

**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**

**РАЗРАБОТКА МОДЕЛИ  
ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ  
СИСТЕМЫ, ПЛАНИРОВАНИЕ И ПРОВЕДЕНИЕ  
ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ В СРЕДЕ  
GPSS WORLD**

**Учебное пособие**

по курсу «Моделирование информационно-вычислительных систем»

Самара 2021

Составитель: к.т.н., доц. Симонова Е. В.

УДК 519.87

**Разработка модели информационно-вычислительной системы, планирование и проведение имитационных экспериментов в среде GPSS World / Е.В. Симонова.** – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2021. – 50 с.

Методическое пособие предназначено для студентов, обучающихся по направлению 09.03.01 – «Информатика и вычислительная техника», выполняющих лабораторные работы по курсу «Моделирование информационно-вычислительных систем».

Представленное 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, получение и интерпретацию результатов моделирования.

Пособие содержит анимированные иллюстрации всех этапов процесса разработки и реализации модели, что имеет большое учебно-методическое значение и необходимо при самостоятельной работе студентов во время выполнения ими лабораторных работ. Учебное пособие содержит варианты заданий для выполнения лабораторных работ.

Учебное пособие разработано на кафедре информационных систем и технологий

УДК 519.876.5

ББК 22.18я73

ISBN

© Самарский университет, 2021

## СОДЕРЖАНИЕ

Введение.....	4
1 Цель ЛАБОРАТОРНОЙ РАБОТЫ.....	5
2 Задание .....	5
3 Реферат .....	6
4 Постановка задачи.....	7
5 Анализ возможных методов решения .....	8
5.1 Аналитический метод .....	8
5.2. Метод имитационного моделирования .....	9
6 Разработка концептуальной модели.....	10
6.1 Таблица функционального соответствия элементов модели и элементов реальной системы .....	11
6.2 Алгоритмы формирования случайных величин .....	11
7 Выбор программных средств моделирования.....	12
8 Разработка алгоритма моделирования и программы имитационной модели.....	13
8.1. Разработка алгоритма моделирования .....	13
8.1.1 Описание элементов модели .....	13
8.2 Алгоритм моделирования.....	14
8.3 Описание программной реализации имитационной модели .....	15
8.4 Программа имитационной модели .....	15
9 Организация экспериментов с моделью .....	16
9.1 Анализ и оценка результатов моделирования.....	16
9.2 Оценка точности результатов моделирования.....	18
9.3 Факторный эксперимент в системе GPSS World .....	19
9.4 Проведение дисперсионного анализа в системе GPSS World .....	23
9.4.1 Дисперсионный анализ ANOVA в планировании эксперимента .....	23
9.4.2 Последовательность проведения дисперсионного анализа для однофакторного эксперимента.....	24
9.4.3 Библиотечная процедура ANOVA .....	27
9.5 Проведение двухфакторного эксперимента .....	27
9.5.1 План двухфакторного эксперимента .....	27
9.5.2 Последовательность проведения дисперсионного анализа для двухфакторного эксперимента .....	32
9.6 Поиск наилучших решений. Оптимизирующий эксперимент .....	35
9.6.1 Организация оптимизирующего эксперимента.....	36
10 Выводы и рекомендации по использованию модели .....	40
11 Список использованных источников. Приложения .....	40
11.1. Список использованных источников .....	40
11.2. Приложения .....	40
Заключение .....	41
Библиографический список.....	42
Приложения .....	43
Приложение А Операторы описания блоков GPSS World.....	43
Приложение Б Операторы описания данных и контроля управления GPSS World.....	45
Приложение В Сообщения GPSS World об ошибках .....	46

## 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.

Разделы учебного пособия последовательно раскрывают порядок выполнения лабораторной работы с использованием языка GPSS для реализации имитационных моделей.

Учебное пособие может быть полезно широкому кругу читателей, практикующихся в области компьютерных наук.

## 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 у студентов умение применять знания и навыки в области имитационного моделирования для решения конкретных инженерных задач проектирования и исследования систем. В задачи лабораторной работы входят [7,8]:

- развитие у студентов 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 приводится описание последовательности этапов выполнения лабораторной работы.

## 2 ЗАДАНИЕ

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 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.

**Задание на моделирование** 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:

Значения параметров			
$\lambda[\text{с}^{-1}]$	$k$	$\alpha[\text{с}^{-1}]$	$q[\text{с}]$
10	2	1	0,25

### 3 РЕФЕРАТ

В тексте реферата необходимо отобразить приведенную в проекте информацию в такой последовательности:

- объект исследования;
- цель работы;
- методы исследования;
- результаты.

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.

## 4 ПОСТАНОВКА ЗАДАЧИ

В данном разделе приводится содержательная постановка задачи, определяются цели исследования, внешние воздействия и ограничения, которые накладываются на систему.

Целью задачи является разработка модели распределения времени центрального процессора (ЦП) и последующая оценка распределения времени ответа и среднего времени ответа путем имитации обслуживания  $N$  заявок. В задании определяются:

- 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 1). 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.

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 выполнения работы необходимо ознакомиться с алгоритмом распределения времени центрального процессора в системах коллективного пользования и получить практический опыт моделирования вычислительных систем.

