

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»  
(САМАРСКИЙ УНИВЕРСИТЕТ)

*Е.В. СИМОНОВА*

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ:  
РАЗРАБОТКА МОДЕЛИ  
И ОРГАНИЗАЦИЯ ЭКСПЕРИМЕНТА

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по программе высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника

Самара  
Издательство Самарского университета  
2023

© Самарский университет, 2023  
ISBN 978-5-7883-1952-0

УДК 519.876.5(075)+004.94(075)

ББК В18я7+3973.1я7

С375

Рецензенты: канд. техн. наук, доц. Л. С. Зеленко;

д-р техн. наук, проф. С. В. Смирнов

***Симонова, Елена Витальевна***

**С375 Имитационное моделирование информационно-вычислительных систем: разработка модели и организация эксперимента:** учебное пособие / *Е.В. Симонова*; Министерство науки и высшего образования Российской Федерации, Самарский университет. – Самара: Издательство Самарского университета, 2023. – 1 CD-ROM (2,96 Мб). – Загл. с титул. экрана. – Текст: электронный.

**ISBN 978-5-7883-1952-0**

Учебное пособие позволяет в интерактивном режиме с единых методологических позиций рассмотреть различные этапы разработки компьютерной имитационной модели информационно-вычислительной системы, включая построение концептуальной модели исследуемой системы и ее формализацию, алгоритмизацию и компьютерную реализацию, анализ модели, планирование и проведение компьютерных экспериментов, получение и интерпретацию результатов моделирования.

Содержит иллюстрации всех этапов процесса разработки и реализации модели, а также проведения компьютерных экспериментов с моделью, что имеет большое учебно-методическое значение и необходимо при самостоятельной работе студентов во время выполнения ими лабораторных работ, а также содержит варианты заданий для выполнения лабораторных работ.

Предназначено для обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, выполняющих лабораторные работы по курсу «Моделирование информационно-вычислительных систем».

Подготовлено на кафедре информационных систем и технологий.

**Минимальные системные требования:**

PC, процессор Pentium, 160 МГц;

Microsoft Windows XP; мышь;

дисковод DVD-ROM; Adobe Acrobat Reader.

© Самарский университет, 2023

[Перейти к оглавлению](#)

Редакционно-издательская обработка издательства  
Самарского университета

Подписано для тиражирования 19.10.2023.

Объем издания 2,96 Мб.

Количество носителей 1 диск.

Тираж 11 дисков.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»  
(САМАРСКИЙ УНИВЕРСИТЕТ)  
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Издательство Самарского университета.  
443086, Самара, Московское шоссе, 34.

[Перейти к оглавлению](#)

## ОГЛАВЛЕНИЕ

<i>Введение</i> .....	7
<i>1. Моделирование как основной метод исследования сложных систем</i> .....	8
1.1. Основные определения.....	8
1.2. Классификация моделей по степени абстрагирования модели от оригинала .....	9
1.3. Формальная модель объекта исследования (реальной системы) .....	13
1.4. Понятие состояния системы .....	14
1.5. Структура имитационной модели .....	16
1.6. Области применения имитационных моделей.....	17
1.7. Основные этапы моделирования систем .....	20
<i>2. Постановка задачи исследования информационно-вычислительной системы</i> .....	24
2.1. Объект исследования .....	24
2.2. Описание задачи исследования .....	25
2.3. Формализованная постановка задачи исследования .....	27
2.4. Анализ возможных методов решения.....	28
2.4.1. Аналитический метод.....	28
2.4.2. Метод имитационного моделирования .....	29
<i>3. Разработка имитационной модели информационно-вычислительной системы</i> .....	30
3.1. Разработка структурно-функциональной модели .....	31
3.2. Разработка концептуальной модели .....	31

3.2.1. Таблица функционального соответствия элементов модели и элементов реальной системы .....	33
3.2.2. Алгоритмы моделирования случайных величин .....	34
3.3. Выбор программных средств моделирования .....	36
3.4. Разработка алгоритма моделирования и программы имитационной модели .....	38
3.4.1. Описание элементов модели .....	38
3.4.2. Алгоритм моделирования .....	40
3.4.3. Описание программной реализации имитационной модели .....	40
3.4.4. Программа имитационной модели .....	41
3.5. Организация единичного эксперимента с имитационной моделью .....	42
3.5.1. Анализ результатов моделирования .....	42
3.5.2. Оценка точности результатов моделирования .....	44
4. <i>Планирование экспериментов в системе GPSS World</i> .....	46
4.1. Организация факторного эксперимента с моделью .....	46
4.2. Однофакторный эксперимент .....	48
4.3. Проведение дисперсионного анализа в системе GPSS World .....	52
4.3.1. Дисперсионный анализ в планировании эксперимента .....	52
4.3.2. Последовательность проведения дисперсионного анализа для однофакторного эксперимента .....	54
4.3.3. Библиотечная процедура ANOVA .....	57
4.4. Проведение двухфакторного эксперимента .....	58
4.4.1. План двухфакторного эксперимента .....	59

4.4.2. Последовательность проведения дисперсионного анализа для двухфакторного эксперимента .....	65
4.5. Поиск наилучших решений. Оптимизирующий эксперимент .....	69
4.5.1. Организация оптимизирующего эксперимента .....	70
4.6. Выводы и рекомендации по использованию модели .....	75
<i>Заключение</i> .....	76
<i>Список литературы</i> .....	77
<i>Приложение А. Лабораторная работа. Моделирование информационно-вычислительных систем в среде GPSS World</i> .....	79
А.1. Цель лабораторной работы .....	79
А.2. Последовательность выполнения лабораторной работы .....	80
А.3. Содержание отчета по лабораторной работе .....	80
А.4. Варианты индивидуальных заданий .....	81
А.5. Контрольные вопросы .....	104
<i>Приложение Б. Операторы описания блоков GPSS World</i> ...	106
<i>Приложение В. Операторы описания данных и контроля управления GPSS World</i> .....	109
<i>Приложение Г. Сообщения GPSS World об ошибках</i> .....	110

## ВВЕДЕНИЕ

В настоящее время происходит интенсивное внедрение информационных технологий во все сферы деятельности. Метод моделирования нашел практическое применение при решении разнообразных задач в широком спектре предметных областей.

Моделирование представляет собой один из основных методов познания, является формой отражения действительности и заключается в выяснении или воспроизведении тех или иных свойств реальных объектов, предметов и явлений с помощью других объектов, процессов, явлений, либо с помощью абстрактного описания в виде изображения, совокупности уравнений, алгоритмов и программ. Возможности моделирования, то есть перенос результатов, полученных в ходе построения и исследования модели, на оригинал основаны на том, что модель в определенном смысле отображает (воспроизводит, моделирует, описывает, имитирует) некоторые интересующие исследователя черты объекта.

Существуют различные методы моделирования: физическое, аналитическое, структурно-функциональное, имитационное. Основная ценность метода имитационного моделирования состоит в применении методологии системного анализа. Имитационное моделирование позволяет осуществить исследование изучаемой или проектируемой системы, последовательно проходя взаимосвязанные этапы:

- содержательная постановка задачи;
- разработка концептуальной модели системы;
- разработка и программная реализация имитационной модели;
- проверка достоверности модели и оценка точности результатов моделирования;

- планирование и проведение компьютерного эксперимента;
- принятие решений.

Это позволяет применять основные принципы системного подхода для решения практических задач, а также использовать имитационное моделирование как универсальный подход для принятия решений в условиях неопределенности с учетом в моделях трудноформализуемых факторов.

Одним из наиболее эффективных и самых распространенных является язык моделирования GPSS (General Purpose Simulation System), который используется для построения дискретных имитационных моделей и проведения компьютерных экспериментов [1, 2].

Разделы учебного пособия последовательно раскрывают порядок разработки имитационных моделей, проведения экспериментов, факторного и дисперсионного анализа с использованием системы имитационного моделирования GPSS World. В заключении обсуждаются перспективы развития данного направления

Учебное пособие, в первую очередь, предназначено для обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, а также может быть полезно широкому кругу читателей, практикующихся в области компьютерных наук.

## **1. МОДЕЛИРОВАНИЕ КАК ОСНОВНОЙ МЕТОД ИССЛЕДОВАНИЯ СЛОЖНЫХ СИСТЕМ**

### **1.1. Основные определения**

*Модель* – это условный образ объекта исследования, конструируемый так, чтобы отобразить существенные для исследования характеристики объекта (свойства, взаимосвязи, параметры). Моделирование заключается в выявлении или

[Перейти к оглавлению](#)



воспроизведении свойств реальных объектов, предметов и явлений с помощью других объектов, процессов, явлений, либо с помощью абстрактного описания в виде изображения, совокупности уравнений, алгоритмов и программ.

Возможности моделирования, т.е. перенос результатов, полученных в ходе построения и исследования модели, на оригинал, основаны на том, что модель в определенном смысле отображает некоторые интересующие исследователя характеристики объекта.

Подобие процесса, протекающего в модели  $M$ , реальному процессу, протекающему в системе-оригинале  $S$ , является условием правильного функционирования модели, а не целью моделирования.

**Цель моделирования** – задача изучения какой-либо стороны функционирования объекта.

**Объект моделирования** – сложные информационно-вычислительные и организационно-технические системы.

## **1.2. Классификация моделей по степени абстрагирования модели от оригинала**

На Рисунок 1 представлена классификация моделей по степени абстрагирования модели от оригинала.

**Физическая модель** эквивалентна или подобна оригиналу, либо процесс ее функционирования такой же, как у оригинала и имеет ту же самую (или иную) физическую природу.

Натурная модель – это реальная исследуемая система, полностью адекватная оригиналу.

Масштабная модель – система той же физической природы, что оригинал, но отличающаяся масштабами. При соблюдении геометрического подобия оригинала и модели натурные

значения всех размеров модели умножаются на один и тот же масштабный коэффициент.

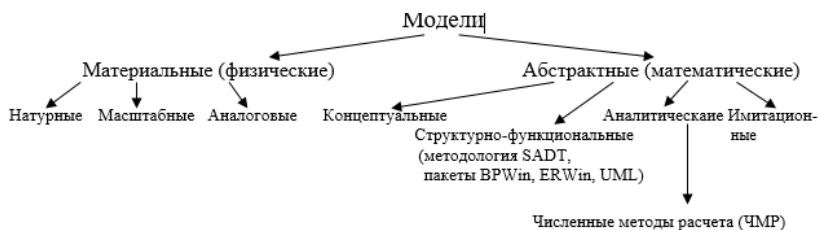


Рисунок 1 – Классификация по степени абстрагирования модели от оригинала

В аналоговых моделях исследуемые процессы изучаются на основании процессов, имеющих иную физическую природу, но описываемых с помощью одинаковых математических соотношений. Например, аналогичными уравнениями описываются колебания груза на пружине, колебания тока в электрическом контуре, движения маятника.

**Математическая модель** – это формальная зависимость между значениями параметров на входе моделируемого объекта или процесса и выходными параметрами. При математическом моделировании абстрагируются от конкретной физической природы объекта и происходящих в нем процессов. Математическая модель представляет собой формализованное описание исследуемой системы с помощью абстрактного языка, в котором могут использоваться математические средства (алгебраические, дифференциальные, интегральные уравнения и т.п.), а также средства языков программирования общего назначения и специализированных языков.

**Концептуальная модель** описывается с помощью специальных символов, знаков, операций над ними или с помощью естественного или искусственного языков.

**Структурно-функциональная модель** – это образ объекта исследования, описанный с помощью взаимосвязанных таблиц, блок-схем, диаграмм, графиков, рисунков, анимационных фрагментов, гипертекстов и т.п. и отображающий структуру элементов объекта и взаимосвязи между ними. Структурно-функциональная модель уточняет и дополняет концептуальную модель.

Среди математических моделей можно выделить аналитические и имитационные.

Для **аналитического моделирования** характерно то, что процессы функционирования элементов системы записываются в виде функциональных соотношений. Аналитическая модель может быть исследована:

- аналитическими методами – для получения в общем виде явных зависимостей искомым характеристик;
- численными методами – если уравнение не решается в общем виде, с помощью численных методов расчета (ЧМР) получают множество частных решений для конкретных исходных данных.

К недостаткам аналитического моделирования следует отнести существенное упрощение первоначальной модели, которое производят, чтобы изучить хотя бы общие свойства системы.

Численные методы позволяют исследовать по сравнению с аналитическим методом более широкий класс систем, но при этом полученные решения носят частный характер. Численный метод расчета аналитических задач называется *методом статистических испытаний (методом Монте-Карло)*.

**Имитационная модель** – совокупность описания статических свойств системы, а также динамики изменения состояния и взаимосвязей элементов системы во времени под влиянием факторов внешней среды и внутренней логики функционирования

системы. Имитационная модель задается совокупностью моделирующих алгоритмов, воспроизводящих (копирующих) с заданной степенью детализации развитие физических процессов, происходящих в исследуемой системе, т.е. задающих модель поведения.

При **имитационном моделировании** алгоритм, реализующий модель, воспроизводит процесс функционирования реальной системы  $S$  во времени с заданной степенью детализации, причем элементарные составляющие процесса имитируются с сохранением их логической структуры и последовательности протекания во времени.

Основные характеристики и преимущества имитационных моделей (ИМ):

- удобство учета в моделях дискретных, непрерывных, нелинейных и стохастических факторов;
- повторяемость – любая реализация процесса функционирования системы может быть многократно повторена;
- управляемость – возможность остановки и реактивации модели, а также возможность исследования процесса функционирования системы  $S$  в любых условиях, в том числе, в экстремальных режимах работы;
- доступность и гибкость – возможность изменить любые характеристики модели, в том числе, варьировать структуру, алгоритмы и параметры моделируемой системы, что важно с точки зрения поиска оптимального варианта системы;
- наглядность – применение средств визуализации.

Недостатки имитационных моделей:

- решение, полученное с помощью имитационной модели  $M$ , всегда носит частный характер в соответствии с фиксированными

элементами структуры, алгоритмами поведения и параметрами моделируемой системы, а также внешними воздействиями.

Если результаты, полученные при воспроизведении на имитационной модели процесса функционирования системы  $S$ , являются реализациями случайных величин и функций, для нахождения характеристик процесса требуется его многократное воспроизведение с последующей статистической обработкой результатов. В этом случае целесообразно в качестве метода компьютерной реализации имитационной модели использовать метод *статистического моделирования*. Таким образом, методом *статистического моделирования* называется метод компьютерной реализации имитационной модели, а методом *статистических испытаний* (Монте-Карло) – численный метод решения аналитической задачи.

### 1.3. Формальная модель объекта исследования (реальной системы)

Модель объекта исследования можно представить в виде множества величин, описывающих процесс функционирования реальной системы  $S$  и образующих следующие подмножества [5, 6]:

$x_i \in X, i = \overline{1, n_X}$  – совокупность входных воздействий на систему;

$v_l \in V, l = \overline{1, n_V}$  – совокупность воздействий внешней среды;

$h_k \in H, k = \overline{1, n_H}$  – совокупность внутренних параметров системы;

$y_j \in Y, j = \overline{1, n_Y}$  – совокупность выходных характеристик системы.

При моделировании системы  $S$  входные воздействия, воздействия внешней среды и внутренние параметры системы являются *независимыми (экзогенными) переменными*, а выходные

характеристики системы – *зависимыми (эндогенными) переменными.*

Переменные и параметры модели в векторной форме, т.е. с учетом параметра времени, имеют вид:

$$\vec{x}(t) = (x_1(t), x_2(t), \dots, x_{nX}(t)),$$

$$\vec{v}(t) = (v_1(t), v_2(t), \dots, v_{nV}(t)),$$

$$\vec{h}(t) = (h_1(t), h_2(t), \dots, h_{nH}(t)),$$

$$\vec{y}(t) = (y_1(t), y_2(t), \dots, y_{nY}(t)).$$

Процесс функционирования системы  $S$  во времени описывается оператором  $F_S$ , который преобразует экзогенные переменные в эндогенные в соответствии с соотношением (1).

$$\vec{y}(t) = F_S(\vec{x}, \vec{v}, \vec{h}, t). \quad (1)$$

Соотношение (1) определяет закон функционирования системы, который реализуется с помощью алгоритма функционирования  $A$ , под которым понимается метод получения выходных характеристик с учетом входных воздействий  $\vec{x}(t)$ , воздействий внешней среды  $\vec{v}(t)$  и параметров системы  $\vec{h}(t)$ .

#### 1.4. Понятие состояния системы

Свойства системы  $S$  в конкретные моменты времени называются *состояниями*. Процесс функционирования системы во времени можно рассматривать как последовательную смену состояний системы  $z_1(t), z_2(t), \dots, z_k(t), k=1..nZ$ . Совокупность всех возможных состояний  $\{z_k(t)\}$  называется пространством состояний  $Z$  объекта моделирования, причем  $z_k(t) \in Z$ .

С помощью векторных уравнений (2) и (3) состояние системы  $S$  в произвольный момент времени  $t_0 < t^* \leq T$  полностью определяется

начальным состоянием  $\vec{Z}^0 = (Z_1^0, Z_2^0, \dots, Z_k^0)$ , где  $Z_1^0 = Z_1(t_0), Z_2^0 = Z_2(t_0), \dots, Z_k^0 = Z_k(t_0)$ , а также входными воздействиями  $\vec{x}(t)$ , внутренними параметрами  $\vec{h}(t)$ , воздействием внешней среды  $\vec{v}(t)$  на интервале времени  $(t^* - t_0)$ .

Соотношение (2) позволяет определить состояние системы в текущий момент времени по начальному состоянию системы и по экзогенным переменным.

$$\vec{Z}(t) = \phi(\vec{Z}^0, \vec{x}, \vec{v}, \vec{h}, t). \quad (2)$$

Соотношение (3) позволяет определить выходные характеристики системы по её состоянию.

$$\vec{y}(t) = F(\vec{Z}, t). \quad (3)$$

Соотношение (4) – это уравнение в общем виде для определения выходных характеристик системы (значение времени входит в формулу (4) один раз).

$$\vec{y}(t) = F[\phi(\vec{Z}^0, \vec{x}, \vec{v}, \vec{h}, t)]. \quad (4)$$

Время в модели системы  $S$  может рассматриваться на интервале моделирования  $(0, T)$  как в непрерывной, так и в дискретной форме, т.е. в виде квантов длиной  $\Delta t$ , где  $m$  – число интервалов дискретизации (соотношение (5)).

$$T = m\Delta t, \quad m = \overline{1, m_T}. \quad (5)$$

## 1.5. Структура имитационной модели

В общем случае, объектом-оригиналом может быть любая естественная или искусственная система, предназначенная для выполнения некоторого определенного вида работ или решения определенного класса задач. Она состоит из компонентов  $Co$ , называемых также элементами или подсистемами, имеет некоторое множество параметров (атрибутов компонентов)  $Ho$ . Система проявляет свои свойства под влиянием входных воздействий  $Xo$ , воздействий внешней среды  $Vo$  и в результате физических процессов  $Ao$ , определяющих принципы ее функционирования. Количественной мерой свойств системы служит множество характеристик  $Yo$ . Элементы множества характеристик  $y_{oj} \in Y_o$  являются частными показателями эффективности системы. Показатель эффективности – это числовая характеристика системы, которая оценивает степень приспособленности системы к выполнению поставленных перед ней задач, т.е. качество ее функционирования. (В данном абзаце «о» означает характеристики объекта-оригинала).

