 'E', 'NE', 'G', 'GE') в операторах с расширенным полем операции.

**Таблица П.4**  
**Логические указатели**

Тип объекта	Логический указатель	Значение
Устройства	FU FNU FI  FNI	Устройство занято Устройство не занято Устройство обслуживает прерывание Устройство не обслуживает прерывание
Памяти	SF SNF SE SNE	Память заполнена Память не заполнена Память пуста Память не пуста
Логические переключатели	LR	Логический переключатель сброшен

	LS	Логический переключатель установлен
--	----	--

## Литература

1. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978.
2. Калашников В.В. Организация моделирования сложных систем. – М.: Знание, 1982
3. Советов Б.Я., Яковлев С.А. Моделирование систем: учебник для вузов. – М.: Высш. Шк., 1985
4. Шрайбер Т.Д. Моделирование на GPSS. – М.: Машиностроение, 1980.
5. Максимей И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988.
6. Коуден Д. Статистические методы контроля качества. – М.: Физматиз., 1961.
7. Кнут Д. Искусство программирования на ЭВМ. – М.: Мир, 1977.
8. Ермаков С.М., Михайлов Г.А. Курс статистического моделирования – М.: Наука, 1976.
9. Поляк Ю.Г. Вероятностное моделирование на ЭВМ. – М.: Сов. радио, 1971.
10. Быков В.В. Цифровое моделирование в статической радиотехнике. – М.: Сов. радио, 1973.
11. Андерсон Т.В. Статистический анализ временных рядов. – М.: Мир, 1978.
12. Ермаков С.М. Метод Монте-Карло и смежные вопросы. – М.: Наука, 1975.
13. Михайлов Г.А. Некоторые вопросы методов Монте-Карло. – Новосибирск: Наука. Сиб. отделение, 1974.
14. Соболев И.М. Метод Монте-Карло. – М.: Наука, 1985.
15. Соболев И.М. Численные методы Монте-Карло. – М.: Наука, 1973.
16. Ермаков С.М. Жиглявский А.А. Математическая теория оптимального эксперимента. – М.: Наука, 1987.
17. Федоров В.В. Теория оптимального эксперимента. – М.: Наука, 1971.
18. Клейнен Дж. Статистические методы в имитационном моделировании. Вып. 1,2. – М.: Статистика, 1978.
19. Асатурян В.И. Теория планирования эксперимента. – М.: Радио и связь, 1983.
20. Ивченко Г.И., Медведев Ю.И. Математическая статистика. – М.: Высш. шк., 1984.
21. Большев Л.Н., Смирнов Н.В. Таблицы математической статистики. – М.: Наука, 1983.
22. Айвазян С.А. и др. Прикладная статистика. Т. 1,2,3 – М.: Финансы и статистика, 1983,1985, 1987.

23. Харин Ю.С., Степанова М.Д. Практикум на ЭВМ по математической статистике. – Минск.: Университетское, 1987.
24. Киндлер Е. Языки моделирования. – М.: Энергоатомиздат, 1985.
25. Прицкер А. Введение в имитационное моделирование и язык СЛАМ-11. – М.: Мир, 1987.
26. Питерсон Д. Теория сетей Петри и моделирование систем . – М.: Мир, 1984.
27. Catalog of simulation software. Microcomputer, minicomputer and mainframe software. //Simulation, 1988.October/
28. Schriber T.J. An introduction to simulation/ Using GPSS/H. John Wiley & Sons, 1991/
29. Гнеденко Б.В. Курв теории вероятностей. – М.: Наука, 1969.
30. Кендалл М.Дж., Стьюарт А. Многомерный статистический анализ и временные ряды. – М.: Наука, 1976.
31. Аптон Г. Анализ таблиц сопряженности. – М.: Наука, 1982.
32. Андерсон Т.В. Введение в многомерный статистический анализ. – М.: Наука, 1963.
33. Технология системного моделирования (Под общей редакцией С.В. Емельянова, В.В. Калашникова, М. Франка, А. Явора). . – М.: Машиностроение; Берлин: Техник, 1983.
34. Параллельная обработка информации. Распараллеливание алгоритмов обработки (т.1). (Под ред. А.Н. Свенсона). – Киев: Навукова думка, 1985.
35. Cox S.W. Cox A.J. GPSS/PC: A user oriented simulation system.// Modeling and simulation on microcomputers (R.G. Lavery ed.) The society for Computers simulation. San-Diego. California. 1985.
36. Schriber T.J. Introduction to GPSS/PC. // Proceeding of IEEE. San-Francicssco. California. 1987.
37. Cox S.W. The interactive grafics and annimation of GPSS/PC. // Proceeding of IEEE. San-Francicssco. California. 1987.
38. Кашьяп Р.А., Рао А.Р. Построение динамичных стохастических моделей по экспериментальным данным. – М.: Наука, 1983.
39. Кенеми Дж., Снелл Дж. Конечные цепи Маркова. – М.: Наука, 1970.
40. Айвазян С.А. Программное обеспечение ПЭВМ по статистическому анализу данных. // Заводская лаборатория. 1991, №1 – 54-58 с.