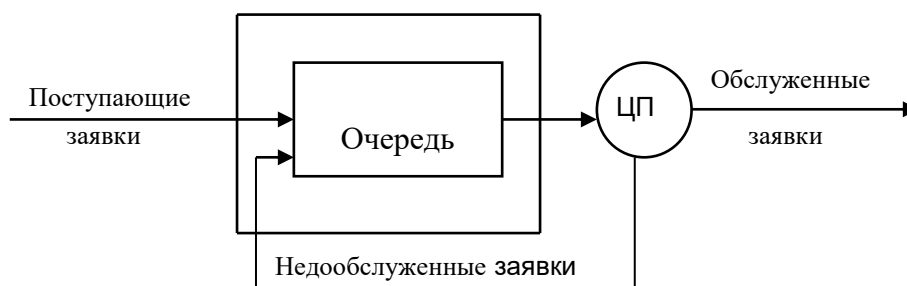


Рисунок 1 – Схема системы с круговым опросом

## 5 АНАЛИЗ ВОЗМОЖНЫХ МЕТОДОВ РЕШЕНИЯ

В данном разделе анализируются методы решения поставленной задачи, указываются их преимущества и недостатки, дается обоснование выбора метода решения. Аргументируется необходимость применения метода имитационного моделирования. 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.

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 такого неоднородного потока должно основываться на использовании адекватных вероятностных моделей информационных потоков [7], мерой же эффективности СРВ являются характеристики времени пребывания заявки в системе (т.е. *времени ответа*), в частности:

- распределение времени ответа  $S(y) = P\{\text{время ответа} \leq y\}$ ;
- 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. Указанные характеристики времени ответа при определенных упрощающих предположениях рассчитываются аналитически [3,7], однако в общем случае их определение связано с использованием методов имитационного моделирования.

### 5.1 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 формуле (1) [3]:



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

$$\rho = \frac{\lambda}{\mu}, \quad (2)$$

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 [3]. Для неустановившегося режима рассчитать параметры системы аналитически невозможно.

## 5.2. 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.

## 6 РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ

В данном разделе 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 (3)

$$f(x) = \lambda e^{-\lambda x}; x \geq 0, \lambda \geq 0 \quad (3)$$

### **Входные переменные:**

- параметр  $\lambda$  экспоненциального распределения интервалов времени между заявками на обслуживание;
- параметр  $\alpha$  распределения Эрланга с плотностью вероятностей (4), описывающего требуемое время обслуживания заявки;
- порядок  $k$  распределения Эрланга, описывающего требуемое время обслуживания заявки (4).

$$f(\tau_s) = \alpha \frac{(\alpha \tau_s)^{k-1}}{(k-1)!} e^{-\lambda \tau_s} \quad (4)$$

### Выходные переменные:

- среднее время обслуживания заявок  $T$ .

В данном разделе приводятся:

- таблица функционального соответствия элементов моделируемой системы и элементов модели, содержательное значение всех статических и динамических объектов с описанием их свойств;
- описание алгоритмов формирования случайных величин.

## 6.1 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 (**Ошибка! Источник ссылки не найден.**).

Таблица 1 – Таблица функционального соответствия

Элемент реальной системы	Элемент модели
1 с	100 единиц модельного времени
Заявка на обслуживание	Транзакт
Экспоненциальное распределение с интенсивностью 1	Функция EXPON
Требуемое время обслуживания	Переменная VV
Время обслуживания	1-й параметр транзакта, СЧА P1
Входная очередь заявок	Очередь QUE
Обслуживающее устройство	Одноканальное устройство CPU
Таблица для сбора статистики	Таблица TT
Время пребывания заявки в модели, включая обслуживание и нахождение в очереди	Время жизни транзакта – СЧА M1

## 6.2 Алгоритмы формирования случайных величин

1. Для моделирования случайной величины, распределенной по экспоненциальному закону, следует использовать метод обратной функции (5) [7].

$$x_i = -\frac{1}{\lambda} \ln R_i; \quad (5)$$

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

2. При моделировании случайных величин, распределенных по закону Эрланга с параметром  $\alpha$ , используют то обстоятельство, что этому закону распределения подчиняется сумма  $k$  случайных величин (6)

$$\sum_{i=1}^k \tau_i = \tau_{\gamma} , \quad (6)$$

независимых друг от друга и распределенных по экспоненциальному закону с параметром  $\lambda_{\gamma} = k * \alpha$ . В этом случае величина  $\tau_{\gamma}$  распределена по закону Эрланга  $k$ -го порядка с плотностью вероятности (7)

$$f(\tau_{\gamma}) = \alpha \frac{(\alpha \tau_{\gamma})^{k-1}}{(k-1)!} e^{-\lambda \tau_{\gamma}} . \quad (7)$$

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

VV FVARIABLE 50#(FN\$EXPON+ FN\$EXPON)

## 7 ВЫБОР ПРОГРАММНЫХ СРЕДСТВ МОДЕЛИРОВАНИЯ

В данном разделе 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 [1, 4, 5, 6, 9]. 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.