Структуру имитационной модели можно представить в виде совокупности множеств:

$$\text{ИМ} = \langle C_M, X_M, V_M, H_M, Y_M, F_M, A_M, O_M, E_M, T_M \rangle,$$

где  $C_M = \{C_{M_r}\}, r = \overline{1, n_{C_M}}$  – множество компонентов модели;

$X_M = \{x_{M_i}\}, i = \overline{1, n_{X_M}}$  – множество входных воздействий на модель;

$V_M = \{v_{M_l}\}, l = \overline{1, n_{V_M}}$  – множество воздействий внешней среды;

$H_M = \{h_{M_k}\}, k = \overline{1, n_{H_M}}$  – множество внутренних параметров модели;



$Y_M = \{y_{M_j}\}, j = \overline{1, n_{Y_M}}$  – множество выходных характеристик модели или множество частных показателей эффективности;

$F_M = \{f_{M_z}\}, z = \overline{1, n_{F_M}}$  – функциональные зависимости, которые описывают поведение переменных и параметров в пределах компонента или определяют соотношения между компонентами модели;

$A_M = \{a_{M_d}\}, d = \overline{1, n_{A_M}}$  – множество алгоритмов, задающих модель поведения объекта моделирования;

$O_M = \{o_{M_q}\}, q = \overline{1, n_{O_M}}$  – множество ограничений, т.е. устанавливаемых пределов изменения управляемых переменных: естественные ограничения являются следствиями свойств самой моделируемой системы, искусственные ограничения вносятся разработчиком модели;

$E_M$  – целевая функция или функция критерия (6).

$$E_M = E_M(Y_M) \rightarrow opt. \quad (6)$$

$T_O = m * T_M$ .  $T_M$  – модельное время, т.е. время, в течение которого на модель оказываются входные воздействия и измеряются выходные характеристики. Модельное время  $T_M$  связано с временем реальной системы  $T_o$  посредством масштабного коэффициента  $m$ .

## 1.6. Области применения имитационных моделей

Имитационное моделирование сложных систем часто является единственным практически реализуемым методом исследования процесса функционирования таких систем, особенно на этапе их

проектирования. Имитационное моделирование целесообразно использовать в следующих случаях:

1. Для исследования системы  $S$  до того, как она спроектирована, с целью получения первоначальных оценок её структуры, характеристик, алгоритмов поведения.

2. На этапе проектирования для анализа и синтеза различных вариантов системы и выбора того варианта, который в наилучшей степени удовлетворяет критерию эффективности системы при заданных ограничениях.

3. На этапе эксплуатации системы для получения оценок её функционирования и составления прогноза эволюции системы.

### **Задача анализа**

Задача анализа включает задачи получения оценок:

- вариантов структуры системы;
- эффективности различных алгоритмов управления системой;
- влияния изменения различных параметров системы.

*Дано:* Структура системы:  $C_M$ ,  $F_M$ ,  $A_M$ , значения внутренних параметров  $H_M$ , внешние воздействия  $X_M$ .

*Определить:* выходные характеристики – значения показателей эффективности  $Y_M$ .

### **Задача синтеза**

Задачи синтеза подразделяются на структурный, алгоритмический, параметрический синтез. Требуется создать систему с заданными характеристиками  $Y_M$  при известных внешних воздействиях  $X_M$  и  $O_M$ , которая являлась бы оптимальной по некоторым показателям эффективности при заданных ограничениях.

Дано:  $Y_M, X_M, O_M, E_M$ .

Определить:  $C_M, F_M, A_M, H_M$ .

На практике решение задачи синтеза сводится к многократному целенаправленному анализу модели системы.

При рассмотрении показателей эффективности используются однокритериальная и многокритериальная оценки.

**Однокритериальная оценка.** Показатель эффективности системы оценивается по одному частному критерию  $Y_{opt}$ , а по остальным накладываются ограничения (7).

$$E_M = Y_{Mopt}, \quad (7)$$
$$Y_{Mjmin} \leq Y_{Mj} \leq Y_{Mjmax}.$$

**Многокритериальная оценка.** Целевая функция (6) представляется в форме интегрального критерия, который связывает достаточно простой зависимостью показатель эффективности системы со всеми частными показателями эффективности. Наиболее распространенным является нормированный аддитивный критерий (8).

$$E_M = \sum_{j=1}^n b_j \gamma(Y_M) \rightarrow opt. \quad (8)$$

Функции  $\gamma(Y_{Mj}) \in [0, 1]$  подобраны так, чтобы исключить размерность  $j$ -й характеристики.

Весовые коэффициенты  $b_j$  согласуют шкалы измерения различных характеристик:  $\sum_{j=1}^n b_j = 1, b_j > 0$ .

В частном случае  $0 \leq Y_{mj} \leq Y_{mj \max}$  функции  $\gamma(Y_{Mj})$  принимают линейную форму (9):

$$\gamma(Y_M) = \frac{Y_{Mj}}{Y_{Mj \max}}. \quad (9)$$

## 1.7. Основные этапы моделирования систем

Выделяются следующие этапы моделирования систем [5, 6]:

1. Построение концептуальной модели системы и её формализация. Основное назначение первого этапа – переход от содержательного описания исследуемой системы  $S$  к её математической модели  $M$ , т.е. формализация.

1.1. Постановка задачи моделирования системы:

- определение структуры системы  $S$  и алгоритмов её поведения, воздействий внешней среды;
- выбор методики решения задачи с учетом имеющихся ресурсов;
- определение возможности разбиения на подзадачи.

1.2. Определение требований к исходной информации об объекте моделирования и организация её сбора:

- выбор необходимой информации о системе  $S$  и внешней среде  $V$ ;
- подготовка априорных данных;
- анализ имеющихся экспериментальных данных;
- выбор методов и средств предварительной обработки информации о системе.

1.3. Определение параметров и переменных модели:

- $x_i \in X, i = \overline{1, n_X}$  – совокупность входных воздействий на систему;
- $v_l \in V, l = \overline{1, n_V}$  – совокупность воздействий внешней среды;
- $h_k \in H, k = \overline{1, n_H}$  – совокупность внутренних параметров системы;
- $y_j \in Y, j = \overline{1, n_Y}$  – совокупность выходных характеристик системы  $Y_j$ .

Описание каждого параметра должно выполняться в следующей форме:

- определение и краткая характеристика;
- символ обозначения и единица измерения;
- диапазон изменения;
- место применения в модели.

1.4. Выбор критериев эффективности системы, которые являются функциями параметров и переменных системы. Критерии могут быть частные или интегральные.

1.5. Описание концептуальной модели системы  $MX$ . Это представление свойств процесса функционирования системы с использованием абстрактных терминов и понятий. Если возможно, производится выбор процедур аппроксимации реальных процессов, протекающих в системе  $S$ : детерминированная, вероятностная аппроксимация или определение средних значений. Концептуальная модель обычно разбивается на блоки, представляющие собой описание некоторых подпроцессов общего процесса функционирования системы  $S$ , воздействий внешней среды  $V$  и т.п.

2. Алгоритмизация модели системы и её компьютерная реализация.

2.1. Построение логической модели системы  $MM$ . Модель функционально подразделяется на подмодели, соответствующие автономным подпроцессам процесса функционирования системы  $S$ , выделенным в концептуальной модели.

2.2. Получение математических соотношений. Модель разбивается на  $m$  подмоделей, для каждой из которых формулируется математическое соотношение, являющееся моделью подпроцесса функционирования системы (10).

$$\begin{aligned}y_1(t) &= f_1(x_1, x_2, \dots, x_{nX}; v_1, v_2, \dots, v_{nV}; h_1, h_2, \dots, h_{nH}; t) \\y_2(t) &= f_2(x_1, x_2, \dots, x_{nX}; v_1, v_2, \dots, v_{nV}; h_1, h_2, \dots, h_{nH}; t) \\&\dots \\y_m(t) &= f_m(x_1, x_2, \dots, x_{nX}; v_1, v_2, \dots, v_{nV}; h_1, h_2, \dots, h_{nH}; t)\end{aligned} \quad (10)$$

Математические соотношения (10) в виде явных функций, т.е. аналитическая модель, определяются, если это возможно. В общем случае, логическая модель  $M_M$ , имеющая комбинированный характер, что отражает аналитико-имитационный подход, должна содержать:

- описание всех блоков с их наименованиями;
- единую систему обозначений и нумерацию блоков;
- отражение логики модели процесса функционирования системы.

2.3. Выбор инструментальных средств моделирования, обеспечивающих следующие возможности:

- наличие необходимых программных и технических средств;
- обеспечение всех этапов реализации модели;
- возможность своевременного получения результатов моделирования.

2.4. Спецификация и построение схемы алгоритма программы. Спецификация – это формализованное представление требований, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также описание задачи и ожидаемых результатов без указания способа их достижения. Схема алгоритма программы строится на основе логической модели  $M_M$ .

2.5. Программирование модели.

2.6. Проверка достоверности программы:

- тестирование отдельных частей программы;
- комплексное тестирование всей программной модели на контрольном примере одного из вариантов организации системы  $S$ .

3. Получение и интерпретация результатов моделирования системы.

3.1. Планирование компьютерного эксперимента с моделью системы. Планирование заключается в том, чтоб определить стратегию и тактику получения максимального объема информации при минимальных затратах ресурсов. На этапе стратегического планирования ставится задача построения оптимального плана эксперимента в соответствии с целью моделирования. Тактическое планирование определяет оптимальные пути достижения целей, поставленных в стратегическом планировании.

3.2. Выполнение рабочих расчетов. Они включают в себя:

- подготовку наборов исходных данных;
- проведение расчетов на контрольном примере с целью определения чувствительности результатов к изменениям исходных данных;
- проведение рабочих расчетов с целью получения выходных характеристик модели.

### 3.3. Анализ результатов моделирования:

- представление результатов моделирования в наглядной форме;
- вычисление статистических характеристик результатов.

3.4. Интерпретация результатов моделирования, т.е. переход от информации, полученной на модели  $M$ , к практическим рекомендациям использования результатов моделирования в реальной системе  $S$  на этапах её проектирования, исследования, эксплуатации.

## 2. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

### 2.1. Объект исследования

Рассмотрим в качестве предметной области исследований *вычислительные системы с разделением времени*.

Один из рациональных методов использования вычислительной техники состоит в создании вычислительных систем с разделением времени (СРВ) – систем коллективного пользования, систем с множественным доступом. СРВ – мультипрограммные системы, в которых в каждый момент времени может находиться несколько программ – заданий на обработку данных. При проектировании таких систем необходимо решить весьма сложную задачу конструирования алгоритмов динамического распределения многих ресурсов системы (средств связи, вычислительных средств обработки, запоминающих устройств) между отдельными заданиями (*заявками*) на обработку данных, присутствующими в СРВ.

При создании СРВ требуется, чтобы распределительный алгоритм, во-первых, *обнаруживал* «короткие» заявки (задания, требующие для своего выполнения небольших ресурсов времени

[Перейти к оглавлению](#)



центрального процессора (ЦП)), во-вторых, *отдавал предпочтение* обработке таких заявок за счет более «длинных» (требующих длительного обслуживания).

Любой распределительный алгоритм характеризуется *дисциплиной диспетчеризации* (ДД), определяющей порядок обслуживания заявок. В СРВ используется множество разнообразных дисциплин.

Изучение алгоритмов распределения времени центрального процессора в системах коллективного пользования с применением методов моделирования является важной практической задачей.

## **2.2. Описание задачи исследования**

Необходимо выполнить содержательную постановку задачи, определить цели исследования, внешние воздействия и ограничения, которые накладываются на систему. Целью задачи исследования является разработка модели распределения времени центрального процессора (ЦП) и последующая оценка распределения времени ответа и среднего времени ответа путем имитации обслуживания  $N$  заявок.

Эффективность использования того или иного алгоритма определяется не только его структурой, но и тем, каковы интервалы времени между поступлениями новых заявок в систему и какое время требуется для обслуживания каждой заявки. Поэтому анализ качества распределительного алгоритма проводится при заданных характеристиках входного потока заявок на обслуживание. Описание входного потока должно использовать адекватные вероятностные модели информационных потоков [7].

Мерой эффективности СРВ являются характеристики времени пребывания заявки в системе (т.е. *времени ответа*), в частности:

- распределение времени ответа  $S(t) = p\{\text{время ответа} \leq t\}$ ;
- среднее время ответа  $T$ .

Время ответа для каждой заявки, поступающей в СРВ, определяется разностью между моментом времени, когда она покидает систему, и моментом ее поступления.

Исходными данными для исследования являются:

- множество  $(DD, k, q)$ , описывающее распределительный алгоритм моделируемой СРВ:

- $DD$  – дисциплина диспетчеризации;
- $k$  – количество квантов, выделяемых для обслуживания заявки;
- $\nabla$  – символ, определяющий квант ЦП, величина которого равна времени полного обслуживания заявки;
- $k_{\nabla}$  – количество квантов, достаточное для полного обслуживания;
- $q$  – длительность кванта;
- входной поток заявок на обслуживание;
- требуемое время для обслуживания заявок.

Одной из наиболее распространенных дисциплин диспетчеризации является *круговой опрос (КО)* (Рисунок 2). Вновь поступающие заявки становятся в единственную очередь, продвигаются к ее началу в порядке поступления и получают квант обслуживания  $q$ . Если предоставленный квант для удовлетворения заявки недостаточен, то заявка возвращается в конец той же очереди и цикл повторяется.

Результатом исследования являются характеристики времени пребывания заявки в системе (т.е. *времени реакции системы*).

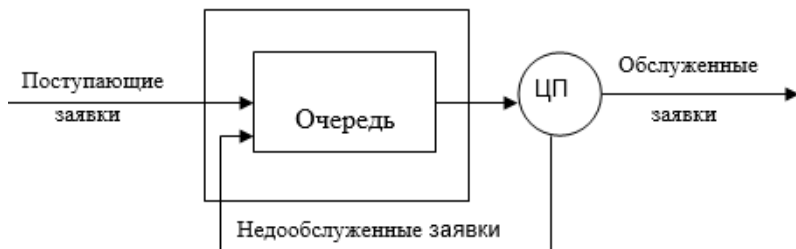


Рисунок 2 – Схема системы с круговым опросом

### 2.3. Формализованная постановка задачи исследования

Задача исследования формулируется следующим образом.

Разработать GPSS-модель распределения времени центрального процессора и выполнить оценку распределения времени ответа и среднего времени ответа путем прогона GPSS-программы на компьютере с имитацией обслуживания заданного количества заявок на основе следующих исходных данных.

1. Дисциплина диспетчеризации центрального процессора – (КО,  $k$ ,  $q$ ).
2. Входной поток заявок является простейшим. Промежутки времени между заявками имеют экспоненциальное распределение с параметром  $\lambda$ .
3. Требуемое время обслуживания заявок распределено по закону Эрланга  $k$ -го порядка с параметром  $\alpha$ .
4. Значения параметров (Таблица 1):

Таблица 1 – Значения параметров модели

Значение параметров			
$\lambda[\text{с}^{-1}]$	$k$	$\alpha[\text{с}^{-1}]$	$q[\text{с}]$
10	2	1	0.25

В процессе компьютерного эксперимента необходимо определить суммарное время обслуживания всех заявок с учетом времени пребывания их в очереди, а затем получить оценку распределения времени обслуживания заявки и среднего времени обслуживания заявки.

Таким образом, в процессе исследования необходимо ознакомиться с алгоритмом распределения времени центрального процессора в системах коллективного пользования и получить практический опыт моделирования вычислительных систем.

## **2.4. Анализ возможных методов решения**

После того как сформулирована задача исследования, необходимо проанализировать методы решения поставленной задачи, указать их преимущества и недостатки, обосновать выбор метода решения. Если задача может быть решена с использованием какого-либо аналитического аппарата при определенных упрощающих предположениях, приводится вариант такого расчета, впоследствии его результаты сравниваются с результатами цифровой имитации.

### ***2.4.1. Аналитический метод***

Для получения аналитического решения задач данного класса введем следующие упрощающие предположения:

1. Задачу исследования дисциплины диспетчеризации центрального процессора можно свести к рассмотрению одноканальной системы массового обслуживания (СМО).
2. Заявки, получающие первый квант обслуживания, и недообслуженные заявки образуют общую очередь входных сообщений, которая является неограниченной.

3. Входной поток заявок на обслуживание является пуассоновским.
4. Интервалы времени обслуживания распределены по экспоненциальному закону.

Для одноканальной разомкнутой СМО с неограниченной очередью входных сообщений время пребывания заявки в системе вычисляется по формулам (11) [7]:

$$T = \frac{\rho}{\mu(1-\rho)} + \frac{1}{\rho}, \quad (11)$$
$$\rho = \frac{\lambda}{\mu},$$

где  $\lambda$  – интенсивность потока заявок на обслуживание;  
 $\mu$  – интенсивность обслуживания;  
 $\rho$  – загрузка системы.

Если принять упрощающие предположения, введенные выше, то при значениях  $\lambda = 10 \text{ с}^{-1}$ ,  $\mu = 1 \text{ с}^{-1}$  загрузка системы  $\rho = 10$ . Известно, что при  $\rho > 1$  установившегося режима в системе не существует и очередь растет неограниченно, т.к. средняя продолжительность обслуживания  $1/\mu$  больше средней длины интервала  $1/\lambda$  между поступлениями заявок [7]. Для неустановившегося режима рассчитать параметры системы аналитически невозможно.

#### ***2.4.2. Метод имитационного моделирования***

Основным преимуществом имитационного моделирования по сравнению с аналитическим является возможность решения более сложных задач. Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов

системы, случайные воздействия, которые часто создают трудности при аналитических исследованиях. Метод имитационного моделирования позволяет решать задачи анализа больших систем, включая задачи оценки: вариантов структуры системы, эффективности различных алгоритмов управления, влияния изменения различных параметров системы. Имитационное моделирование может быть положено также в основу структурного, алгоритмического и параметрического синтеза больших систем, когда требуется создать систему, с заданными характеристиками при определенных ограничениях, которая является оптимальной по некоторым критериям оценки эффективности.

Основным недостатком, проявляющимся при компьютерной реализации метода имитационного моделирования, является то, что решение, полученное при анализе имитационной модели, всегда носит частный характер, так как оно соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы, начальных условий и воздействий внешней среды.

Однако, имитационное моделирование сложных систем часто является единственным практически реализуемым методом исследования процесса функционирования сложных систем, особенно на этапе проектирования.

Хотя указанные в п. 2.2 характеристики времени ответа при определенных упрощающих предположениях рассчитываются аналитически [5, 7], однако в общем случае их определение связано с использованием метода имитационного моделирования.

### **3. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ**

Рассмотрим этапы разработки имитационной модели ИВС.

### 3.1. Разработка структурно-функциональной модели

Основным назначением этого этапа является переход от содержательного описания объекта исследования к его формализованному описанию в виде графического образа, отображающего его структуру. Структурно-функциональная модель описывает моделируемую систему на верхнем уровне абстракции и отображает последовательность элементов различных типов, составляющих модель, и связи между ними. На Рисунок 3 показана структурно-функциональная модель распределения времени центрального процессора для дисциплины диспетчеризации кругового опроса согласно постановке задачи исследования (п. 2.3).

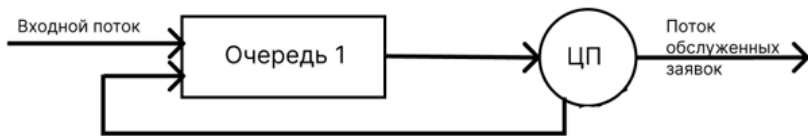


Рисунок 3 – Структурно-функциональная модель

### 3.2. Разработка концептуальной модели

При разработке концептуальной модели необходимо:

- определить цели моделирования;
- описать входные, выходные переменные и параметры модели;
- представить функциональные зависимости, описывающие поведение переменных и параметров;

- описать ограничения на возможные изменения величин;
- выбрать степень детализации представленной модели (определить масштаб времени);
- сформулировать целевые функции (критерии эффективности) моделируемой системы.

На этапе построения концептуальной модели и ее формализации разрабатывается модель объекта исследования и строится ее формальная схема, т.е. основным назначением этого этапа является переход от содержательного описания объекта к его формализованному описанию в виде математической модели.

Ниже описаны объект моделирования, внешние воздействия, входные переменные, параметры модели и выходные переменные модели.

*Цель моделирования* – разработать модель распределения времени центрального процессора и выполнить оценку распределения времени ответа и среднего времени ответа.

*Объект моделирования* – центральный процессор, для которого определена дисциплина диспетчеризации.

*Параметры модели:*

- дисциплина диспетчеризации – одноуровневая дисциплина кругового опроса;
- величина кванта обслуживания  $q$ .

*Внешние воздействия:*

- простейший входной поток заявок на обслуживание с функцией плотности вероятностей (13)

$$f(x) = \lambda e^{-\lambda x}, x \geq 0, \lambda \geq 0. \quad (13)$$



*Входные переменные:*

- параметр  $\lambda$  экспоненциального распределения интервалов времени между заявками на обслуживание;
- параметр  $\alpha$  распределения Эрланга с плотностью вероятностей (14), описывающего требуемое время обслуживания заявки;
- порядок  $k$  распределения Эрланга, описывающего требуемое время обслуживания заявки (14).

$$f(\tau_3) = \alpha \frac{(\alpha\tau)^{k-1}}{(k-1)!} e^{-\lambda\tau}. \quad (14)$$

*Выходные переменные:*

- среднее время обслуживания заявок  $T$ .
- Концептуальная модель содержит:
- таблицу функционального соответствия элементов моделируемой системы и элементов модели, включая содержательное описание всех статических и динамических объектов с описанием их свойств;
  - описание алгоритмов формирования случайных величин.

### ***3.2.1. Таблица функционального соответствия элементов модели и элементов реальной системы***

Функциональное соответствие между элементами реальной системы и элементами модели должно быть зафиксировано в таблице функционального соответствия (Таблица 2).

Таблица 2 – Таблица функционального соответствия

Элемент реальной системы	Элемент модели
1 ед. реального времени	100 ед. модельного времени
Обслуживание заявок	Процесс
Заявка	Транзакт процесса
Очередь заявок	Очередь №1
Центральный процессор	Одноканальное устройство CPU
Интервалы времени между поступлениями заявок в простейшем потоке	Модификатор-функция
Функция с экспоненциальным распределением и параметром $\lambda = 1$	Функция EXPON
Требуемое время обслуживания заявки	Переменная ERL
Требуемое время обслуживания заявки для организации моделирования	Значение параметра №1 транзакта процесса – СЧА P1
Продолжительность обслуживания заявки в системе	Продолжительность пребывания в модели транзакта процесса – СЧА M1
Статистика о продолжительности обслуживания заявок в системе	Таблица TAB

### 3.2.2. Алгоритмы моделирования случайных величин

1. Для того чтобы перейти к целочисленным значениям, необходимо ввести масштаб времени: 1 единица реального времени  $\rightarrow$  100 единиц модельного времени.  
 $q = 0.25$ , с учетом масштаба времени  $q = 25$ .
2. Для моделирования случайной величины, распределенной по экспоненциальному закону, следует использовать метод обратной функции (15).

$$x_i = -\frac{1}{\lambda} \ln r_i, r_i \sim [0..1). \quad (15)$$

3. Интервал времени между событиями в простейшем потоке – случайная величина, распределенная по

экспоненциальному закону  $T_{\text{эксп}}$  с параметром  $\lambda_{\text{эксп}} \neq 1$ , моделируется в соответствии с (16)

$$T_{\text{эксп}} = \frac{1}{\lambda} * FN\$EXPON. \quad (16)$$

$\lambda_{\text{эксп}} = 10$ ,  $1/\lambda_{\text{эксп}} = 0.1$ , с учетом масштаба времени  
 $1/\lambda_{\text{эксп}} = 10$ .

Случайная величина, распределенная по экспоненциальному закону, моделируется с помощью модификатора-функции:

GENERATE 10, FN\$EXPON

4. При моделировании случайных величин, распределенных по закону Эрланга с параметром  $\alpha$ , используют то обстоятельство, что этому закону распределения подчиняется сумма  $k$  независимых реализаций случайной величины, распределенной по экспоненциальному закону с параметром  $\lambda_{\text{эксп}} = k * \alpha$ .

$$T_{\text{эрл}} = \sum_{i=1}^k T_{\text{эксп}}. \quad (17)$$

В этом случае величина  $T_{\text{эрл}}$  распределена по закону Эрланга  $k$ -го порядка с плотностью вероятности (14).

Продолжительность обслуживания – случайная величина  $T_{\text{эрл}}$ , распределенная по закону Эрланга  $k = 2$ ,  $\lambda_{\text{эрл}} = 1$ .

От параметра распределения Эрланга необходимо перейти к параметру экспоненциального распределения  $\lambda_{\text{эксп}} = 2 * 1 = 2$ .

Для моделирования случайной величины, распределенной по экспоненциальному закону, необходима величина среднего времени между событиями, т.е.  $1/\lambda_{\text{экс}} = 1/2 = 0.5$ ,  
 $1/\lambda_{\text{экс}} = 50$  с учетом масштаба времени.

$$T_{\text{эрл}} = T_{\text{экс}} + T_{\text{экс}} = 1/\lambda_{\text{экс}} * \text{FN\$EXPON} + 1/\lambda_{\text{экс}} * \text{FN\$EXPON} = 1/\lambda_{\text{экс}} * (\text{FN\$EXPON} + \text{FN\$EXPON})$$

Для моделирования такого выражения необходимо использовать вещественную переменную.

Случайная величина, распределенная по закону Эрланга, моделируется с помощью переменной VV:

$$VV \quad \text{FVARIABLE} \quad 50\#(\text{FN\$EXPON} + \text{FN\$EXPON}).$$

### 3.3. Выбор программных средств моделирования

После разработки концептуальной модели необходимо обосновать выбор программного средства моделирования. При выборе программных средств моделирования следует руководствоваться наличием у программного средства следующей функциональности:

- обеспечение всех этапов реализации модели;
- возможность моделирования стохастических факторов;
- управление модельным временем;
- возможность описания сложных динамических процессов, происходящих в реальной системе;
  - наличие средств визуализации процесса моделирования;
  - планирование и управление имитационным экспериментом;
  - сбор статистики о характеристиках исследуемой системы с последующим проведением статистического анализа их значений.

При кратком описании выбранного программного средства необходимо указать:

- существует ли доступная инструкция для пользователя программного средства;
- имеющиеся средства генерации случайных величин;
- возможности отладки программной реализации модели;
- средства сбора статистических данных о работе модели;
- возможности отображения структуры моделируемой системы;
- возможности редактирования модели.

Для решения задачи исследования распределения времени обслуживания заявки центральным процессором выбран язык имитационного моделирования GPSS [8-13]. GPSS – универсальный язык моделирования систем с дискретными событиями. Язык GPSS является средством спецификации моделируемой системы и в этом качестве характеризуется возможностью разрабатывать структурные описания, задавать правила и алгоритмы функционирования моделируемой системы, концептуальной эффективностью. Язык GPSS является также средством программирования, средством интерпретации модели системы в компьютерной среде и, с этой точки зрения характеризуется вычислительной эффективностью, удобством разработки, отладки и использования модели.

Модели систем на GPSS могут быть представлены в виде блок-схем или последовательности инструкций, эквивалентных блок-схеме. В основе реализации GPSS лежит схема транзактов – динамических объектов, каждый из которых, активируя свою группу функциональных элементов языка, передвигается по блокам модели. Одновременно в модели может находиться несколько транзактов, что позволяет имитировать параллельно протекающие процессы.

Решение задачи исследования распределения времени обслуживания заявки центральным процессором реализовано в системе имитационного моделирования GPSS World, предназначенной для использования в среде Windows [8-13].

### **3.4. Разработка алгоритма моделирования и программы имитационной модели**

На данном этапе выполняются следующие действия:

- описание элементов модели;
- алгоритм моделирования;
- описание программной реализации имитационной модели;
- программа имитационной модели.

#### ***3.4.1. Описание элементов модели***

Описание элементов модели выполняется в соответствии со следующими правилами:

1. В модели необходимо соблюдать единый масштаб времени, который выбирается таким образом, чтобы значения времени были целочисленными.
2. Процесс обслуживания заявок естественно представить в GPSS-модели как перемещение транзактов. Каждый транзакт, моделирующий заявку, целесообразно снабдить параметром «требуемое время обслуживания» (рекомендуется использовать 1-й параметр транзакта). Для моделирования значения «требуемого времени обслуживания» следует использовать переменную. Начальное значение этого параметра устанавливается с помощью блока ASSIGN сразу при входе транзакта в модель. Всякий раз при имитации получения кванта обслуживания заявки параметр «требуемое

время обслуживания» следует уменьшать на величину полученного кванта. Достижение параметром нулевого значения означает, что обслуживание заявки закончено и транзакт, представляющий ее, следует вывести из модели (уничтожить).

3. Необходимо учитывать, что обслуживание заявки может завершиться до истечения очередного предоставленного ей кванта обслуживания. Тогда, если первый параметр транзакта есть «требуемое время обслуживания», а квант обслуживания равен  $q$ , то устройство, имитирующее центральный процессор, следует занимать на  $\min(P1, q)$  единиц времени.

4. Для сбора статистических данных об очередях заявок целесообразно использовать регистраторы этих очередей: блоки QUEUE и DEPART.

5. Имитация стохастических факторов, таких как входной поток заявок или продолжительность требуемого обслуживания заявки процессором связана с индивидуальным определением переменных и функций GPSS.

6. Для получения оценок распределения времени ответа  $S(t)$  и среднего времени ответа  $T$  стандартный числовой атрибут  $MI$  – «время пребывания транзакта в модели» необходимо подвергнуть статистическому анализу в момент удаления транзакта из модели. Для определения в GPSS-программе необходимой для такого анализа таблицы можно дать следующие рекомендации:

- верхняя граница левого частотного интервала должна быть не меньше величины кванта первого (высшего) уровня обслуживания заявок;
- число интервалов 15;

- ширину интервалов предварительно взять не меньше  $\min(q, m/3)$ , где  $q$  – наименьший квант обслуживания, предоставляемый заявке,  $m$  – математическое ожидание требуемого времени обслуживания заявки.

### 3.4.2. Алгоритм моделирования

Алгоритм моделирования на языке GPSS может быть представлен в виде блок-схемы – набора фигур с характерными контурами блоков языка GPSS, соединенных между собой линиями (Рисунок 4).

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number
1 GEN	GENERATE	0	0	0	5
2 ASN	ASSIGN	0	0	0	6
MET	QUEUE	0	0	0	7
4 SEI	SEIZE	0	0	0	8
5 DEP	DEPART	0	0	0	9
6 TES	TEST	0	0	0	10
7 ADV	ADVANCE	0	0	0	11
8 REL	RELEASE	0	0	0	12
9 TRA	TRANSFER	0	0	0	13
QUANT	ADVANCE	0	0	0	14
11 ASN	ASSIGN	0	0	0	15
12 REL	RELEASE	0	0	0	16
13 BUF	BUFFER	0	0	0	17
14 TRA	TRANSFER	0	0	0	18
OUT	TABULATE	0	0	0	19
16 TER	TERMINATE	0	0	0	20

Рисунок 4 – Алгоритм модели

### 3.4.3. Описание программной реализации имитационной модели

В данном разделе приводится описание каждого блока модели с комментариями к ним.

[Перейти к оглавлению](#)



В программной реализации имитационной модели распределения времени обслуживания заявки центральным процессором использованы следующие команды языка GPSS:

GENERATE – сгенерировать поток транзактов в соответствии с заданным законом распределения вероятностей случайных величин – интервалов времени между поступлениями транзактов;

ASSIGN – присвоить или модифицировать значение параметра транзакта;

SEIZE – занять одноканальное устройство;

RELEASE – освободить одноканальное устройство;

QUEUE – зарегистрировать транзакт в очереди;

DEPART – вывести транзакт из очереди;

ADVANCE – задержать движение транзакта на заданный интервал времени;

TEST – изменить направление движения транзакта в зависимости от выполнения логического условия, заданного на множестве СЧА;

TRANSFER – направить транзакт к заданному блоку;

BUFFER – просмотреть список текущих событий;

TABULATE – собрать статистику об изменении значения заданного СЧА;

TERMINATE – вывести транзакт из модели и уничтожить его.

#### ***3.4.4. Программа имитационной модели***

На данном этапе представляется листинг программы модели с комментариями.

Программная реализация имитационной модели распределения времени обслуживания заявки центральным процессором представлена на Рисунок 5.

```
lab_1.gps
EXPON    FUNCTION RN1,C6
         0,0/.1,.1/.2,.2/.5,.69/.8,1.6/.999,8.0
VV       FVARIABLE 50#(FN$EXPON+FN$EXPON)
TT       TABLE M1,1500,1500,20
         GENERATE 10, FN$EXPON,,100
         ASSIGN 1,V$VV
MET      QUEUE 1
         SEIZE CPU
         DEPART 1
         TEST LE P1,25,QUANT
         ADVANCE P1
         RELEASE CPU
         TRANSFER ,OUT
QUANT    ADVANCE 25
         ASSIGN 1-,25
         RELEASE CPU
         BUFFER
         TRANSFER ,MET
OUT      TABULATE TT
         TERMINATE 1
```

Рисунок 5 – Программа модели

### 3.5. Организация единичного эксперимента с имитационной моделью

Результатами компьютерного эксперимента с моделью ИВС являются протокол моделирования, графики, таблицы, а также качественные и количественные оценки результатов моделирования.

#### 3.5.1. Анализ результатов моделирования

Результаты моделирования представлены в протоколе моделирования (Рисунок 6) и в гистограмме распределения времени обслуживания транзактов (Рисунок 7).

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	15531.833	16	1	0

NAME	VALUE
CPU	10003.000
EXPON	10000.000
MET	3.000
OUT	15.000
QUANT	10.000
IT	10002.000
VV	10001.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	ASSIGN	100	0	0
MET	3	QUEUE	670	0	0
	4	SEIZE	670	0	0
	5	DEPART	670	0	0
	6	TEST	670	0	0
	7	ADVANCE	100	0	0
	8	RELEASE	100	0	0
	9	TRANSFER	100	0	0
QUANT	10	ADVANCE	570	0	0
	11	ASSIGN	570	0	0
	12	RELEASE	570	0	0
	13	BUFFER	570	0	0
	14	TRANSFER	570	0	0
OUT	15	TABULATE	100	0	0
	16	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CPU	670	1.000	23.172	1	0	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
1	89	0	670	5	45.643	1058.078	1066.033

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
IT	7244.370	4566.078		0		
			- - 1500.000		14	14.00
			1500.000 - 3000.000		8	22.00
			3000.000 - 4500.000		9	31.00
			4500.000 - 6000.000		15	46.00
			6000.000 - 7500.000		9	55.00
			7500.000 - 9000.000		10	65.00
			9000.000 - 10500.000		3	68.00
			10500.000 - 12000.000		9	77.00
			12000.000 - 13500.000		12	89.00
			13500.000 - 15000.000		11	100.00

Рисунок 6 – Протокол моделирования

Из раздела протокола *Facility* следует, что центральный процессор загружен на 100%. Средняя продолжительность

пребывания транзакта в очереди (раздел *Queue*) составляет 1058.078 единиц модельного времени, средняя длина очереди – 46 транзактов. Среднее время пребывания транзакта в модели (среднее время обслуживания заявки) составляет 7244.370 единиц модельного времени, стандартное отклонение – 4566.078 единиц модельного времени (раздел *Table*).

Гистограмма распределения вероятностей случайной величины – продолжительности обслуживания заявок показана на Рисунок 7.

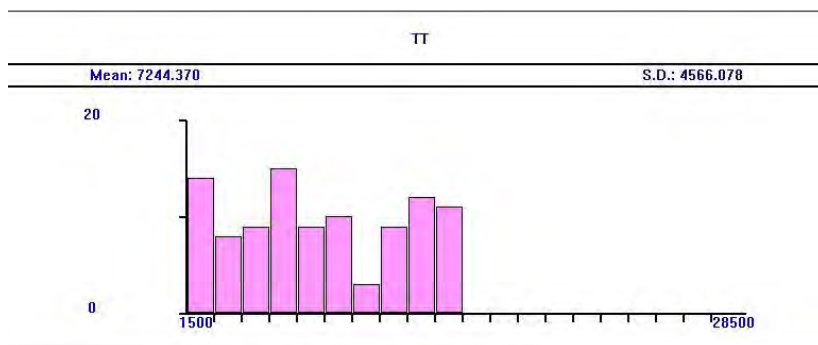


Рисунок 7 – Гистограмма распределения времени обслуживания транзактов

### 3.5.2. Оценка точности результатов моделирования

На данном этапе выполняются следующие действия:

- приводятся основные математические соотношения и зависимости, связывающие точность модели, ее достоверность и количество реализаций модели. Значение величины, подлежащей определению путем моделирования (показателя эффективности модели), должно быть оценено с помощью пробного прогона программы модели;

- определяется длительность прогонов модели, при которых гарантирована точность полученных оценок;
- осуществляется подбор тестовых данных для проверки функционирования модели во всем диапазоне исходных данных.

Если в качестве показателя эффективности используется значение *вероятности наступления какого-либо события*, число прогонов имитационной модели оценивается по формуле (18) [2]

$$N = t_{\frac{1+\delta}{2}}^2 \frac{p(1-p)}{\varepsilon^2}, \quad (18)$$

где  $p$  – вероятность наступления события;

$\varepsilon$  – точность модели (абсолютная);

$\delta$  – достоверность результатов моделирования.

Если в качестве показателя эффективности используется *среднее значение некоторой случайной величины*, число прогонов имитационной модели оценивается по формуле (19) [2]

$$N = t_{\frac{1+\delta}{2}}^2 \frac{\sigma^2}{\varepsilon^2}, \quad (19)$$

где  $\sigma$  – среднееквадратическое отклонение случайной величины;

$\varepsilon$  – точность модели (абсолютная);

$\delta$  – достоверность результатов моделирования.