Язык GPSS обладает всеми свойствами, перечисленными выше. 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 среде GPSS World, предназначенной для использования в среде Windows [9, 10, 11].

## **8 РАЗРАБОТКА АЛГОРИТМА МОДЕЛИРОВАНИЯ И ПРОГРАММЫ ИМИТАЦИОННОЙ МОДЕЛИ**

### **8.1. Разработка алгоритма моделирования**

В данном разделе приводятся:

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

#### **8.1.1 Описание элементов модели**

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 обслу-

живания» следует уменьшить 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 время требуемого обслуживания заявки процессором связана с индивидуальным определением переменных и функций GPSS.

6. Для получения оценок распределения времени ответа  $S(y)$  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  $\min(q, m_x/3)$ , где  $q$  – наименьший квант обслуживания, предоставляемый заявке,  $m_x$  – математическое ожидание требуемого времени обслуживания заявки.

## 8.2 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 2).

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
3 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
10 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
15 OUT	TABULATE	0	0	0	19
16 TER	TERMINATE	0	0	0	20

Рисунок 2 – Алгоритм модели

### 8.3 Описание программной реализации имитационной модели

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

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.

### 8.4 Программа имитационной модели

В данном разделе представляется листинг программы модели с комментариями.

Программная реализация имитационной модели распределения времени обслуживания заявки центральным процессором представлена на Рисунке 3.

```
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 3 – Программа модели**

## **9 ОРГАНИЗАЦИЯ ЭКСПЕРИМЕНТОВ С МОДЕЛЬЮ**

### **9.1 Анализ и оценка результатов моделирования**

В данном разделе приводятся результаты компьютерных экспериментов в виде протоколов моделирования, графиков, таблиц, а также качественные и количественные оценки результатов моделирования.

Результаты моделирования представлены в протоколе моделирования (Рисунок 4) и в гистограмме распределения времени обслуживания транзактов (Рисунок 5).



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
TT		10002.000
VV		10001.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
MET	1	GENERATE	100		0	0
	2	ASSIGN	100		0	0
	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
QUANT	9	TRANSFER	100		0	0
	10	ADVANCE	570		0	0
	11	ASSIGN	570		0	0
	12	RELEASE	570		0	0
OUT	13	BUFFER	570		0	0
	14	TRANSFER	570		0	0
	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

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

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM. %