В данной задаче оценивается среднее время обслуживания заявок, поэтому для расчета числа реализаций модели следует использовать формулу (19).

Пусть точность модели  $\varepsilon = 0.01$ , достоверность  $\delta = 0.95$ , квантиль нормального распределения вероятностей  $t_\alpha = 1.96$ . Из протокола моделирования и гистограммы распределения следует, что среднеквадратическое отклонение времени пребывания транзактов в модели  $\sigma = 4566.078$  единиц модельного времени (с учетом масштаба времени,  $\sigma = 45$  ед. времени). При подстановке значений в формулу (19) получим

$$N = 1.96^2 * \frac{2025}{0.0001} = 77760000.$$

Таким образом, для обеспечения заданной точности и достоверности необходимо 77760000 реализаций имитационной модели.

#### **4. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТОВ В СИСТЕМЕ GPSS WORLD**

**План имитационного эксперимента** – это метод получения с помощью эксперимента необходимой информации, стоимость которой зависит от способа сбора и обработки данных [14].

##### **4.1. Организация факторного эксперимента с моделью**

При проведении факторного эксперимента с моделью ИВС необходимо:

- определить факторы;
- определить уровни факторов;
- определить отклики;
- сформировать план факторного эксперимента в виде текстового файла.

Входные переменные модели называются *факторами*, выходные переменные называются *откликами*. Каждый фактор в эксперименте может принимать одно или несколько значений, называемых *уровнями фактора*. Множество уровней фактора определяет одно из возможных состояний моделируемой системы и представляет условия проведения одного из возможных экспериментов. Минимальное количество уровней два – нижняя и верхняя границы диапазонов изменения значений факторов. При использовании двух уровней можно определить только линейные эффекты. Для учета квадратичных эффектов необходимо использовать три уровня, для кубических – четыре уровня. Анализ значительно упрощается, если брать только равноотстоящие друг от друга значения уровней факторов. В этом случае получаем *ортогональное* разбиение уровней.

При проведении многофакторных экспериментов требуется, чтобы для дисперсионного анализа использовался ортогональный эксперимент, т.е. оценки должны быть некоррелированными. На практике использование одних и тех же значений для каждой комбинации уровней факторов при выполнении экспериментов гарантирует ортогональность.

**Факторный эксперимент** [14] представляет собой план, в котором все уровни каждого фактора встречаются в сочетании со всеми уровнями всех других факторов. Различные уровни некоторого фактора могут соответствовать качественным значениям (например, разные дисциплины обслуживания в устройстве) или количественным значениям (например, число устройств обслуживания). Применение факторного плана вместо классической схемы, согласно которой каждый раз изменяется только один фактор, имеет ряд преимуществ:

- Более полно выявляется влияние каждого фактора, поскольку они изучаются в самых различных условиях (вследствие одновременного изменения других факторов);
- Большое число комбинаций факторов, используемых в эксперименте, облегчает предсказание результатов, которые могут быть достигнуты при определенной комбинации условий;
- Если эффекты, вызываемые каждым фактором, статистически независимы, то о каждом факторе можно получить не меньше информации, чем при изменении в экспериментах только одного фактора и фиксации остальных;
- Если (как это часто бывает) различные факторы не являются независимыми, а вызывают эффекты, которые в определенной степени коррелированы, то в этом случае только факторный эксперимент может дать информацию о характере этих взаимодействий. При наличии нескольких взаимосвязанных существенных факторов обойтись без постановки факторного эксперимента невозможно.

## 4.2. Однофакторный эксперимент

Определим факторный план для **однофакторного эксперимента**. Для этого выполним следующую последовательность действий:

- в качестве фактора выберем величину кванта, обозначим этот фактор KVANT;
- выберем два уровня фактора: 10 и 25;
- присвоим фактору начальное значение 10:  
KVANT EQU 10;
- в качестве отклика определим результат эксперимента, которым является СЧА М1 – время пребывания транзакта в модели;



• сформируем план факторного эксперимента в виде текстового файла *name.txt*, структура которого приведена ниже.

Перед началом моделирования необходимо определить матрицу результатов в виде

```
NAME    MATRIX    ,R,C;
```

где *NAME* – имя матрицы;

*R* – количество строк, которое должно быть больше или равно количеству уровней фактора ( $R \geq 2$ );

*C* – количество столбцов, которое должно быть равно количеству прогонов модели для каждого уровня фактора ( $C=10$ ).

Затем необходимо создать командный текстовый файл (назовем его *KVANT\_If.txt*), содержащий *C* блоков команд для каждого уровня фактора. Каждый блок обеспечивает получение результата для заданного уровня фактора в одном прогоне модели.

План однофакторного эксперимента представлен ниже.

```
RES      MATRIX      ,5,10
```

```
;1-й уровень фактора
```

```
KVANT    EQU 10
```

```
;начальное значение множителя для ГПП
```

```
RMULT    401
```

```
;запуск процесса моделирования, NP – запрет вывода статистики
```

```
START    1,NP
```

```
;запись результата моделирования в матрицу результатов
```

```
MSAVEVALUE    RES,1,1,M1
```

```
;обнуление статистики, собранной за предыдущий прогон
```

```
;OFF - запрет обнуления матрицы результатов
```

```
CLEAR    OFF
```

RMULT	411	
START	1,NP	
MSAVEVALUE		RES,1,2,M1
CLEAR	OFF	
RMULT	421	
START	1,NP	
MSAVEVALUE		RES,1,3,M1
CLEAR	OFF	
RMULT	431	
START	1,NP	
MSAVEVALUE		RES,1,4,M1
CLEAR	OFF	
RMULT	441	
START	1,NP	
MSAVEVALUE		RES,1,5,M1
CLEAR	OFF	
RMULT	451	
START	1,NP	
MSAVEVALUE		RES,1,6,M1
CLEAR	OFF	
RMULT	461	
START	1,NP	
MSAVEVALUE		RES,1,7,M1
CLEAR	OFF	
RMULT	471	
START	1,NP	
MSAVEVALUE		RES,1,8,M1
CLEAR	OFF	
RMULT	481	
START	1,NP	
MSAVEVALUE		RES,1,9,M1
CLEAR	OFF	
RMULT	491	
START	1,NP	
MSAVEVALUE		RES,1,10,M1
CLEAR	OFF	

```

;2-й уровень фактора
KVANT    EQU 25
;С блоков команд для 2-го уровня фактора
RMULT    401
START    1, NP
MSAVEVALUE    RES, 2, 1, M1
CLEAR    OFF
RMULT    411
START    1, NP
MSAVEVALUE    RES, 2, 2, M1
CLEAR    OFF
RMULT    421
START    1, NP
MSAVEVALUE    RES, 2, 3, M1
CLEAR    OFF
RMULT    431
START    1, NP
MSAVEVALUE    RES, 2, 4, M1
CLEAR    OFF
RMULT    441
START    1, NP
MSAVEVALUE    RES, 2, 5, M1
CLEAR    OFF
RMULT    451
START    1, NP
MSAVEVALUE    RES, 2, 6, M1
CLEAR    OFF
RMULT    461
START    1, NP
MSAVEVALUE    RES, 2, 7, M1
CLEAR    OFF
RMULT    471
START    1, NP
MSAVEVALUE    RES, 2, 8, M1
CLEAR    OFF
RMULT    481
START    1, NP
MSAVEVALUE    RES, 2, 9, M1

```

```
CLEAR    OFF
RMULT    491
START    1,NP
MSAVEVALUE RES,2,10,M1
CLEAR    OFF
```

### **4.3. Проведение дисперсионного анализа в системе GPSS World**

Для определения, является ли фактор значимым, используется дисперсионный анализ **ANOVA (analysis of variance)**, который применим только к количественным факторам. С помощью него определяются количественные отклонения наблюдений от среднего значения. Если какой-либо фактор не оказывает существенного влияния на отклик, он является незначимым. Если фактор влияет на отклик, то количественное значение фактора сравнивают с оценкой изменчивости наблюдения, т.е. со стандартной ошибкой. Это делается для исключения эффектов, которые являются случайной флуктуацией.

С помощью дисперсионного анализа необходимо выполнить следующие действия:

- получить таблицу результатов дисперсионного анализа;
- с помощью библиотечной процедуры ANOVA определить, является ли каждый фактор значимым, и проанализировать влияние факторов, взаимодействие факторов и случайных помех наблюдения на результаты моделирования.

#### ***4.3.1. Дисперсионный анализ в планировании эксперимента***

Неявно в дисперсионном анализе ANOVA используется аддитивная математическая модель, определяющая компоненты

изменения в наблюдениях, которую называют статистической моделью.

Статистическая модель для анализа данных экспериментов с одним фактором  $A$  имеет вид (20) [14]:

$$y_{ik} = \mu + \alpha_i^A + e_{ik}, \quad (20)$$

где  $i$  – номер уровня;

$k$  – номер наблюдения;

$y_{ik}$  – отклик;

$\mu$  – общее среднее по всем наблюдениям;

$\alpha_i^A$  – главный эффект фактора  $A$  на уровне  $i$ ;

$e_{ik}$  – случайная ошибка наблюдения.

В этой модели общее среднее не изменяется от эксперимента к эксперименту, в отличие от ошибки. Так как в однофакторном эксперименте имеется единственный фактор, число комбинаций обработки определяется числом уровней этого фактора.

Для двух факторов  $A$  и  $B$  общая модель факторного плана имеет вид (21):

$$y_{ijk} = \mu + \alpha_i^A + \alpha_j^B + \alpha_{ij}^{AB} + e_{ijk}, \quad (21)$$

где  $i$  – номер уровня фактора  $A$ ;

$j$  – номер уровня фактора  $B$ ;

$k$  – номер наблюдения;

$y_{ijk}$  – отклик;

$\mu$  – общее среднее по всем наблюдениям;

$\alpha_i^A$  – главный эффект фактора  $A$  на уровне  $i$ ;

$\alpha_j^B$  – главный эффект фактора  $B$  на уровне  $j$ ;

$\alpha_{ij}^{AB}$  – взаимодействие фактора  $A$  на уровне  $i$  и фактора  $B$  на уровне  $j$ ;

$e_{ijk}$  – случайная ошибка наблюдения  $i$ .

Главный эффект фактора определяет долю участия фактора в значении функции отклика во время перехода фактора с нижнего уровня к верхнему. Дисперсионный анализ, основанный на статистической модели (20–21), заканчивается построением таблицы ANOVA, в которой анализируется влияние факторов  $A$  и  $B$ , взаимодействие между факторами и случайные помехи наблюдения.

С помощью ANOVA проверяется гипотеза об отсутствии влияния фактора. Для проверки гипотезы используется  $F$ -распределение Фишера. Критерий Фишера определяет отношение двух выборочных дисперсий. Если фактор существенно влияет на отклик, то  $F$ -распределение приобретает большие значения и  $F$ -статистика становится значимой. Таким образом, соответствующий фактор является *значимым*.

#### **4.3.2. Последовательность проведения дисперсионного анализа для однофакторного эксперимента**

1. Цель дисперсионного анализа – оценка влияния фактора «длительность кванта времени ЦП» на значение времени обслуживания заявки. Необходимо ввести в модель фактор  $KVANT$  (Рисунок 8), задать длительность моделирования с помощью таймера (файл *lab1\_ANOVA\_If.gps*).

2. Создать текстовый файл *KVANT\_If.txt*, содержащий план однофакторного эксперимента, в том же каталоге, что и модель *lab1\_ANOVA\_If.gps*.

3. Транслировать модель *lab1\_ANOVA\_1f.gps* с помощью команд *Command/Create Simulation*.

4. Выбрать пункт меню *Command/Custom*. Откроется диалоговое окно *Simulation Command*. В поле ввода команд ввести *include "KVANT\_1f.txt"*, нажать кнопку <OK> (Рисунок 9).

Начинается процесс моделирования, который может занять продолжительное время в зависимости от количества прогонов модели. После его завершения в нижней строке состояния главного окна появится сообщение «*The simulation has ended*».

```
lab_1_ANOVA_1f
KVANT EQU 10

FY      FUNCTION RN1, C6
        0, 0/.1, .1/.2, .2/.5, .69/.8, 1.6/.999, 8.0
VV      FVARIABLE 50#(FN$FY+FN$FY)
TT      TABLE M1, 1500, 1500, 20
        GENERATE 10, FN$FY
        ASSIGN 1, V$V
MET     QUEUE 1
        SEIZE CPU
        DEPART 1
        TEST LE P1, KVANT, QUANT
        ADVANCE P1
        RELEASE CPU
        TRANSFER , OUT
QUANT   ADVANCE QUANT
        ASSIGN 1-, QUANT
        RELEASE CPU
        BUFFER
        TRANSFER , MET
OUT     TABULATE TT
        TERMINATE

        GENERATE 10000
        TERMINATE 1
        START 1
```

Рисунок 8 – Программа модели для однофакторного эксперимента

5. С помощью пункта меню *Command/Show* открыть диалоговое окно *Show Command* и в строке ввода ввести `SHOW ANOVA(RES,2,1)` (Рисунок 10), где:

-  $A = 2$  – параметр, определяющий индекс матрицы результатов, по которому записываются значения отклика для каждого прогона модели;

-  $B$  – параметр, определяющий максимальное количество анализируемых взаимодействий между факторами (1, 2 или 3). Для однофакторного эксперимента  $B = 1$ .

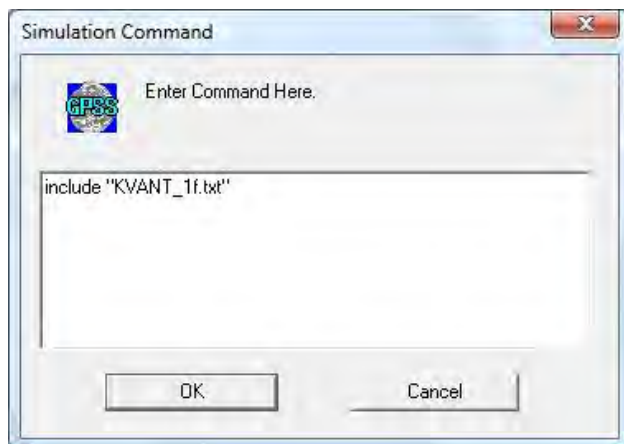


Рисунок 9 – Диалоговое окно *Simulation Command*

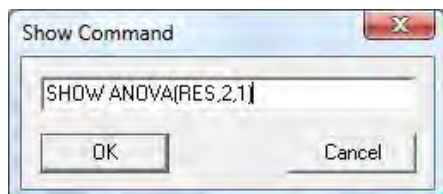


Рисунок 10 – Задание на выполнение дисперсионного анализа для однофакторного эксперимента



По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 11).

ANOVA					
Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F ( $p=,05$ )
A	164229923.209	4	41057480.802	8.002	3.30
Error	230898015.073	45	5131067.002		
Total	395127938.281	49			

Treatment Level	Count	Mean	Minimum	Maximum	95% C.I. [SE]
A					
1	10	4406.329	-3.212	9940.884	[ 2973.700, 5838.959 ]
2	10	2626.865	-7.198	8389.505	[ 1194.235, 4059.494 ]
3	10	0.000	0.000	0.000	[ -1432.629, 1432.629 ]
4	10	0.000	0.000	0.000	[ -1432.629, 1432.629 ]
5	10	0.000	0.000	0.000	[ -1432.629, 1432.629 ]

Рисунок 11 – Результаты дисперсионного анализа для однофакторного эксперимента

### 4.3.3. Библиотечная процедура ANOVA

Библиотечная процедура ANOVA системы GPSS World анализирует эксперименты от 1 до 6 факторов, включая взаимодействия 2-го и 3-го порядка между факторами.

Физический смысл параметров таблицы ANOVA:

*Source of variance* – составляющая компонента разброса значений;

*Treatment Level* – разброс значений за счет уровней факторов;

*Error* – разброс значений внутри уровня фактора;

[Перейти к оглавлению](#)

*Total* – общая ошибка;

*Error Mean Square* – среднеквадратическая ошибка;

*Sum of Squares* – сумма квадратов ошибок;

*Degrees of Freedom* – число степеней свободы;

*Mean Square* – средний квадрат;

$$\text{Mean Square} = \text{Sum of Squares} / \text{Degrees of Freedom}$$

*F* – вычисленное значение критерия;

$$F = \text{Mean Square} / \text{Error Mean Square}.$$

Нижняя часть таблицы отображает результаты расчета доверительных интервалов значений отклика.

Каждый фактор и взаимодействие в таблице ANOVA представлены отдельной строкой. Значимость фактора определяется путем сравнения вычисленного значения критерия *F* с пороговым значением. Если вычисленное значения критерия *F* больше порогового значения, эффект фактора является значимым. Согласно результатам, представленным в таблице ANOVA для однофакторного эксперимента (Рисунок 11), фактор *A* – величина кванта времени центрального процессора – является значимым, т.к. величина критерия  $F = 8.002$  больше порогового значения, равного 3.3.

#### **4.4. Проведение двухфакторного эксперимента**

Для проведения двухфакторного эксперимента необходимо выполнить следующие действия:

- создать план двухфакторного эксперимента;
- выполнить дисперсионный анализ для двухфакторного эксперимента, чтобы определить значимость факторов.

#### 4.4.1. План двухфакторного эксперимента

Определим факторный план для двухфакторного эксперимента. Для этого выполним следующую последовательность действий:

- в качестве первого фактора выберем величину кванта, обозначим первый фактор KVANT;
- выберем два уровня первого фактора: 10 и 25;
- присвоим первому фактору начальное значение 10:  
KVANT EQU 10;
- в качестве второго фактора выберем величину среднего времени между поступлениями заявок в систему, обозначим второй фактор TZA;
- выберем два уровня второго фактора: 10 и 100;
- присвоим второму фактору начальное значение 10:  
TZA EQU 10;
- в качестве отклика определим результат эксперимента, которым является СЧА М1 – время пребывания транзакта в модели;
- сформируем план факторного эксперимента в виде текстового файла *name.txt*, структура которого приведена ниже.

Перед началом моделирования необходимо определить матрицу результатов в виде

NAME MATRIX ,R,C,N;

где *NAME* – имя матрицы;

*R* – количество уровней первого фактора ( $R=2$ );

*C* – количество уровней второго фактора ( $C=2$ );

*N* – количество прогонов модели для каждого сочетания уровней факторов (эксперимента) ( $N=10$ ).

Таким образом, для сохранения результатов двухфакторного эксперимента необходимо использовать трехмерную матрицу. С помощью блока MSAVEVALUE можно обращаться только к двум первым размерностям матрицы. В этом случае все

отсутствующие размерности предполагаются равными единице. Для обращения ко всем элементам матрицы любой размерности необходимо создать процедуру на языке PLUS – низкоуровневом языке, добавленном к системе GPSS World для усиления алгоритмических возможностей языка GPSS. Матрица, определенная в операторе MATRIX, имеет глобальную область действия, т.е. доступна процедурам PLUS в любой точке программы модели.

Затем необходимо создать командный текстовый файл (назовем его *KVANT\_2f.txt*), содержащий *N* блоков команд для каждого сочетания уровней факторов (Таблица 3). Каждый блок обеспечивает получение результата для заданного сочетания уровней факторов в одном прогоне модели.