TT	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 4 – 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*).

Гистограмму распределения вероятностей распределения случайной величины – времени обслуживания заявок можно видеть на Рисунок 5.

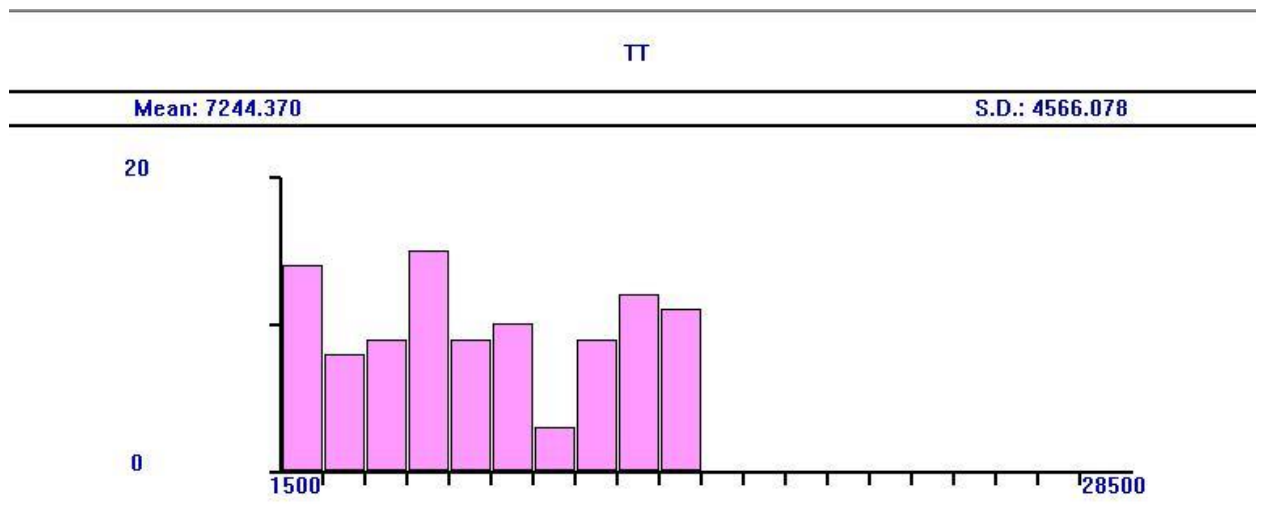


Рисунок 5 – Гистограмма распределения времени обслуживания транзактов

## 9.2 Оценка точности результатов моделирования

В данном разделе 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 (8) [7]

$$N = t_{\frac{1+\delta}{2}}^2 \frac{p(1-p)}{\varepsilon^2}, \quad (8)$$

где

$p$  – вероятность наступления события,

$\varepsilon$  – точность модели (абсолютная),

$\delta$  – достоверность модели.

Если в качестве показателя эффективности используется среднее значение некоторой случайной величины, число прогонов имитационной модели оценивается по формуле (9) [7]

$$N = t_{\frac{1+\delta}{2}}^2 \frac{\sigma^2}{\varepsilon}, \quad (9)$$

где

$\sigma$  – среднееквадратическое отклонение случайной величины,

$\varepsilon$  – точность модели (абсолютная),

$\delta$  – достоверность модели.

В данной задаче оценивается среднее время обслуживания заявок, поэтому для расчета числа реализаций модели следует использовать формулу (9).

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 среднееквадратическое отклонение времени пребывания транзактов в модели  $\sigma = 4566.078$  единиц модельного времени. При подстановке значений в формулу (9) получим

$$N = 3.8416 * \frac{20848356}{0.0001} = 800910444096.$$

Таким образом, для обеспечения заданной точности и достоверности необходимо 800910444096 реализаций имитационной модели.

### 9.3 Факторный эксперимент в системе GPSS World

В данном разделе 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. В этом случае получаем *ортогональное разбиение уровней*.

При проведении многофакторных экспериментов требуется, чтобы для дисперсионного анализа использовался ортогональный эксперимент. Это

означает, что оценки в пределах анализа должны быть некоррелированные. На практике использование одних и тех же значений при выполнении экспериментов в пределах каждой комбинации уровней факторов гарантирует ортогональность.

**Факторный эксперимент** [2] 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 картина влияния 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.

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 ГСП
```

```
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

## 9.4 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 влияние 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 не более чем 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.

### 9.4.1 Дисперсионный анализ ANOVA в планировании эксперимента

Неявно в ANOVA используется аддитивная математическая модель, которая определяет компоненты изменения в наблюдениях. Ее называют статистической моделью.

Статистическая модель для анализа данных экспериментов с одним фактором  $A$  имеет вид (10) [9]:

$$y_{ik} = \mu + \alpha_i^A + e_{ik}, \quad (10)$$

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 (11):

$$y_{ijk} = \mu + \alpha_i^A + \alpha_j^B + \alpha_{ij}^{AB} + e_{ijk}, \quad (11)$$

где

$i$  – номер уровня фактора А,

$j$  – номер уровня фактора В,

$k$  – номер наблюдения,

$y_{ijk}$  – отклик,

$\mu$  – общее среднее по всем наблюдениям,

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

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

$\alpha_{ij}^{AB}$  – взаимодействие фактора А на уровне  $i$  и фактора В 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 (10)-(11), заканчивается построением таблицы ANOVA, в которой анализируется влияние факторов А, В, 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*.

#### **9.4.2 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 6), задать длительность моделирования с помощью таймера (файл *lab1\_ANOVA\_1f.gps*).



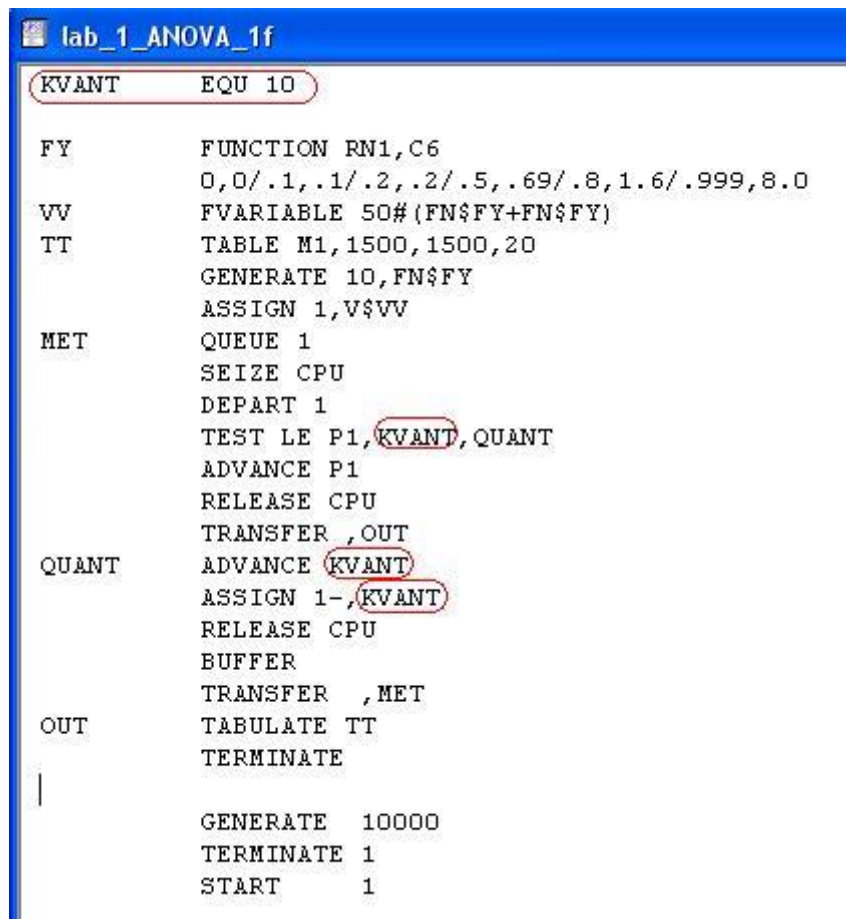


Рисунок 6 – Программа модели для однофакторного эксперимента

2. Создать текстовый файл *KVANT\_1f.txt*, содержащий план однофакторного эксперимента, в том же каталоге, что и модель *lab1\_ANOVA\_1f.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 ОК (Рисунок 7).

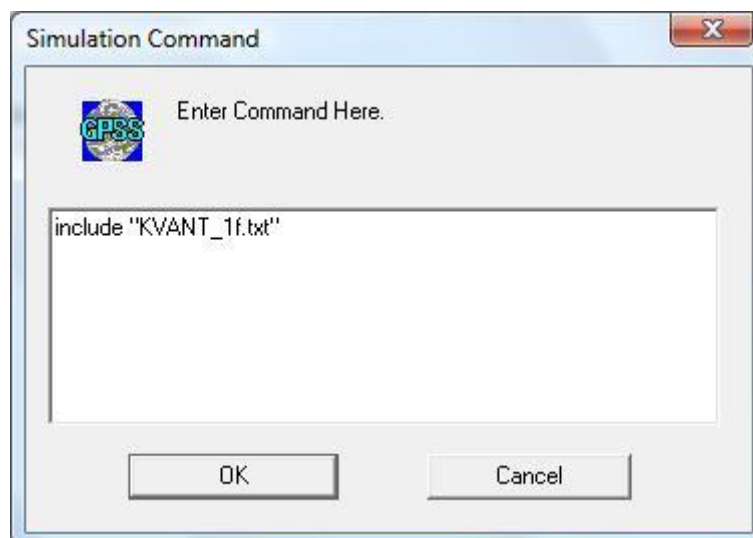


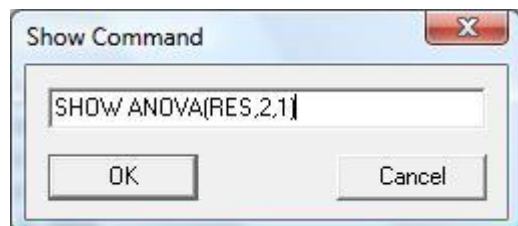
Рисунок 7 – Диалоговое окно *Simulation Command*

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*».

5. С помощью пункта меню *Command/Show* открыть диалоговое окно *Show Command* и в строке ввода ввести `SHOW ANOVA(RES,2,1)` (Рисунок 8), 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 8 – Задание на выполнение дисперсионного анализа для однофакторного эксперимента**

По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 9).

**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 9 – Результаты дисперсионного анализа для однофакторного эксперимента**

### 9.4.3 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,
- *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 9), фактор А – величина кванта времени центрального процессора – является значимым, т.к. величина критерия  $F = 8.002$  больше порогового значения, равного 3.3.

## 9.5 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.

### 9.5.1 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 MSVALUE 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 2). Каждый блок обеспечивает получение результата для заданного сочетания уровней факторов в одном прогоне модели.

**Таблица 2 – Сочетания уровней факторов для двухфакторного эксперимента**

№ 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
TZA        EQU 10