Таблица 3 – Комбинации уровней факторов для двухфакторного эксперимента 1

№ эксперимента	KVANT	TZA
1	10	10
2	10	100
3	25	10
4	25	100

План двухфакторного эксперимента представлен ниже.

```

RES          MATRIX          ,2,2,10
KVANT      EQU 10 ;значение 1-го уровня 1-го фактора
TZA        EQU 10 ;значение 1-го уровня 2-го фактора
;процедура на языке PLUS для занесения значений откликов в
;матрицу результатов
PROCEDURE ToMatr(I_,J_,K_)
BEGIN
  RES[I_,J_,K_]=M1;
END;
```

;N блоков команд для сочетания значений факторов на 1-м

:уровне

RMULT 401

START 1,NP

S\_ EQU ToMatr(1,1,1) ;оператор для обращения к процедуре PLUS

CLEAR OFF

RMULT 411

START 1,NP

S\_ EQU ToMatr(1,1,2)

CLEAR OFF

RMULT 421

START 1,NP

S\_ EQU ToMatr(1,1,3)

CLEAR OFF

RMULT 431

START 1,NP

S\_ EQU ToMatr(1,1,4)

CLEAR OFF

RMULT 441

START 1,NP

S\_ EQU ToMatr(1,1,5)

CLEAR OFF

RMULT 451

START 1,NP

S\_ EQU ToMatr(1,1,6)

CLEAR OFF

RMULT 461

START 1,NP

S\_ EQU ToMatr(1,1,7)

CLEAR OFF

RMULT 471

START 1,NP

S\_ EQU ToMatr(1,1,8)

CLEAR OFF

RMULT 481

START 1,NP

S\_ EQU ToMatr(1,1,9)

CLEAR OFF

RMULT	491	
START	1,NP	
S_ EQU	ToMatr(1,1,10)	
CLEAR	OFF	
KVANT	EQU 10	;значение 1-го уровня 1-го фактора
TZA	EQU 100	;значение 2-го уровня 2-го фактора
RMULT	401	
START	1,NP	
S_ EQU	ToMatr(1,2,1)	
CLEAR	OFF	
RMULT	411	
START	1,NP	
S_ EQU	ToMatr(1,2,2)	
CLEAR	OFF	
RMULT	421	
START	1,NP	
S_ EQU	ToMatr(1,2,3)	
CLEAR	OFF	
RMULT	431	
START	1,NP	
S_ EQU	ToMatr(1,2,4)	
CLEAR	OFF	
RMULT	441	
START	1,NP	
S_ EQU	ToMatr(1,2,5)	
CLEAR	OFF	
RMULT	451	
START	1,NP	
S_ EQU	ToMatr(1,2,6)	
CLEAR	OFF	
RMULT	461	
START	1,NP	
S_ EQU	ToMatr(1,2,7)	
CLEAR	OFF	
RMULT	471	
START	1,NP	
S_ EQU	ToMatr(1,2,8)	

CLEAR OFF  
 RMULT 481  
 START 1,NP  
 S\_ EQU ToMatr(1,2,9)  
 CLEAR OFF  
 RMULT 491  
 START 1,NP  
 S\_ EQU ToMatr(1,2,10)  
 CLEAR OFF

KVANT EQU 25 ;значение 2-го уровня 1-го фактора  
 TZA EQU 10 ;значение 1-го уровня 2-го фактора  
 RMULT 401  
 START 1,NP  
 S\_ EQU ToMatr(2,1,1)  
 CLEAR OFF  
 RMULT 411  
 START 1,NP  
 S\_ EQU ToMatr(2,1,2)  
 CLEAR OFF  
 RMULT 421  
 START 1,NP  
 S\_ EQU ToMatr(2,1,3)  
 CLEAR OFF  
 RMULT 431  
 START 1,NP  
 S\_ EQU ToMatr(2,1,4)  
 CLEAR OFF  
 RMULT 441  
 START 1,NP  
 S\_ EQU ToMatr(2,1,5)  
 CLEAR OFF  
 RMULT 451  
 START 1,NP  
 S\_ EQU ToMatr(2,1,6)  
 CLEAR OFF  
 RMULT 461  
 START 1,NP

S\_ EQU ToMatr(2,1,7)  
 CLEAR OFF  
 RMULT 471  
 START 1,NP  
 S\_ EQU ToMatr(2,1,8)  
 CLEAR OFF  
 RMULT 481  
 START 1,NP  
 S\_ EQU ToMatr(2,1,9)  
 CLEAR OFF  
 RMULT 491  
 START 1,NP  
 S\_ EQU ToMatr(2,1,10)  
 CLEAR OFF

KVANT EQU 25 ;значение 2-го уровня 1-го фактора  
 TZA EQU 100 ;значение 2-го уровня 2-го фактора  
 RMULT 401  
 START 1,NP  
 S\_ EQU ToMatr(2,2,1)  
 CLEAR OFF  
 RMULT 411  
 START 1,NP  
 S\_ EQU ToMatr(2,2,2)  
 CLEAR OFF  
 RMULT 421  
 START 1,NP  
 S\_ EQU ToMatr(2,2,3)  
 CLEAR OFF  
 RMULT 431  
 START 1,NP  
 S\_ EQU ToMatr(2,2,4)  
 CLEAR OFF  
 RMULT 441  
 START 1,NP  
 S\_ EQU ToMatr(2,2,5)  
 CLEAR OFF  
 RMULT 451



START	1,NP
S_ EQU	ToMatr(2,2,6)
CLEAR	OFF
RMULT	461
START	1,NP
S_ EQU	ToMatr(2,2,7)
CLEAR	OFF
RMULT	471
START	1,NP
S_ EQU	ToMatr(2,2,8)
CLEAR	OFF
RMULT	481
START	1,NP
S_ EQU	ToMatr(2,2,9)
CLEAR	OFF
RMULT	491
START	1,NP
S_ EQU	ToMatr(2,2,10)
CLEAR	OFF

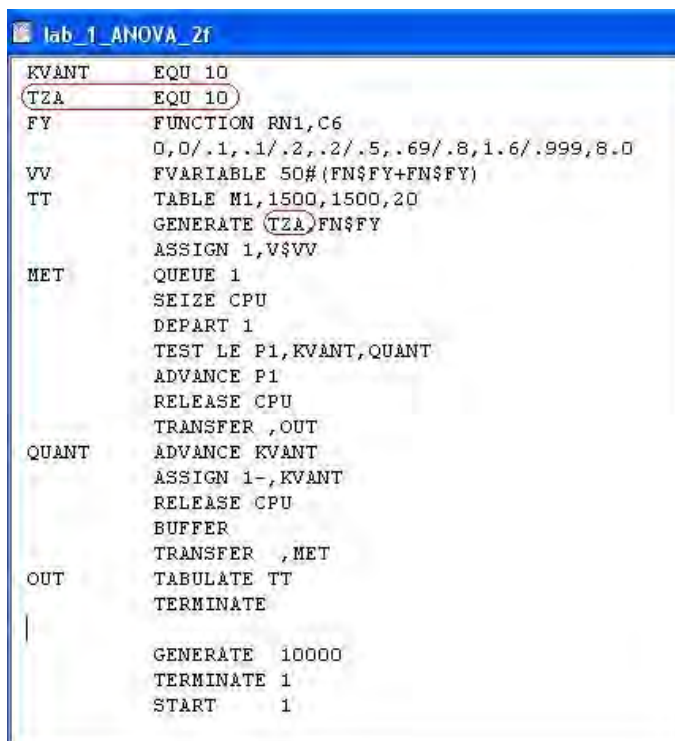
#### ***4.4.2. Последовательность проведения дисперсионного анализа для двухфакторного эксперимента***

1. Цель дисперсионного анализа – оценка влияния факторов «длительность кванта времени ЦП» и «среднее время между поступлениями заявок в систему» на значение времени обслуживания заявки. Необходимо ввести в модель факторы *KVANT* и *TZA* (Рисунок 12), задать длительность моделирования с помощью таймера, причем управляющая инструкция *START* должна присутствовать в программе модели (файл *lab1\_ANOVA\_2f.gps*).

2. Создать текстовый файл *KVANT\_2f.txt*, содержащий план двухфакторного эксперимента, в том же каталоге, что и модель *lab1\_ANOVA\_2f.gps*.

3. Транслировать модель *lab1\_ANOVA\_2f.gps* с помощью команд *Command/Create Simulation*.

Выбрать пункт меню *Command/Custom*. Откроется диалоговое окно *Simulation Command*. В поле ввода команд ввести *include "KVANT\_2f.txt"*, нажать кнопку ОК. Начинается процесс моделирования, который может занять продолжительное время в зависимости от количества прогонов модели. После его завершения в нижней строке состояния главного окна появится сообщение «*The simulation has ended*».



```
lab_1_ANOVA_2f
KVANT EQU 10
TZA EQU 10
FY FUNCTION RN1,C6
D,0/.1,.1/.2,.2/.5,.69/.8,1.6/.999,8.0
VV FVARIABLE 50#(FN$FY+FN$FY)
TT TABLE M1,1500,1500,20
GENERATE (TZA),FN$FY
ASSIGN 1,V$VV
MET QUEUE 1
SEIZE CPU
DEPART 1
TEST LE P1,KVANT,QUANT
ADVANCE P1
RELEASE CPU
TRANSFER ,OUT
QUANT ADVANCE KVANT
ASSIGN 1-,KVANT
RELEASE CPU
BUFFER
TRANSFER ,MET
OUT TABULATE TT
TERMINATE
)
GENERATE 10000
TERMINATE 1
START 1
```

Рисунок 12 – Программа модели для двухфакторного эксперимента

С помощью пункта меню *Command/Show* открыть диалоговое окно *Show Command* и в строке ввода ввести `SHOW ANOVA(RES,3,2)` (Рисунок 13), где:

-  $A = 3$  – параметр, определяющий индекс матрицы результатов, по которому записываются значения отклика для каждого прогона модели;

-  $B$  – параметр, определяющий максимальное количество анализируемых взаимодействий между факторами (1, 2 или 3). Для двухфакторного эксперимента  $B = 2$ .

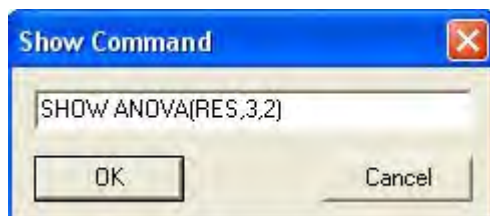


Рисунок 13 – Задание на выполнение дисперсионного анализа для двухфакторного эксперимента

По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 14).

Согласно результатам, представленным в таблице ANOVA для двухфакторного эксперимента (Таблица 4), фактор  $A$  – величина кванта времени центрального процессора – является незначимым, т.к. величина критерия  $F = 1.749$  меньше порогового значения, равного 4.11; фактор  $B$  – величина среднего времени между поступлениями заявок в систему – является значимым, т.к. величина критерия  $F = 6.922$  больше порогового значения, равного 4.11; взаимодействие между факторами  $A$  и  $B$  является незначимым, т.к. величина критерия  $F = 0.451$  меньше порогового значения, равного 4.11. Такие результаты эксперимента означают, что при

данных уровнях факторов на отклик системы оказывает влияние среднее время между поступлениями заявок в систему, т.е. интенсивность входного потока заявок.

ANOVA						
Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F [ $p=.05$ ]	
A	13901565.879	1	13901565.879	1.749	4.11	
B	55000938.998	1	55000938.998	6.922	4.11	
AB	3585829.986	1	3585829.986	0.451	4.11	
Error	286057295.230	36	7946035.979			
Total	358545630.093	39				

Treatment Level	Count	Mean	Minimum	Maximum	95% C.I. [SE]
1 1	10	4409.806	0.000	9944.976	[2626.995, 6192.616]
1 2	10	1465.760	0.000	7203.603	[-317.051, 3240.571]
2 1	10	2631.939	0.000	8390.521	[849.128, 4414.749]
2 2	10	885.529	0.000	2917.204	[-897.282, 2668.339]

2818.8714016

Рисунок 14 – Результаты дисперсионного анализа для двухфакторного эксперимента 1

Изменим уровни фактора *KVANT*. Создадим командный текстовый файл (назовем его *KVANT\_2f\_2.txt*), содержащий *N* блоков команд для каждого сочетания уровней факторов (Таблица 4).

По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 15).

Таблица 4 – Комбинации уровней факторов для двухфакторного эксперимента 2

№ эксперимента	KVANT	TZA
1	10	10
2	10	100
3	100	10
4	100	100

04/09/09 18:27:00

ANOVA

04/09/09 18:27:00

Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)
--------------------	----------------	--------------------	-------------	---	-----------------------------

A	54261840.549	1	54261840.549	10.778	4.11
---	--------------	---	--------------	--------	------

B	7459041.268	1	7459041.268	1.482	4.11
---	-------------	---	-------------	-------	------

AB	43280162.598	1	43280162.598	8.597	4.11
----	--------------	---	--------------	-------	------

04/09/09 18:27:00

Error	181243379.314	36	5034538.314		
-------	---------------	----	-------------	--	--

Total	286244423.729	39			
-------	---------------	----	--	--	--

04/09/09 18:27:00

04/09/09 18:27:00

Treatment Level	Count	Mean	Minimum	Maximum	95% C.I. (SE)
-----------------	-------	------	---------	---------	---------------

1 1	10	4409.806	0.000	9944.976	{ 2990.716, 5828.895 }
-----	----	----------	-------	----------	------------------------

1 2	10	1465.760	0.000	7203.603	{ 46.670, 2884.849 }
-----	----	----------	-------	----------	----------------------

2 1	10	0.000	0.000	0.000	{ -1419.090, 1419.090 }
-----	----	-------	-------	-------	-------------------------

2 2	10	1216.731	0.000	3428.583	{ -202.359, 2635.821 }
-----	----	----------	-------	----------	------------------------

04/09/09 18:27:00

2243.7776883

Рисунок 15 – Результаты дисперсионного анализа для двухфакторного эксперимента 2

## 4.5. Поиск наилучших решений. Оптимизирующий эксперимент

Для выбора наилучших показателей эффективности системы или выбора ее структуры, необходимо выполнить оптимальное

планирование экспериментов с применением численных методов оптимизации.

Оптимизирующий эксперимент предназначен для построения уравнения поверхности отклика для заданных факторов модели и поиска численного значения оптимума (сочетания таких значений факторов, при которых заданная функция принимает экстремальное значение – минимум или максимум). Максимально возможное количество изменяемых факторов – 5. Факторы, которые будут изменяться, должны быть представлены в программе модели в виде констант, которые определяются с помощью оператора EQU, или переменных. В ходе эксперимента производится многократный прогон модели, фиксация результатов и использование их для получения поверхности отклика и оптимума. Оптимизирующий эксперимент добавляется в созданную и отлаженную модель.

На данном этапе проведения эксперимента необходимо выполнить следующие действия:

- добавить в модель оптимизирующий эксперимент,
- получить уравнение поверхности отклика,
- проанализировать влияние факторов на результаты моделирования.

#### **4.5.1. Организация оптимизирующего эксперимента**

1. Создать программу модели для проведения оптимизирующего эксперимента (файл *Lab\_1\_OPT\_EXP.gps*) (Рисунок 16). *Примечание:* управляющую команду *START* в тексте модели задавать не следует, длительность моделирования задается с помощью процесса-таймера. Начальные значения факторов определены в начале программы с помощью операторов EQU.

```

lab_1_OPT_EXP
KVANT EQU 25
TZA EQU 100
FY FUNCTION RM1,C6
0,0/.1,.1/.2,.2/.5,.69/.8,1.6/.999,8.0
VV FVARIABLE 50#(FN$FY+FN$FY)
TT TABLE M1,1500,1500,20
GENERATE TZA, FN$FY,,100
ASSIGN 1,V$VV
MET QUEUE 1
SEIZE CPU
DEPART 1
TEST LE P1, KVANT, QUANT
ADVANCE P1
RELEASE CPU
TRANSFER ,OUT
QUANT ADVANCE KVANT
ASSIGN 1-, KVANT
RELEASE CPU
BUFFER
TRANSFER ,MET
OUT TABULATE TT
TERMINATE

GENERATE 100000
TERMINATE 1

```

Рисунок 16 – Программа модели для проведения оптимизирующего эксперимента

2. Добавить оптимизирующий эксперимент в созданную и отлаженную модель с помощью команд *Edit/Insert Experiment/optimizing* (Рисунок 17).

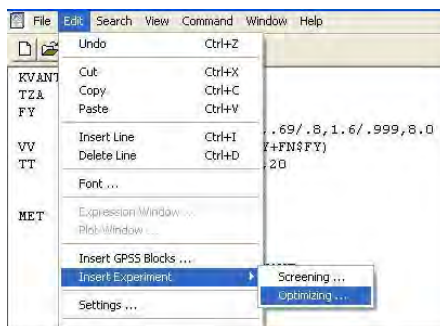


Рисунок 17 – Добавление оптимизирующего эксперимента

3. Заполнить поля диалогового окна *Optimizing Experiment Generator* (Рисунок 18):

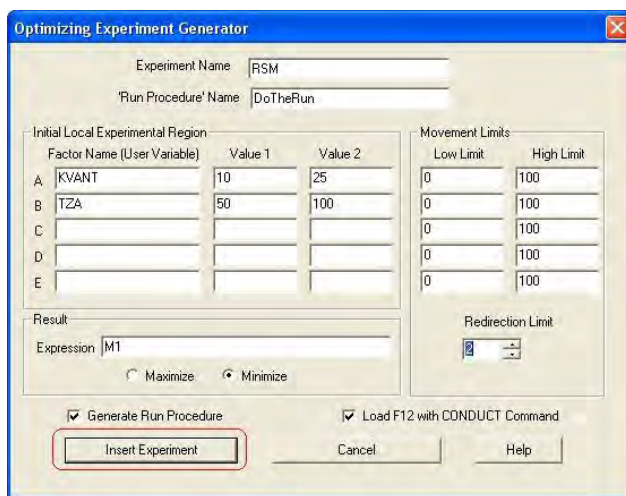


Рисунок 18 – Определение параметров оптимизирующего эксперимента

- *Experiment Name* – название эксперимента;
- *'Run Procedure' Name* – название автоматически генерируемой процедуры на языке PLUS, организующей прогоны модели, рекомендуется использовать *DoTheRun*;
- *Factor Name* – имя фактора, участвующего в эксперименте, в соответствующие поля необходимо ввести имена факторов (не более 5), заданных в модели оператором EQU, и два значения уровней факторов (Value 1 и Value 2);
- *Movement Limits* – максимальные границы перемещения при поиске оптимума (в большинстве случаев можно оставлять значения по умолчанию 0 и 100);
- *Redirection Limit* – ограничение количества изменений направлений при поиске оптимума, в некоторых случаях достаточно 1, но обычно нужно выбирать 2 или 3;



- *Expression* – целевая функция моделирования – переменная модели или выражение на языке PLUS, заключенное в круглые скобки;
- *Maximize/Minimize* – переключатель для выбора направления оптимизации;
- *Insert Experiment* – запуск автоматической генерации эксперимента.