```

```

;значение 2-го уровня 1-го фактора
;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
TZA        EQU 100

```

```

;значение 2-го уровня 1-го фактора
;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

```

### 9.5.2 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 10), 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*.
4. 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*».



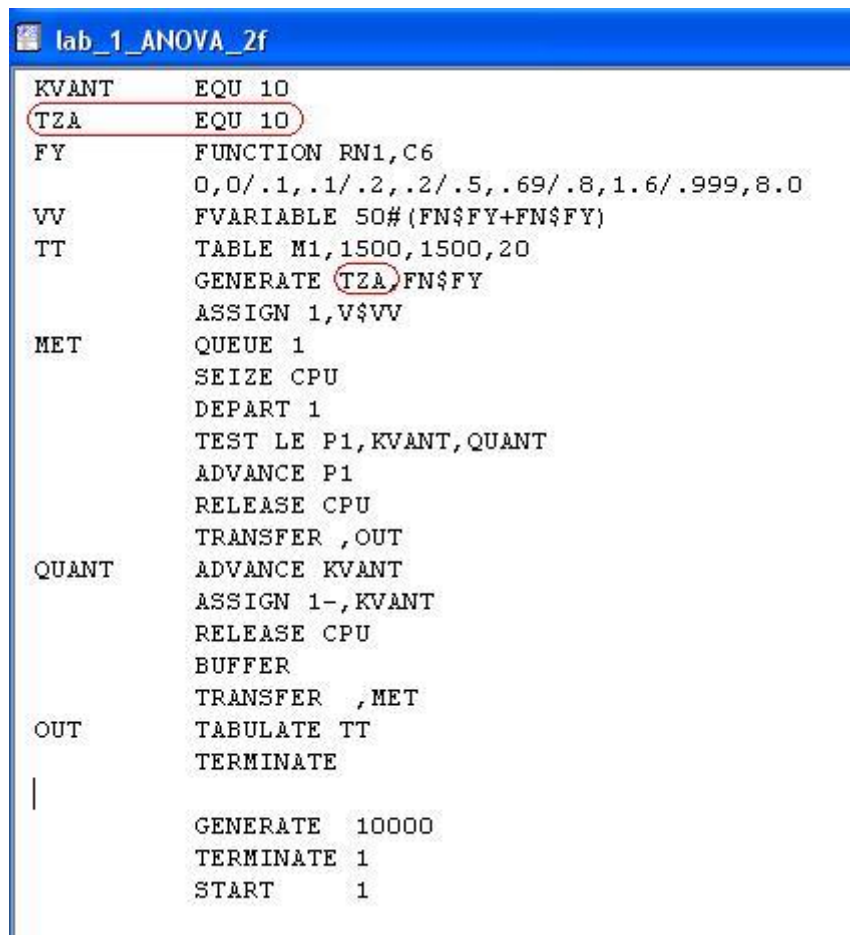


Рисунок 10 – Программа модели для двухфакторного эксперимента

С помощью пункта меню *Command/Show* открыть диалоговое окно *Show Command* и в строке ввода ввести *SHOW ANOVA(RES,3,2)* (Рисунок 11), 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  $B = 2$ .

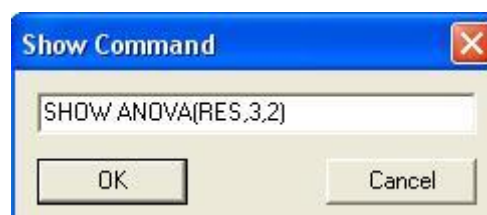


Рисунок 11 – Задание на выполнение дисперсионного анализа для двухфакторного эксперимента

По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 12).

04/09/09 17:54:44

## ANOVA

04/09/09 17:54:44

04/09/09 17:54:44 04/09/09 17:54:44 04/09/09 17:54:44	Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)
04/09/09 17:54:44	A	13901565.879	1	13901565.879	1.749	4.11
04/09/09 17:54:44	B	55000938.998	1	55000938.998	6.922	4.11
04/09/09 17:54:44	AB	3585829.986	1	3585829.986	0.451	4.11
04/09/09 17:54:44	Error	286057295.230	36	7946035.979		
04/09/09 17:54:44	Total	358545630.093	39			
04/09/09 17:54:44						
04/09/09 17:54:44						
04/09/09 17:54:44 04/09/09 17:54:44 04/09/09 17:54:44	Treatment Level A B	Count	Mean	Minimum	Maximum	95% C.I. (SE)
04/09/09 17:54:44	1 1	10	4409.806	0.000	9944.976	[ 2626.995, 6192.616 ]
04/09/09 17:54:44	1 2	10	1465.760	0.000	7203.603	[ -317.051, 3248.571 ]
04/09/09 17:54:44	2 1	10	2631.939	0.000	8390.521	[ 849.128, 4414.749 ]
04/09/09 17:54:44	2 2	10	885.529	0.000	2917.204	[ -897.282, 2668.339 ]
04/09/09 17:54:44						
04/09/09 17:54:44	2818.8714016					

Рисунок 12 – Результаты дисперсионного анализа для двухфакторного эксперимента

Согласно результатам, представленным в таблице ANOVA для двухфакторного эксперимента (Рисунок 12), 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.

Изменим уровни фактора KVANT. Создадим командный текстовый файл (назовем его *KVANT\_2f\_2.txt*), содержащий *N* блоков команд для каждого сочетания уровней факторов (Таблица 3).

Таблица 3 – Сочетания уровней факторов для двухфакторного эксперимента

№ эксперимента	KVANT (A)	TZA (B)
1	10	10
2	10	100
3	100	10
4	100	100

По завершении эксперимента в журнале процесса моделирования появляется таблица ANOVA, содержащая результаты дисперсионного анализа (Рисунок 13).

Согласно результатам, представленным в таблице ANOVA для двухфакторного эксперимента (Рисунок 13), фактор  $A$  – величина кванта времени центрального процессора – является значимым, т.к. величина критерия  $F = 10.778$  больше порогового значения, равного 4.11; фактор  $B$  – величина среднего времени между поступлениями заявок в систему – является незначимым, т.к. величина критерия  $F = 1.482$  меньше порогового значения, равного 4.11; взаимодействие между факторами  $A$  и  $B$  является значимым, т.к. величина критерия  $F = 8.597$  больше порогового значения, равного 4.11. Такие результаты эксперимента означают, что при данных уровнях факторов на отклик системы оказывает влияние величина кванта времени центрального процессора.