4. При нажатии кнопки *Insert Experiment* выполняется автоматическая генерация эксперимента. При этом в окне *Run Procedure Generation* отображается автоматически сгенерированный текст процедуры прогона DoTheRun, которая выполняет установку начальных значений генераторов случайных чисел и запуск каждого прогона (Рисунок 19). При нажатии на кнопку <OK> автоматически сгенерированный код на языке PLUS будет добавлен в текст программы модели.



Рисунок 19 – Окно автоматической генерации эксперимента

5. Транслировать модель *Lab\_1\_OPT\_EXP.gps* с помощью команд *Command/Create Simulation*.
6. Запустить оптимизирующий эксперимент нажатием клавиши *F12*. В журнале сессии отобразится информация о ходе эксперимента (Рисунок 20).

В результате оптимизирующего эксперимента было получено уравнение отклика в виде полинома второго порядка

$$Y = 6.76734e+009 + 3.18048e+007 A - 1.77271e+008 B - 146791 AB + 1.3403e+007 A^2 + 1.260627e+006 B^2.$$

Локальный оптимум достигается при следующих значениях факторов: KVANT = 17.5, TZA = 75.

```

04/10/09 17:52:37 **** Experiment in Progress. ****
04/10/09 17:52:37 Simulation in Progress.
04/10/09 17:52:37 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:37 Simulation in Progress.
04/10/09 17:52:37 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:37 "Run 1. Yield=100000. KVANT=10; TZA=50;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 2. Yield=109986148.034948. KVANT=10; TZA=100;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 3. Yield=109994376.069364. KVANT=25; TZA=50;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 4. Yield=109987053.448804. KVANT=25; TZA=100;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 5. Yield=109993214.8541953. KVANT=17.5; TZA=75;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 6. Yield=109988229.4671403. KVANT=17.5; TZA=75;"
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 10000000.000000.
04/10/09 17:52:38 Simulation in Progress.
04/10/09 17:52:38 A Simulation in an Experiment has ended. Clock is 110000000.000000.
04/10/09 17:52:38 "Run 7. Yield=109997206.9697758. KVANT=17.5; TZA=75;"
04/10/09 17:52:38 Using Model:
04/10/09 17:52:38 Y = 6.76724e+009 +3.18048e+007 A -1.77271e+008 B
04/10/09 17:52:38 -146791 A B
04/10/09 17:52:38 +1.3403e+007 A^2 +1.260627e+006 B^2
04/10/09 17:52:38 Predicted optimum yield is 4.70744e+008.
04/10/09 17:52:38 Optimum is in the local Experimental Region.
04/10/09 17:52:38 RSM_FitSurfaceToData[] returns 4.
04/10/09 17:52:38 Experiment ended.

```

Рисунок 20 – Информация о ходе оптимизирующего эксперимента

#### 4.6. Выводы и рекомендации по использованию модели

На данном этапе формируются качественные и количественные оценки результатов моделирования. На основании полученных результатов должны быть сформулированы выводы по проведенным исследованиям и определены рекомендации по использованию модели.

В данной работе исследуется алгоритм распределения времени центрального процессора в системе с множественным доступом. Цель исследования – оценить характеристики дисциплины диспетчеризации центрального процессора системы с множественным доступом. В проекте применяется метод имитационного моделирования. Разработана концептуальная модель, структурная схема и алгоритм моделирования системы с множественным доступом. Выполнена программная реализация алгоритма на языке GPSS World. Проведен эксперимент с моделью, на основе которого выполнена оценка точности результатов моделирования. Сформирован план эксперимента на основе факторного и дисперсионного анализа модели. Проведен оптимизирующий эксперимент с целью выявления наилучших показателей эффективности системы.

Так, на основании имитационного моделирования с использованием GPSS-модели распределения времени центрального процессора и получения оценки распределения времени ответа и среднего времени ответа обслуживания заявки получены следующие результаты:

- величина кванта центрального процессора  $KVANT = 17.5$ ;
- среднее время между приходом заявок  $TZA = 75$ .

## ЗАКЛЮЧЕНИЕ

Компьютерное моделирование является эффективным инструментом исследования и проектирования сложных систем вследствие гибкости и адекватности реальным процессам. Особенно эффективно применение моделирования на ранних этапах проектирования автоматизированных систем. Современные вычислительные средства позволяют существенно увеличить сложность моделей при изучении систем, использовать комбинированные аналитико-имитационные модели, учитывающие все многообразие факторов в реальных системах.

Перспективным и значимым для теории и практики системного моделирования является дальнейшее развитие научных основ моделирования с ориентацией на новые информационные технологии в научных исследованиях, проектировании и управлении. Компьютерное имитационное моделирование получает все более широкое применение в широком спектре информационных технологий:

- системы на базе мультимедиа технологий;
- интеллектуальные экспертные системы, использующие имитационный подход в различных предметных областях;
- распределенные базы данных по отраслям знаний;
- средства телекоммуникации с использованием интеллектуальных сетевых технологий;
- геоинформационные системы;
- технологии защиты информации.

Освоение технологии имитационного моделирования как рабочего инструмента инженера является перспективным для решения крупномасштабных задач в широком спектре предметных областей.

## СПИСОК ЛИТЕРАТУРЫ

1. Девятков, В.В. Методология и технология имитационных исследований сложных систем: современное состояние и перспективы развития: монография / В.В. Девятков. – М.: Вузовский учебник; ИНФРА-М, 2013. – 448 с.

2. Томашевский, В.Н. Имитационное моделирование в среде GPSS / В.Н. Томашевский, Е.Г. Жданова. – М.: Бестселлер, 2003. – 219 с.

3. GPSS World Tutorial Manual // Minuteman Software: [сайт]. – 2023. – URL: <http://www.minutemansoftware.com/tutorial/t1.htm#Chapter%201> (дата обращения 10.02.2023).

4. GPSS World Reference Manual // Minuteman Software: [сайт]. – 2023. – ГКДЖ <http://www.minutemansoftware.com/reference/r4.htm> (дата обращения 10.02.2023).

5. Советов, Б.Я. Моделирование систем: учебник для академического бакалавриата / Б.Я. Советов, С.А. Яковлев. – 7-е изд. – М.: Изд-во Юрайт, 2019. – 343 с. – (Бакалавр. Академический курс). – ISBN 978-5-9916-3916-3. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/425228> (дата обращения: 10.02.2023).

6. Советов, Б.Я. Моделирование систем. Практикум: учебное пособие для бакалавров / Б.Я. Советов, С.А. Яковлев. – 4-е изд., перераб. и доп. – М.: Изд-во Юрайт, 2016. – 295 с. – (Серия: Бакалавр. Академический курс).

7. Клейнрок, Л. Вычислительные системы с очередями / Л. Клейнрок. М.: Мир, 1979. – 600 с.

8. Шрайбер, Т. Моделирование на GPSS / Т. Шрайбер. – М.: Машиностроение, 1980. – 592 с.

9. Боев, В.Д. Компьютерное моделирование: учебное пособие / В.Д. Боев, Р.П. Сыпченко. – М.: Интернет Университет Информационных Технологий, 2010. – 350 с.

10. Маликов, Р.Ф. Практикум по дискретно-событийному моделированию сложных систем в расширенном редакторе GPSS World: практикум / Р.Ф. Маликов. – Уфа: Изд-во БГПУ, 2017. – 273 с.

11. Девятков, В.В. Имитационные исследования в среде моделирования GPSS STUDIO: учебное пособие / В.В. Девятков, Т.В. Девятков, М.В. Федотов. – М.: ИНФРА-М, Вузовский учебник, 2018. – 283 с.

12. Боев, В.Д. Имитационное моделирование систем: учебное пособие для прикл. бакалавриата / В.Д. Боев. – М.: Изд-во Юрайт, 2018. – 253 с.

13. Боев, В.Д. Компьютерное моделирование: учебное пособие для СПО / В.Д. Боев, Р.П. Сыпченко. – Саратов: Профобразование, 2021. – 517 с. – ISBN 978-5-4488-0998-9. – Текст: электронный // IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/102191.html> (дата обращения 11.02.2023).

14. Ермаков, С.М. Математическая теория планирования эксперимента / С.М. Ермаков, А.А. Жиглявский. – М.: Наука, 1987. – 392 с.

### Лабораторная работа.

## Моделирование информационно-вычислительных систем в среде GPSS World

### А.1 Цель лабораторной работы

Лабораторная работа ориентирована на освоение математических (аналитических и имитационных) моделей и методов, а также современной информационной технологии на базе метода компьютерного моделирования, применяемых при проектировании и исследовании автоматизированных систем, распределенных информационных систем и сетей в широком спектре предметных областей.

Цель лабораторной работы – выработать умение применять знания и навыки в области имитационного моделирования для решения конкретных инженерных задач проектирования и исследования систем. В задачи лабораторной работы входят [4]:

- развитие навыков научно-исследовательской и проектно-конструкторской работы в области исследования и разработки сложных систем;
- постановка и проведение имитационных экспериментов с моделями процессов функционирования систем для оценки их вероятностно-временных характеристик;
- принятие экономически и технически обоснованных инженерных решений.

Ниже приведено описание последовательности этапов выполнения лабораторной работы.

## А.2 Последовательность выполнения лабораторной работы

Лабораторная работа состоит из следующих этапов:

1. Получение задания на моделирование ИВС.
2. Разработка структурно-функциональной модели.
3. Разработка концептуальной модели.
4. Разработка алгоритма и программы модели.
5. Проведение единичного эксперимента с моделью.
6. Проведение однофакторного эксперимента с моделью.
7. Проведение двухфакторного эксперимента с моделью.
8. Проведение оптимизирующего эксперимента с моделью.
9. Расчет требуемого числа реализаций модели по заданным значениям точности и достоверности результатов моделирования.

## А.3 Содержание отчета по лабораторной работе

Отчет по лабораторной работе должен содержать следующие разделы:

1. Текст задания на моделирование.
2. Структурно-функциональная модель системы.
3. Концептуальная модель системы:
  - таблица функционального соответствия,
  - описание метода моделирования случайных величин.
4. Листинг программы моделирования.
5. Результаты моделирования:
  - отчет по результатам моделирования,
  - гистограмма распределения случайной величины, которая является откликом модели.
6. Описание однофакторного эксперимента:



- код с указанием фактора, введенного в модель для исследования,
  - таблица со значениями уровней фактора в экспериментах,
  - таблица с результатами дисперсионного анализа,
  - вывод о значимости фактора.
7. Описание двухфакторного эксперимента:
- код с указанием факторов, введенных в модель для исследования,
  - таблица со значениями уровней факторов в экспериментах,
  - таблица с результатами дисперсионного анализа,
  - вывод о значимости факторов.
8. Описание оптимизирующего эксперимента:
- программа модели для оптимизирующего эксперимента,
  - задание на проведение оптимизирующего эксперимента,
  - уравнение отклика модели системы в виде полинома второго порядка, оптимальные значения факторов.
9. Расчет требуемого числа реализаций модели системы по заданным значениям точности и достоверности.
10. Выводы по работе.

#### А.4 Варианты индивидуальных заданий

##### **Вариант 1**

Система передачи данных (СПД) обеспечивает передачу пакетов данных из пункта А в пункт С через транзитный пункт В. В пункт А пакеты поступают через 10+-5мс. Здесь они буферизуются в накопителе емкостью 20 пакетов и передаются по

любой из двух линий АВ1 – за время 20 мс или АВ2 – за время 20+-5 мс. В пункте В они снова буферизуются в накопителе емкостью 25 пакетов и далее передаются по линиям ВС1 за 25+-3 мс и ВС2 за 25 мс. Причем пакеты из АВ1 поступают в ВС1, а их АВ2 – в ВС2. Чтобы не было переполнения накопителя, в пункте В вводится пороговое значение его емкости – 20 пакетов. При достижении очередью порогового значения подключается резервная аппаратура и время передачи снижается для линий ВС1 и ВС2 до 15 мс.

Смоделируйте прохождение через СПД 500 пакетов. Определите вероятность подключения резервной аппаратуры и характеристики очереди пакетов в пункте В.

### **Вариант 2**

Система обработки информации содержит мультиплексный канал и три рабочих станции. Сигналы от датчиков поступают на вход канала через интервалы времени 10+-5 мкс. В канале они буферизуются и предварительно обрабатываются в течение 10+-3 мкс. Затем они поступают на обработку в ту рабочую станцию, где имеется наименьшая по длине входная очередь. Емкости входных накопителей во всех рабочих станциях рассчитаны на хранение величин 10 сигналов. Время обработки сигнала в любой рабочей станции равно 33 мкс.

Смоделируйте процесс обработки 1000 сигналов, поступающих с датчиков. Определите средние времена задержки сигналов в канале и рабочей станции, а также вероятности переполнения входных накопителей.

### **Вариант 3**

Магистраль передачи данных состоит из двух каналов (основного и резервного) и общего накопителя. При нормальной

работе сообщения передаются по основному каналу за  $7+3$  с. В основном канале происходят сбои через интервалы времени  $200+35$ с. Если сбой происходит во время передачи, то за 2 с запускается запасной канал, который передает прерванное сообщение с самого начала. Восстановление основного канала занимает  $23+7$  с. После восстановления резервный канал выключается и основной канал продолжает работу с очередного сообщения. Сообщения поступают через  $9+4$  с и остаются в накопителе до окончания передачи. В случае сбоя передаваемое сообщение передается повторно по запасному каналу.

Смоделируйте работу магистрали передачи данных в течение 1 ч. Определите загрузку запасного канала, частоту отказа канала и число прерванных сообщений.

#### **Вариант 4**

В системе передачи данных осуществляется обмен пакетами данных между пунктами А и В по дуплексному каналу связи. Пакеты поступают в пункт А системы от абонентов с интервалами времени между ними  $8+2$  мс, в пункт В – с интервалами  $12+2$  мс. Передача пакета занимает 10 мс. В пунктах имеются буферные регистры, которые могут хранить два пакета (включая передаваемый). В случае прихода пакета в момент занятости регистров пунктам системы предоставляется выход на спутниковую полудуплексную линию связи, которая осуществляет передачу пакетов данных за  $10+5$  мс. При занятости спутниковой линии пакет получает отказ.

Смоделируйте обмен информацией в системе передачи данных в течение 1 мин. Определите частоту вызовов спутниковой линии и ее загрузку.

### **Вариант 5**

Специализированная вычислительная система состоит из трех процессоров и общей оперативной памяти ограниченного объема. Задания, поступающие на обработку через интервалы времени  $5+2$  мин, занимают объем оперативной памяти размером в страницу. После обработки первым процессором в течение  $5+1$  мин их объем увеличивается до двух страниц, и они поступают в оперативную память. Затем после обработки во втором процессоре, которое занимает  $2.5+0.5$  мин на страницу, объем возрастает до трех страниц. Далее задания через оперативную память поступают в третий процессор на решение, требующее  $1.5+0.4$  мин на страницу, и покидают систему, минуя оперативную память.

Смоделируйте работу вычислительной системы в течение 50 ч. Определите характеристики занятости оперативной памяти.

### **Вариант 6**

В системе клиент-сервер на обработку принимаются клиентские задания трех классов А, В и С. Исходя из наличия ресурсов сервера, задания классов А и В могут решаться одновременно, а задания класса С монополизируют сервер. Задания класса А поступают через  $20+5$  мин, класса В – через  $20+10$  мин, класса С – через  $30+10$  мин и требуют для выполнения: класс А –  $20+5$  мин, класс В –  $21+3$  мин и класс С –  $28+5$  мин. Задачи класса С поступают на сервер, если он полностью свободен. Задачи классов А и В могут дозагружаться к решаемой задаче.

Смоделируйте работу системы клиент-сервер в течение 80 ч. Определите загрузку сервера.

### **Вариант 7**

В системе клиент-сервер имеется два клиентских рабочих места и один сервер печати. Клиенты приходят с интервалом  $8 \pm 2$  мин и каждый третий из них выполняет задания, требующие использования сервера печати. Допустимая очередь к серверу составляет 4 задания. Работа сервера печати занимает  $8 \pm 1$  мин, а выполнение задания – 17 мин. Кроме того, 20% клиентов возвращается для повторного выполнения задания с использованием сервера печати.

Смоделируйте работу системы в течение 60 ч. Определите загрузку клиентских рабочих мест, сервера печати и вероятность отказа в обслуживании вследствие переполнения очереди.

### **Вариант 8**

Система передачи данных обеспечивает передачу пакетов данных из пункта А в пункт С через транзитный пункт В. В пункт А пакеты поступают через  $10 \pm 5$  мс. Здесь они сохраняются в накопителе с максимальной вместимостью 25 пакетов и с равной вероятностью передаются по одной из двух линий: АВ1 – за 20 мс; АВ2 – за  $20 \pm 5$  мс. В пункте В пакеты снова буферизуются в накопителе с максимальной вместимостью 20 пакетов и дальше передаются по линии ВС1 за  $20 \pm 3$  мс и по линии ВС2 – за 25 мс. Причем, пакеты, которые передавались по АВ1, поступают в ВС1, а те, которые передавались по АВ2 – в ВС2. При достижении предельного значения количества пакетов в накопителе пакет, который пытается попасть в этот накопитель, уничтожается.

Смоделируйте работу системы на протяжении 1 мин. Определить вероятность уничтожения пакета.

### **Вариант 9**

В системе передачи цифровой информации передается речь в цифровом виде. Речевые пакеты передаются через два транзитных канала, перед каждым каналом имеются буферы. Время передачи пакета по каналу имеет экспоненциальное распределение со средним значением 5 мс. Пакеты поступают через  $6+3$  мс. Пакеты, передававшиеся более 10 мс, на выходе системы уничтожаются, так как их появление в декодере значительно снизит качество передаваемой речи. Уничтожение более 30% пакетов недопустимо. При достижении такого уровня система за счет ресурсов ускоряет передачу до 4 мс на канал. При снижении уровня до приемлемого значения происходит отключение ресурсов и возврат к прежней скорости передачи.

Смоделируйте 10 с работы системы. Определите частоту уничтожения пакетов, частоту подключения ресурса и среднее время нахождения одного пакета в системе (с учетом времени ожидания).

### **Вариант 10**

Локальная сеть обслуживает три рабочих станции по кольцевому циклическому алгоритму с передачей маркера, предоставляя каждой станции 30 с. Если в течение этого времени задание обрабатывается, то обслуживание завершается. В противном случае, остаток задания ставится в специальную очередь, которая использует свободные циклы станций, т.е. задача обслуживается, если на какой-либо станции нет заданий. Задания на рабочую станцию поступают через  $30+5$  с и имеют длину  $300+50$  команд. Скорость обработки заданий равна 10 команд/с.

Смоделируйте 5 ч работы локальной сети. Определите загрузку локальной сети, параметры очереди неоконченных заданий.

Определите величину цикла, при котором все задания будут обслужены без специальной очереди.

### **Вариант 11**

В системе клиент-сервер поступающий запрос обрабатывается на клиентском рабочем месте и с вероятностью 50% требуется дальнейшая обработка запроса на сервере. Запросы поступают через  $20+3$  с, первичная обработка запроса занимает 2 с, выдача ответа требует  $18+2$  с, передача по каналу связи занимает 3с. Временные характеристики обработки на сервере аналогичны.

Смоделируйте прохождение 1000 запросов. Определите необходимую емкость буферов на клиентском рабочем месте и на сервере, обеспечивающие безотказную работу системы, а также характеристики времени обслуживания запроса.

### **Вариант 12**

Система клиент-сервер состоит из трех клиентских рабочих мест и сервера. Каждый клиент формирует задание, состоящее из  $8+2$  запросов. Ввод одного запроса занимает  $10+-5$  с. Когда задание полностью сформировано, оно поступает для обработки на сервер. Получение ответа на каждый запрос требует 3 с работы сервера и 5 с работы компьютера клиента. Вывод результата задания требует 8 с работы компьютера клиента. Анализ результата занимает у клиента 30 с, после чего цикл повторяется.

Смоделируйте работу системы в течение 6 ч. Определите вероятность простоя проектировщика из-за занятости сервера и коэффициент загрузки сервера.

### **Вариант 13**

Имеется специализированный вычислительный комплекс, который используется с интервалом  $10+-5$  мин. Время,

необходимое для выполнения расчетов, характеризуется интервалом 15+5 мин. Третья часть пользователей после окончания расчетов распечатывает результаты работы в течение 3+2 мин, причем работа принтера не влияет на расчеты. Не допускается, чтобы более пяти пользователей ожидали своей очереди произвести расчеты.

Смоделируйте процесс обслуживания 100 пользователей. Определите вероятность отказа в использовании специализированного вычислительного комплекса, среднее число пользователей в очереди, а также коэффициенты загрузки специализированного вычислительного комплекса и принтера.

#### **Вариант 14**

Распределенная АИС состоит из трех рабочих станций. С интервалом 3+1 мин в систему поступают задания, которые с вероятностями  $P_1=0,4$ ,  $P_2=P_3=0,3$  адресуются одной из трех станций. В буфере каждой станции имеется очередь заданий неограниченной длины. После обработки задания на первой станции оно с вероятностью  $P_{12}=0,3$  поступает в очередь ко второй станции и с вероятностью  $P_{13}=0,7$  – в очередь к третьей станции. После обработки на второй или третьей станции задание считается выполненным. Продолжительность обработки заданий на разных станциях характеризуется интервалами  $T_1=7+4$  мин,  $T_2=3+1$  мин,  $T_3=5+2$  мин.

Смоделируйте процесс обработки 500 заданий. Определите среднее время обработки заданий в АИС, коэффициенты загрузки рабочих станций, средние длины очередей в буферах станций.

#### **Вариант 15**

Система автоматизации проектирования состоит из ЭВМ и трех подключенных к ней терминалов. За каждым терминалом



работает один проектировщик, который формирует задания на расчет в интерактивном режиме. Набор строки задания занимает  $10 \pm 5$  с. Анализ строки требует 3 с работы ЭВМ и 5 с работы терминала. В каждый момент времени может анализироваться только одна строка. После набора десяти строк считается, что задание сформировано и поступает на решение, которое занимает  $10 \pm 3$  с работы ЭВМ (решение заданий имеет больший приоритет, чем анализ строк). Вывод результата решения требует 8 с работы терминала, а анализ результата проектировщика требует  $30 \pm 10$  с, после чего цикл повторяется.

Промоделировать работу системы на протяжении 6 часов. Определить вероятность простоя проектировщика из-за занятости ЭВМ, коэффициент загрузки ЭВМ и параметры очереди к ЭВМ.

### **Вариант 16**

Информационно-поисковая библиографическая система построена на базе двух компьютеров и имеет один терминал для ввода и вывода информации. Первый компьютер содержит базу данных по научно-техническим проблемам (вероятность обращения к ней – 0,7), а вторая – по медицинским (вероятность обращения к ней – 0,3). Пользователи обращаются к услугам системы каждые  $5 \pm 2$  мин. Если в очереди к терминалу ожидают 10 пользователей, то вновь прибывшие пользователи получают отказ в обслуживании. Поиск информации в первой базе данных продолжается  $6 \pm 4$  мин, а во второй –  $3 \pm 2$  мин. Для передачи текста запроса к нужной базе пользователи тратят  $2 \pm 1$  мин. Вывод результатов поиска происходит за 1 мин.

Смоделируйте процесс работы системы за 8 часов. Определите среднюю длину очереди к терминалу, а также коэффициенты загрузки технических средств системы.

### **Вариант 17**

В специализированной вычислительной системе периодически выполняются три вида заданий, которые характеризуются уровнями приоритета: нулевым, первым и вторым. Каждый новый запуск задания оператор производит с терминала, работая на нем 50+-30 с. После запуска задания оно требует для своего выполнения 100+-50 с времени работы процессора, причем задания более высокого приоритета прерывают выполнения задач более низкого приоритета. Результаты обработки задания выводятся на печать без прерываний в течение 30+-10 с, после чего производится их анализ в течение 60+-20 с, и задание запускается снова. При работе терминала и при выводе результатов на печать процессор не используется.

Смоделируйте процесс работы систем при условии, что задание второго уровня приоритета выполняется 100 раз. Подсчитайте число циклов выполнения остальных заданий и определите коэффициенты загрузки технических средств системы.

### **Вариант 18**

Задания на обработку данных характеризуются требуемым временем работы процессора и условно подразделяются на короткие и длинные. Короткие задания поступают через каждые 8+-3 мин и требуют для своей обработки 4+-3 мин времени работы процессора. Короткие задания вводятся за 2+-1 мин. Короткие задания имеют абсолютный приоритет над длинными заданиями при использовании процессора. Длинные задания поступают через каждые 8+-5 мин и требуют для своей обработки 8+-2 мин времени работы процессора. Длинные задания вводятся за 5+-2 мин. После обработки процессором как коротких, так и длинных заданий производится вывод результатов на печать в течение 3+-1 мин.

Смоделируйте процесс функционирования системы при условии, что обработать необходимо 200 заданий. Определите число коротких и длинных заданий за время обработки, а также коэффициент загрузки процессора.

### **Вариант 19**

Распределенная АИС содержит три компьютера. Задания на обработку поступают с интервалом  $20 \pm 5$  мин. На подготовку запроса требуется  $12 \pm 3$  мин, после чего каждое задание поступает на один из свободных компьютеров. Примерно в 70% заданий в результате их первичной обработки обнаруживаются ошибки, которые сразу же в течение  $3 \pm 2$  мин исправляются пользователями на соответствующем компьютере. Возможность ошибки при повторной обработке исключается, т.е. повторная обработка всегда является окончательной. Обработка задания в каждом случае занимает  $10 \pm 5$  мин.

Смоделируйте процесс функционирования АИС при условии, что необходимо обработать 500 заданий. Определите среднее время ожидания в очереди на обработку, а также коэффициенты загрузки компьютеров.

### **Вариант 20**

АИС реального времени состоит из центрального процессора (ЦП), основной памяти (ОП), буфера емкостью 10000 байтов и накопителя на магнитных дисках (МД). Запросы от удаленных терминалов поступают каждые  $75 \pm 25$  мс и обрабатываются ЦП в течение 1 мс. После этого каждый запрос помещается в буфер либо получает отказ в обслуживании, если буфер заполнен (каждый запрос занимает 200 байтов). Для обслуживаемых запросов производится поиск информации на МД за время  $120 \pm 25$  мс и ее считывание за время  $10 \pm 5$  мс. Работа с МД не

требует ресурсов ЦП. Для подготовки ответа необходима работа ЦП в течение 5 мс. После этого запрос считается обслуженным и освобождает место в буфере.

Смоделируйте процесс обслуживания 500 запросов. Подсчитайте число запросов, получивших отказ в обслуживании. Определите среднюю загрузку буфера, а также коэффициенты загрузки ЦП и МД.

### **Вариант 21**

Для ускорения обработки коротких заданий выбран пакетный режим работы с квантованием времени процессора. Всем заданиям пакета по очереди предоставляется процессор на одинаковое время 10 с (круговой циклический алгоритм разделения времени). Если в течение этого времени заканчивается выполнение задания, оно покидает систему и освобождает процессор. Если же очередного кванта времени не хватает для завершения задания, оно помещается в конец очереди-пакета. Последнее задание пакета выполняется без квантования. Пакет считается готовым к обработке. Если в нем содержится 5 заданий. Новый пакет вводится после окончания обработки предыдущего. Задания поступают в систему с интервалом времени 60+-30 с и требуют обработки в течение 50+-45 с.

Смоделируйте процесс обработки 500 запросов. Определите максимальную длину очереди готовых к обработке пакетов и коэффициент загрузки процессора.

### **Вариант 22**

Система автоматизации проектирования (САПР) создана на базе АИС, функционирующей в режиме множественного доступа. Пятеро инженеров-проектировщиков с помощью своих терминалов одновременно и независимо проводят диалог, определяя очередной

вариант расчета. Каждый диалог состоит из 10 циклов ввода-вывода данных. Во время одного цикла происходит следующее: за 10+-5 с инженер обдумывает и вводит текст строки; в течение 2 с работает процессор, подготавливая текст ответа; в течение 5 с текст ответа выводится на дисплей. После ввода 11-й строки начинается работа процессора по расчету конструкции и продолжается 30+-10 с. За 5 с результат расчета выводится на дисплей, после чего инженер в течение 15+-5 с анализирует его и начинает новый диалог. Операции по подготовке текста ответа имеют абсолютный приоритет над расчетными операциями, т.е. прерывают выполнение последних.

Смоделируйте процесс работы САПР при условии, что расчет вариантов конструкции повторяется 500 раз. Определите среднее время выполнения диалога и расчетных операций, а также коэффициент загрузки процессора.

### **Вариант 23**

Распределенный банк данных организован на базе трех удаленных дуг от друга вычислительных центров А, В и С. Все центры связаны между собой каналами передачи информации, работающими в дуплексном режиме независимо друг от друга. В каждый из центров с интервалом времени 50+-20 мин поступают заявки на проведение информационного поиска.

Если центр, получивший заявку от пользователя, свободен, в течение 2+-1 мин производится предварительная обработка заявки, в результате которой формируются запросы для остальных центров. В центре, получившем заявку от пользователя, начинается поиск информации по запросу, а на другие центры по соответствующим каналам передаются за 1 мин тексты запросов, после чего там также может начаться поиск информации, который продолжается:

в центре А – 5+-2 мин, в центре В – 10+-2 мин, в центре С – 15+-2 мин. Тексты ответов передаются за 2 мин по соответствующим каналам в центр, получивший заявку на поиск. Заявка считается выполненной, если получены ответы от всех трех центров. Каналы при своей работе не используют ресурсы процессоров центров.

Смоделируйте процесс функционирования распределенного банка данных при условии, что обслуживается 500 заявок. Подсчитайте число заявок, поступивших и обслуженных в каждом центре. Определите коэффициенты загрузки процессоров центров.

### **Вариант 24**

В системе автоматизации экспериментов (САЭ) данные от измерительных устройств поступают в буферную зону оперативной памяти каждые 800+-400 мс. Объем буфера 256 байт, длина одного информационного сообщения – 2 байта. Для записи сообщения в буфер требуется 20 мс времени работы процессора. После заполнения буфера его содержимое переписывается на магнитный диск (МД), для чего сначала необходима работа процессора в течение 30 мс, а потом – совместная работа процессора и накопителя на МД в течение 100+-30 мс. Для обработки каждой новой порции информации на МД, объем которой 2560 байт, запускается специальная программа, требующая 100+-20 мс времени работы процессора. Эта программа имеет самый низкий приоритет и прерывается программами сбора и переписи данных на МД.

Смоделируйте процесс сбора и обработка данных с САЭ при условии, что обработать необходимо 100 порций информации. Определите среднюю длительность выполнения программы обработки. Определите, сколько раз ее выполнения было прервано.

### **Вариант 25**

В узел коммутации сообщений, который состоит из входного буфера, процессора, двух выходных буферов и двух выходных линий, поступают сообщения с двух направлений (по каждому через интервалы времени  $15 \pm 7$  мс). Сообщения с первого направления поступают во входной буфер, обрабатываются в процессоре, накапливаются в выходном буфере первой линии и передаются по первой выходной линии. Сообщения со второго направления обрабатываются аналогично, но накапливаются в выходном буфере второй линии и передаются по второй линии. Примененный метод контроля потоков разрешает одновременное присутствие в системе не больше трех сообщений с каждого направления. Если при наличии в системе трех сообщений с некоторого направления поступает сообщение с этого же направления, то оно получает отказ. Время обработки в процессоре равняется 7 мс на сообщение, время передачи по каждой из выходных линий составляет  $15 \pm 5$  мс.

Смоделируйте работу узла коммутации на протяжении 10 с. Определите загрузку устройств и вероятность отказа в обслуживании.

### **Вариант 26**

Мультиплексор подключен к высокоскоростному каналу (ВК) связи и работает в режиме разделения времени с четырьмя низкоскоростными каналами (НК), опрашивая их циклически. На один опрос каждого из НК мультиплексор тратит 50 мс, время переключения между НК – 10 мс. Если в опрашиваемом НК есть сообщение, оно передается по ВК в течение  $500 \pm 60$  мс. За один опрос передается одно сообщение. Время возникновения сообщения в четырех НК равномерно распределено в интервале

1500 +- 500 мс, и сообщение могут равновероятно появляться на любом канале.

Определите загрузку ВК при передаче по нему 1000 сообщений.

### **Вариант 27**

Подключенные к полудуплексному каналу связи АП, передав сообщение, ожидают ответа от ЭВМ. После передачи входного сообщения от АП канал связи освобождается: другое сообщение может занять канал, не ожидая выдачи ответа на предыдущее; однако, выходным сообщениям назначается более высокий приоритет по отношению к входным. Поток сообщений, поступающий от всех АП в канал, простейший с интенсивностью 200 сообщений в час. Закон распределения вероятностей длины входных сообщений, поступающих от АП, равномерный в диапазоне [10,20] символов. Время обработки одного сообщения в ЭВМ распределено экспоненциально. Интенсивность обработки – 1 сообщение в 5 с. Закон распределения вероятностей длины выходного сообщения (ответа от ЭВМ) – дискретный

$$\begin{pmatrix} 11 & 12 & 13 & 14 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}.$$

Скорость передачи информации в канале связи – 20 символов в секунду. Время подготовки канала  $T = 0.5$  с для входного и выходного сообщений.

Постройте модель канала. Определите продолжительность обработки сообщения при передаче 200 сообщений.

### **Вариант 28**

Информация с трех датчиков передается по каналу связи в ЭВМ. Поток сообщений с первого датчика – простейший с интенсивностью 0.3, со второго – регулярный с интервалом времени 2 ед., с третьего датчика – Эрланга второго порядка с



интенсивностью 0.2. Программа управления каналом связи опрашивает датчики в соответствии со списком опроса. Время, затрачиваемое на опрос одного датчика – 1 ед. Интервалы времени между опросами – 1 ед. Время передачи сообщения с датчика в ЭВМ – 2 ед. Время обработки сообщения в ЭВМ распределено по усеченно-нормальному закону с математическим ожиданием 5 и дисперсией 3.

Постройте модель такой системы. Определите продолжительность обработки сообщения при передаче 200 сообщений.

### **Вариант 29**

Пять операторов работают в справочной телефонной станции, сообщая номера телефонов по запросам абонентов, которые обращаются по номеру 09. Автоматический коммутатор переключает абонента на того оператора, в очереди которого ожидает наименьшее число абонентов, причем наибольшая допустимая длина очереди перед оператором – два абонента. Если все очереди имеют максимальную длину, вновь поступивший вызов получает отказ. Обслуживание абонента оператором длится 30+-20 с. Вызовы поступают в справочную через каждые 5+-3 с.

Смоделируйте обслуживание 500 вызовов. Подсчитайте количество отказов. Определите коэффициенты загрузки операторов справочной.

### **Вариант 30**

К серверу базы данных подключены четыре клиентских рабочих места, на которых выполняются операции четырех типов, требующие использования ресурсов сервера. Причем, если хоть на одном рабочем месте выполняется операция 1, для остальных рабочих мест сервер заблокирован. Если два рабочих места

выполняют операцию 2, то для оставшихся двух рабочих мест сервер заблокирован. Если три клиента выполняют операцию 3, оставшийся клиент не получает доступ к серверу. При выполнении операции 4 все четыре клиента получают доступ к серверу. Интенсивности поступления операций различных типов равны. Операции от одного клиента поступают через экспоненциально распределенные интервалы времени со средним значением 160 с. Выполнение любой операции длится 10 с.

Смоделируйте работу сервера базы данных в течение 4 ч. Определите загрузку сервера, вероятности простоя клиентов и частоту одновременного выполнения операций вида 2 и вида 3.

### **Вариант 31**

Компьютер задействован в управлении технологическим оборудованием. Для контроля состояния оборудования каждые 20 мин запускается задача одного из трех типов. Через каждые 5 мин работы процессора каждая задача выводит результаты работы в базу данных (БД). При обращении двух и более задач к БД образуется очередь, которая обслуживается по правилу FIFO.

Общий объем памяти компьютера 1024 Кбайт. В первоначальный момент запуска компьютера загружается ОС, ядро которой постоянно находится в памяти и занимает 200 Кбайт. Компьютер работает в мультипрограммном режиме и во время выполнения операций вывода в БД процессор может выполнять другую задачу, если она загружена в память. После последнего вывода в БД задача выгружается из памяти и завершает свою работу.

Периодически с интенсивностью  $0.005 \text{ мин}^{-1}$  и экспоненциальным распределением возникает аварийный режим оборудования, при котором немедленно запускается на выполнение задача четвертого типа, выводящая оборудование из аварийного

режима. Она прерывает работу всех других задач. Прерванная задача выгружается из памяти без вывода результатов в БД. По окончании выполнения задачи четвертого типа, она имеет преимущество для вывода в БД перед другими задачами. Вытесненные задачи загружаются с диска в память и продолжают работу. Данные для моделирования приведены в Таблица 5.

Таблица 5 – Данные для варианта 31

Тип задачи	1	2	3	4
Вероятность возникновения	0.5	0.35	0.15	-
Объем памяти, Кбайт	200	300	400	500
Время обработки ЦП, мин	15	20	25	5
Время вывода в БД, мин	3	5	7	2

Смоделируйте работу компьютера в течение 5 суток и оцените размер очереди к памяти, ее загрузку и загрузку процессора.

### **Вариант 32**

Распределенная обработка данных в АИС ведется вычислительной сетью, состоящей из трех последовательно соединенных рабочих станций, связанных идеальными каналами передачи данных. Первичный поток запросов на решение задач обработки данных – простейший. Запросы могут иметь относительный приоритет, позволяющий занять место в очереди перед низкоприоритетными заявками. Время обработки заявки каждой рабочей станцией подчинено усеченно-нормальному закону распределения. Запрос считается выполненным при завершении обработки данных последней рабочей станцией. Очереди в сети могут быть неограниченной длины, но существует ограничение на время пребывания запроса в каждой из очередей в

зависимости от приоритета. Чем выше приоритет, тем меньше допустимое время пребывания в очереди. Запрос, время обслуживания которого не удовлетворяет этому ограничению, попадает в очередь с более низким приоритетом. Запрос выбирается на обслуживание из низкоприоритетной очереди, если очереди с более высоким приоритетом свободны.

Исследуйте длины очередей, распределение времени выполнения заявок в зависимости от приоритета, среднее время пребывания в очередях, коэффициент загрузки каждой рабочей станции.