04/09/09 18:27:00	ANOVA					
04/09/09 18:27:00						
04/09/09 18:27:00	Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F [p=.05]
04/09/09 18:27:00	A	54261840.549	1	54261840.549	10.778	4.11
04/09/09 18:27:00	B	7459041.268	1	7459041.268	1.482	4.11
04/09/09 18:27:00	AB	43280162.598	1	43280162.598	8.597	4.11
04/09/09 18:27:00	Error	181243379.314	36	5034538.314		
04/09/09 18:27:00	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]
04/09/09 18:27:00	A B					
04/09/09 18:27:00	1 1	10	4409.806	0.000	9944.976	[ 2990.716, 5828.895 ]
04/09/09 18:27:00	1 2	10	1465.760	0.000	7203.603	[ 46.670, 2884.849 ]
04/09/09 18:27:00	2 1	10	0.000	0.000	0.000	[ -1419.090, 1419.090 ]
04/09/09 18:27:00	2 2	10	1216.731	0.000	3428.583	[ -202.359, 2635.821 ]
04/09/09 18:27:00						
04/09/09 18:27:00	2243.7776883					

Рисунок 13 – Результаты дисперсионного анализа для двухфакторного эксперимента при измененных значениях уровней

## 9.6 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.

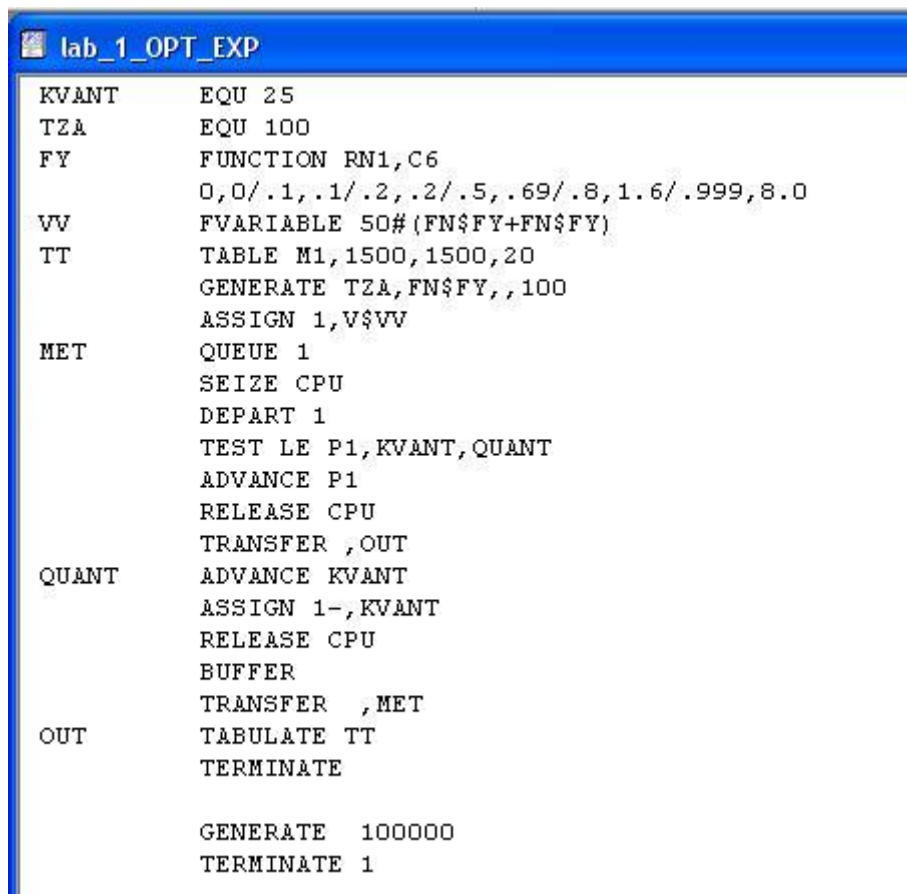
Оптимизирующий эксперимент предназначен для построения уравнения поверхности отклика для заданных факторов модели и поиска численного значения оптимума (сочетания таких значений факторов, при которых заданная функция принимает экстремальное значение – минимум или максимум). Максимально возможное количество 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.

### 9.6.1 Организация оптимизирующего эксперимента

1. Создать программу модели для проведения оптимизирующего эксперимента (файл *Lab\_1\_OPT\_EXP.gps*) (Рисунок 14). *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 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 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 14 – Программа модели для проведения оптимизирующего эксперимента

2. Добавить оптимизирующий эксперимент в созданную и отлаженную модель с помощью команд *Edit/Insert Experiment/optimizing* (Рисунок 15).

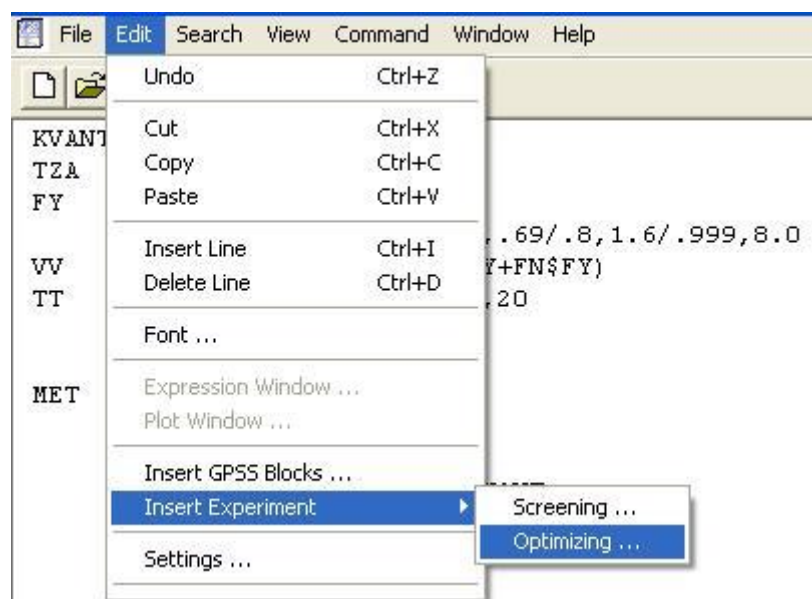


Рисунок 15 – Добавление оптимизирующего эксперимента

3. Заполнить поля диалогового окна *Optimizing Experiment Generator* (Рисунок 16).
- *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.