### **Вариант 33**

АИС использует иерархическую базу данных, структура которой представляется деревом. Поток запросов к базе – простейший. Обработка каждого запроса связана с прохождением пути, начинающегося в корневом узле и завершающегося в одном из терминальных узлов. Каждый узел дерева связан с физическим блоком данных определенного размера, который адресуется номером дорожки на магнитном диске. Время поиска блока данных зависит от времени позиционирования магнитной головки на нужную дорожку. Прохождение узла возможно только тогда, когда он доступен и соответствующий ему блок данных с диска прочитан и размещен в буфере оперативной памяти. Время считывания блока данных пропорционально его длине, имеющей равномерное распределение в определенном диапазоне. Если при этом выполняется обновление данных, узел становится недоступным для других запросов. Доступ возобновляется только после перезаписи данных на диск.

Определите среднее время обработки запросов, среднее время ожидания доступа к узлам, среднее время доступа к блокам данных на диске, коэффициент загрузки диска.

### **Вариант 34**

Распределенная обработка данных в АИС осуществляется вычислительной сетью звездообразной структуры. К серверу подключено несколько рабочих станций, а к каждой из таких станций – группа терминалов. Поток запросов от терминалов – простейший. Для поступившего запроса определен приоритет. Обработка на сервере требуется с определенной вероятностью, в остальных случаях достаточно обработки на рабочей станции, с которой связан терминал. Если требуется обработка на сервере, то производится передача запроса и результатов обработки по каналам передачи данных. Каналы – полудуплексные с заданной скоростью передачи информации. Образующиеся очереди запросов на передачу и обработку упорядочиваются по приоритетам и имеют неограниченную длину. Закон распределения вероятностей длин запросов и результирующих сообщений – Пуассона с заданной средней длиной сообщения. Время обработки запросов распределено по усеченно-нормальному закону.

Определите распределение времени обработки запросов в зависимости от приоритетов, средние длины очередей, коэффициенты загрузки рабочих станций, сервера и каналов передачи данных.

### **Вариант 35**

Подсистема передачи данных в АИС представляет собой сеть с коммутацией каналов. Запросы, подлежащие передаче, поступают на станции сети в виде простейших потоков. Каждый запрос характеризуется длиной, распределенной по усеченно-нормальному закону и номером станции-получателя. Тракт передачи между станцией-отправителем и станцией-получателем состоит из совокупности станций и каналов передачи между ними. Время передачи запроса пропорционально его длине. На время

передачи запроса станции закрепляются за определенным трактом и недоступны для остальных трактов. Запрос должен быть полностью передан до момента разрыва тракта. Очереди в станциях имеют неограниченную длину. Время обработки запроса на станции-получателе распределено равномерно в заданном диапазоне. Ответ направляется к станции-отправителю аналогичным образом с аналогичными характеристиками процесса передачи.

Определите средние длины очередей запросов в станциях, среднее время ожидания передачи, среднее время реакции АИС, коэффициенты загрузки каналов и станций.

### **Вариант 36**

Подсистема передачи данных в АИС представляет собой сеть пакетной коммутации. Запросы, подлежащие передаче, поступают на станции сети в виде простейших потоков. Каждый запрос характеризуется длиной, распределенной по усеченно-нормальному закону и номером станции-получателя. Время передачи запроса пропорционально его длине. По мере продвижения к станции-получателю запросы попадают в очереди в промежуточных станциях. Очереди в станциях имеют неограниченную длину, но время пребывания запроса в сети ограничено. Запрос, превысивший это время, считается необслуженным и удаляется из сети. Время обработки в станции-получателе распределено по равномерному закону. Пакет-ответ направляется к станции-отправителю аналогичным образом с аналогичными характеристиками процесса передачи.

Определите средние длины очередей запросов в станциях, среднее время ожидания передачи, среднее время реакции АИС, коэффициенты загрузки каналов и станций, вероятность того, что запрос останется необслуженным.

### **Вариант 37**

Распределенная обработка данных в АИС выполняется вычислительной сетью, состоящей из трех связанных между собой рабочих станций. Для связи станций используются полудуплексные каналы передачи данных, время передачи по которым пропорционально передаваемому объему данных. В каждую станцию поступает входной простейший поток запросов. С определенной вероятностью может потребоваться обработка на любой из трех станций. Закон распределения вероятностей длин запросов и ответов – пуассоновский с определенной средней длиной запроса и ответа. Время обработки запроса каждой из станций подчинено усеченно-нормальному закону. Очереди в станциях имеют неограниченную длину.

Определите средние длины очередей запросов в станциях, среднее время ожидания передачи, среднее время реакции АИС, коэффициенты загрузки каналов и станций.

### **Вариант 38**

Подсистема передачи данных в АИС представляет собой сеть с коммутацией виртуальных каналов. Запросы, подлежащие передаче, поступают на станции сети в виде простейших потоков. Каждый запрос характеризуется длиной, распределенной по усеченно-нормальному закону и номером станции-получателя. Тракт передачи между станцией-отправителем и станцией-получателем состоит из совокупности станций и каналов передачи между ними. Каждый канал и каждая станция могут одновременно участвовать в нескольких трактах. Время передачи запроса пропорционально его длине. Запрос должен быть полностью передан до момента разрыва тракта. Очереди в станциях имеют неограниченную длину. Время обработки запроса на станции-

получателе распределено равномерно в заданном диапазоне. Ответ направляется к станции-отправителю аналогичным образом с аналогичными характеристиками процесса передачи.

Определите средние длины очередей запросов в станциях, среднее время ожидания передачи, среднее время реакции АИС, коэффициенты загрузки каналов и станций.

### **Вариант 39**

Распределенная сетевая АИС функционирует в режиме клиент-сервер и содержит три терминала, файл-сервер и принт-сервер. Входные потоки заявок от каждого из трех терминалов – простейшие. Время первичной обработки распределено по усеченно-нормальному закону. Дальнейшая обработка на файл-сервере и затем на принт-сервере требуется с определенной вероятностью. Время обработки на серверах распределено равномерно. Очереди на серверах имеют ограниченную длину. Запрос, которому не хватает места в очереди, считается необслуженным.

Определите среднее время обработки запросов в АИС, коэффициенты загрузки терминалов и серверов, длины очередей, вероятность того, что запрос окажется необслуженным.

### **А.5 Контрольные вопросы**

1. Охарактеризуйте моделирование как основной метод исследования сложных систем.

2. Приведите классификацию моделей по степени абстрагирования модели от оригинала.

3. В чем заключается различие между аналитическими и имитационными моделями? В чем их достоинства и недостатки?

4. Опишите формальную модель объекта исследования.

5. Определите понятие состояния моделируемой системы.



6. Определите структуру имитационной модели.
7. Какие области применения имитационного моделирования вам известны? В чем заключаются отличия имитационного эксперимента от натурального? Определите преимущества и недостатки имитационного моделирования.
8. Выполните постановку задач анализа и синтеза, решаемых с помощью имитационного моделирования.
9. Определите однокритериальную и многокритериальную оценки показателя эффективности моделируемой системы.
10. Перечислите основные этапы моделирования систем.
11. Дайте определение факторного эксперимента. Что такое фактор и отклик?
12. Как провести однофакторный и двухфакторный эксперименты?
13. Какие компоненты включает план эксперимента?
14. Как определить количество прогонов имитационной модели для однофакторного и двухфакторного экспериментов?
15. В чем заключается задача дисперсионного анализа результатов эксперимента?
16. Какой фактор является значимым? Как определить значимость фактора?
17. Может ли значимость фактора измениться?
18. С какой целью проводится оптимизирующий эксперимент? Каков порядок его проведения?
19. Какие результаты получаются при проведении оптимизирующего эксперимента?
20. Для чего используется уравнение отклика, которое строится в результате оптимизирующего эксперимента?

## Приложение Б

### Операторы описания блоков GPSS World

В поле операции пользователь должен записать обозначение блока. Задание исходных данных, необходимых для выполнения операций, соответствующих блоку, производится в поле операндов. Всего может быть 7 операндов, значения которых определяются типом блока. Операнды обозначаются буквами А, В, С, D и т.д. Если у блока несколько операндов, они разделяются запятыми, вместо пропущенного операнда ставится запятая.

Ниже приводится список блоков GPSS и дается их краткая характеристика:

**ADVANCE** – задержка сообщения на определенное время, с включением его в список будущих событий;

**ALTER** – проверка и модификация сообщения в группе;

**ASSEMBLE**– вывод из модели одного или нескольких сообщений;

**ASSIGN** – модификация параметров сообщений;

**BUFFER** – размещение сообщения в списке текущих событий последним в своем приоритетном уровне;

**COUNT** – подсчет числа элементов заданного множества, удовлетворяющих указанному условию;

**DEPART** – вывод сообщения из очереди;

**ENTER** – захват сообщением всего или части многоканального устройства;

**EXECUTE** – принудительная обработка сообщения некоторым блоком модели;

**FAVAIL** – смена состояния устройства на состояние «готово к использованию»;

**FUNAVAIL** – смена состоянии устройства на состояние «не готово к использованию»;

GATE – проверка состояния объекта модели и изменение направления движения потока сообщений;

GATHER – накопление нескольких связанных друг с другом сообщений (ансамбль сообщений);

GENERATE – создание сообщения и размещение его в списке будущих событий;

INDEX – изменение параметров сообщения;

JOIN – размещение членов в числовой группе или в группе сообщений;

LEAVE – освобождение всего или части многоканального устройства, пересылка сообщения в список пользователя;

LOGIC – модификация логического ключа;

LOOP – организация цикла с уменьшением значения параметра сообщения;

MARK – сохранение значения системного времени в параметре сообщения;

MATCH – ожидание сообщением связанных с ним сообщений в других блоках модели С синхронизация сообщений;

MSAVEVALUE – присваивание значений элементам матриц;

PREEMPT – прерывание обработки обрабатываемого устройством сообщения и захват устройства активным сообщением;

PRIORITY – изменение приоритета сообщения;

QUEUE – включение сообщения в очередь;

RELEASE – освобождение устройства сообщением;

REMOVE – удаление члена из числовой группы или группы сообщений;

RETURN – освобождение захваченного устройства;

SAVAIL – изменение состояния многоканального устройства на состояние «готово к использованию»;

SAVEVALUE – присваивание значений ячейке сохраняемых величин;

SCAN – просмотр содержимого группы до выполнения некоторого условия;

SEIZE – занятие устройства или ожидание его освобождения для последующего занятия;

SELECT – выбор блока для дальнейшего продвижения сообщения с использованием значения параметра сообщения;

SPLIT – создание копии сообщения с адресацией ее в некоторый блок;

SUNAVAIL – изменение состояния многоканального устройства на состояние «не готово к использованию»;

TABULATE – обновление данных таблицы;

TERMINATE – уничтожение сообщения, уменьшение счетчика завершенных сообщений;

TEST – проверка условий и модификация направления движения потока сообщений;

TRACE – установка флага трассировки для активного сообщения;

TRANSFER – пересылка сообщения на указанный блок;

UNLINK – вывод сообщения из списка пользователя;

UNTRACE – сброс флага трассировки для активного сообщения.

## Приложение В

### Операторы описания данных и контроля управления GPSS World

Для описания многоканальных устройств, переменных, таблиц и т.д. требуются операторы описания объектов. Информация, необходимая для контроля за процессом моделирования, задается с помощью управляющих операторов GPSS. Ниже приводится список операторов описания данных и контроля управления GPSS с краткими комментариями по их назначению:

FVARIABLE – определение булевских переменных;

CLEAR – приведение модели к исходному состоянию;

END – завершение работы GPSS;

EQU – присвоение целых значений именам;

FUNCTION – определение функций;

FVARIABLE – определение переменных с плавающей точкой;

INITIAL – присвоение или модификация значений ячеек охраняемых величин или элементов матриц ячеек сохраняемых величин;

MATRIX – определение матриц ячеек сохраняемых величин;

RESET – сброс статистики без сброса датчиков случайных чисел и таймера абсолютного времени;

RMULT – установка начальных значений одного или более генераторов случайных чисел;

SIMULATE – установка предела времени моделирования для последующей имитации;

START – установка значения счетчика завершения и запуск процесса моделирования;

STORAGE – определение многоканальных устройств;

TABLE – определение таблиц;

VARIABLE – определение целых переменных.

## Приложение Г

### Сообщения GPSS World об ошибках

A transaction is blocked on an impossible condition – сообщение заблокировано при невыполнимых условиях.

A transaction tried to SEIZE or PREEMPT its own facility – сообщение пытается занять (SEIZE) или захватить (PREEMPT) уже занятое им устройство.

A transaction which owns a facility is attempting to terminate – сообщение, которое владеет устройством, пытается выйти из системы.

A transaction which was preempted at a facility tried to SEIZE or PREEMPT it – сообщение, обработка которого устройством была прервана, пытается занять или захватить его.

Arithmetic overflow – арифметическое переполнение.

Assembly count was not positive – счетчик объединения положительный.

Attempt to release an idle facility – попытка освободить незанятое устройство.

Attempt to release an unwonted facility – попытка освободить не свое устройство.

Attempt to release more storage than existed – попытка освободить большую емкость многоканального устройства, чем определено.

Block Index is not positive – индекс блока неположительный.

Block index is too big – индекс блока слишком большой.

Block limit error out of blocks – ошибка границы блока, выход из блоков.

Block limit is too small – граница блока слишком маленькая.

Circular reference in expressions – циклическая зависимость в выражении.

Conjugate block is not a MATCH block – парные блоки не являются MATCH блоками.

Count block ran out of entities – счетчик блоков вышел за допустимые границы.

Edit from a program file is not allowed – редактирование из программного файла не разрешается.

Entity number is too high – номер объекта слишком большой.

Error priming this or another GENERATE block – ошибка определения блока GENERATE.

Entity is not a BVARIABLE – объект не BVARIABLE.

Entity is not a VARIABLE – объект не VARIABLE.

Error during evaluation of PLOT argument – ошибка при оценке аргумента PLOT.

Error in expression – ошибка в выражении.

EXECUTE block cannot act on another EXECUTE block – блок EXECUTE не может оказывать действия на другой EXECUTE блок.

Expression too long – выражение слишком длинное.

First cumulative probability must be 0 – первая координата аргумента непрерывной случайной функции должна быть 0.

Illegal attempt to make queue entity content negative – незаконная попытка создать очередь с отрицательным содержимым.

Illegal combination of operands in REMOVE block – незаконная комбинация операндов в блоке REMOVE.

Illegal combination of operands in UNLINK block – незаконная комбинация операндов в блоке UNLINK.

Illegal SNA class in COUNT or SELECT block – неправильный тип СЧА в блоках COUNT или SELECT.

Internal error – внутренняя ошибка.

Internal string error at location xxxx – системная ошибка размещения строки xxxx.

Invalid action after savable program has been deleted – недопустимое действие после удаления рабочей программы.

Invalid arithmetic expression – недопустимое арифметическое выражение.

Invalid character – недопустимый символ.

Invalid character in program file – недопустимый символ в программном файле.

Invalid cumulative probability distribution – недопустимое распределение вероятности.

Invalid discrete function argument – недопустимый аргумент дискретной функции.

Invalid file specification – недопустимая спецификация файла.

Invalid function follower statement – недопустимое описание значений функции.

Invalid function statement – недопустимый оператор функции.

Invalid (nonincreasing) X values – недопустимое (невозрастающее) значение  $x$ .

Invalid 0 or negative argument – недопустимый нулевой или отрицательный аргумент.

L or M type functions cannot be random – функции типа L или M не могут быть случайными.

Labels may not be keywords – метки не могут быть СЧА или ключевыми словами.

Line number overflow. Please renumber – превышен максимально допустимый номер строки программы. Пожалуйста, перенумеруйте.

List function argument is too large – список аргументов функции слишком большой.

Matrix is too large (8K entries) – матрица слишком большая.

Memory request is too big – требуемая память слишком большая.

More than 50 items in a function – более чем 50 элементов данных в функции.

Name has not been given a value – имени не было дано значение.



Negative A operand in SPLIT block – отрицательный операнд A в блоке SPLIT.

Negative storage request – отрицательная запрашиваемая емкость многоканального устройства.

Negative time Increment – отрицательный прирост времени.

Non-positive list function argument – неположительный аргумент табличной функции.

Not saved – не сохранено.

Not enough memory to store statement – не достаточно памяти для записи оператора.

Operand B of QUEUE or DEPART is negative – операнд B блоков QUEUE или DEPART отрицателен.

Out of memory. Press <SPACE> to stop, any other key uses program listing space – выход из памяти. Нажмите <SPACE> для останова, любая другая клавиша использует программу листания пространства.

Overflow – переполнение.

Parenthesis error in expressions – ошибка в выражении, в скобках.

Program line not found – строка программы не найдена.

Reference to a non-existent block – ссылка на несуществующий блок.

Reference to a non-existent function – ссылка на несуществующую функцию.

Reference to a non-existent matrix savevalue – ссылка к несуществующей матрице сохраняемых величин.

Reference to a non-existent parameter – ссылка на несуществующий параметр.

Reference to a non-existent savevalue – ссылка к несуществующей сохраняемой величине.

Reference to a non-existent storage – ссылка к несуществующему устройству.

Reference to a non-existent table – ссылка к несуществующей таблице.

Reference to a non-existent variable entity – ссылка на несуществующую переменную.

Remove option was not specified – Remove – опция не определена.

Remove option was used with no destination – опция использована не по назначению.

Result savevalue entity number must be positive – номер сохраняемой величины должен быть положительным.

Storage request exceeds total capacity – запрос емкости многоканального устройства превышает полную емкость.

System error. Code xxH – системная ошибка. Код xxH.

The disk is full – диск заполнен.

The file directory is full – каталог файлов полон.

The matrix row number is too large – номер строки матрицы слишком большой.

The matrix row number is not positive – номер строки матрицы не положительный.

The matrix column number is too large – номер столбца матрицы слишком большой.

The matrix column number is not positive – номер столбца матрицы не положительный.

The matrix offset exceeds 8192 elements – содержимое матрицы превышает 8192 элемента.

The program line is too long – строка программы слишком длинная.

The required label is missing – требуемая метка пропущена.

The simulation has been perturbed by an interaction. Use RESET and then rerun – при моделировании нарушилось содержимое файла результатов. Используйте RESET и затем перезапустите.

The table has more then 8191 frequency classes – таблица не может иметь более чем 8191 частотных классов.

The upper count limit is too low – верхний предел счетчика слишком низкий.

There are no group members – нет элементов групп.

There are no transactions on userchain entities – нет сообщений в цепочках пользователя.

There are no transactions – нет сообщений.

There is no positive treatment level – обработка неположительного уровня.

There is no transaction for the parameter refence – нет сообщения для упоминаемого параметра.

There is no transaction for this SNA evaluation – нет сообщений для вычисления СЧА.

Use of nonpositive entity number – использование неположительного номера объекта.

You cannot change a block location value in an EQU statement – вы не можете изменить значение метки блока в операторе EQU.

You must specify an SNA – вы должны определить СЧА.

\*\*\*\*\*Formatting overflow\*\*\*\*\*– переполнение формата.

\*\*\*\*\*Stack overflow\*\*\*\*\* – переполнение стека.

Other Error Codes – другие коды ошибок.