Optimizing Experiment Generator

Experiment Name: RSM

'Run Procedure' Name: DoTheRun

Initial Local Experimental Region		
Factor Name (User Variable)	Value 1	Value 2
A KVANT	10	25
B TZA	50	100
C		
D		
E		

Movement Limits	
Low Limit	High Limit
0	100
0	100
0	100
0	100
0	100

Result

Expression: M1

☐ Maximize ☒ Minimize

☒ Generate Run Procedure ☒ Load F12 with CONDUCT Command

Redirection Limit: 2

Buttons: Insert Experiment, Cancel, Help

Рисунок 16 – Определение параметров оптимизирующего 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 17). При нажатии на кнопку ОК автоматически сгенерированный код на языке PLUS будет добавлен в текст программы модели.

Run Procedure Generation

To Include a Run Procedure, edit and click OK.

```

PROCEDURE DoTheRun(Run_Number) BEGIN
  DoCommand("CLEAR OFF"); /* Must use OFF to preserve result */

  /* EXPAND THIS RMULT IF YOU HAVE MORE RNGs. */
  /* All Random Number Streams must have new seeds. */
  TEMPORARY CommandString;
  /* Evaluate before passing to DoCommand. */

```

Buttons: OK, Cancel

Рисунок 17 – 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 18).

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 +1.20627e+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 18 – Информация о ходе оптимизирующего эксперимента

## **10 ВЫВОДЫ И РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ МОДЕЛИ**

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

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  $TZA = 75$ .

## **11 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ. ПРИЛОЖЕНИЯ**

### **11.1. Список использованных источников**

Данный раздел содержит перечень источников, использованных при выполнении лабораторной работы. Указываются только те источники, на которые имеются ссылки в тексте пояснительной записки.

### **11.2. Приложения**

Данный раздел содержит вспомогательные материалы (расчеты вспомогательного характера, листинги программ, инструкции по использованию программ).



## ЗАКЛЮЧЕНИЕ

Компьютерное моделирование стало эффективным инструментом исследования и проектирования сложных систем вследствие гибкости, 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, имеющее место 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, широко 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.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Боев В. Моделирование систем. Инструментальные средства GPSS World. – Спб.: БХВ-Петербург, 2004.
2. Ермаков С.М., Жиглявский А.А. Математическая теория планирования эксперимента. – М.: Наука, 1987.
3. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 600с.
4. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004.
5. Кудрявцев Е.М., Добровольский А. Основы работы с универсальной системой моделирования GPSS World. – М.: Изд-во ассоциации строительных вузов, 2005.
6. Рыжиков Ю.И. Имитационное моделирование. Теория и технологии. – Спб.: КОРОНА принт; М.: Альтекс-А, 2004.
7. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб. для вузов – 6-е изд., стереотипное. – М.: Высш. шк., 2009.
8. Советов Б.Я. Моделирование систем. Практикум: Учеб. пособие для вузов / Б.Я. Советов, С.А. Яковлев. – 4-е изд., стереотипное. – М.: Высш. шк., 2009.
9. Томашевский В., Жданова Е. Имитационное моделирование в среде GPSS. – М.: Бестселлер, 2003.
10. GPSS World Tutorial Manual.  
<http://www.minutemansoftware.com/tutorial/t1.htm#Chapter%201>
11. GPSS World Reference Manual.  
<http://www.minutemansoftware.com/reference/r4.htm>
12. Методология и технология имитационных исследований сложных систем: современное состояние и перспективы развития: Монография / В.В. Девятков - М.: Вузовский учебник: ИНФРА-М, 2013. – 448 с.: 60х90 1/16. – (Научная книга). ISBN 978-5-9558-0338-8 (Вузовский учебник).

## ПРИЛОЖЕНИЯ

### Приложение А 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 A, B, C, 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 состояния 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 сооб-

щения;

**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 Б 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;
- QTABLE** – определение Q-таблиц;
- 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 В 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 B VARIABLE** – объект не B VARIABLE.
- **Entity is not a VARIABLE** – объект не VARIABLE.
- **Error during evalution 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 В 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.

Учебное издание

***Разработка модели информационно-вычислительной системы,  
планирование и проведение экспериментов в среде GPS World***

Учебное пособие

по курсу «Моделирование информационно-вычислительных систем»

Составитель: ***Симонова Елена Витальевна***

Самарский национальный исследовательский университет

имени академика С.П. Королева

(Самарский университет)

443086 Самара, Московское шоссе, 34