

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**«Ухтинский государственный технический университет»  
(УГТУ)**

# **Имитационное моделирование**

## **Контрольные работы**

Методические указания

Ухта, УГТУ, 2015

УДК 004.94(075.8)

ББК 32.81 я7

С 30

**Семериков, А. В.**

С 30 Имитационное моделирование. Контрольные работы [Текст] : метод. указания / А. В. Семериков. – Ухта : УГТУ, 2015. – 22 с.

Настоящие методические указания предназначены для выполнения контрольных работ по дисциплине «Имитационное моделирование». Они включают в себя пояснения для выполнения заданий и варианты заданий для контрольных работ. Методические указания предназначены для студентов дневной и заочной формы обучения по специальности  
Содержания указаний соответствует рабочей учебной программе.

**УДК 004.94(075.8)**

**ББК 32.81 я7**

Методические указания рассмотрены и одобрены кафедрой ВТИСиТ, протокол №4 от 19.12.14 г.

Рецензент: С. Г. Куделин, доцент кафедры ВТИСиТ, к.т.н.

Корректор: А. Ю. Васина. Технический редактор: Л. П. Коровкина.

План 2015 г., позиция 339.

Подписано в печать 31.08.2015 г. Компьютерный набор.

Объем 22 с. Тираж 100 экз. Заказ №298.

© Ухтинский государственный технический университет, 2015  
169300, Республика Коми, г. Ухта, ул. Первомайская, д. 13.  
Типография УГТУ.  
169300, Республика Коми, г. Ухта, ул. Октябрьская, д. 13.

## Оглавление

Введение .....	4
1. Определеление площади геометрической фигуры .....	5
2. Модель системы массового обслуживания .....	9
2.1 Механика моделирования СМО .....	10
2.2 Краткое описание языка GPSS .....	12
2.3 Работа кассы с двумя кассирами .....	13
2.4 Работа парикмахерской с одним мастером .....	16
2.5 Работа парикмахерской с несколькими мастерами при отказе от обслуживания .....	18
3. Варианты контрольных заданий .....	20
3.1 Задание 1 .....	20
3.2 Задание 2 .....	21
Библиографический список .....	22

## Введение

Моделирование является одним из эффективных способом изучения реального мира. Моделирование позволяет проводить исследования существующих в окружающем мире объектов не с самим объектом, а с его моделью. Анализ проведённых экспериментов позволяет принять взвешенное и логически оправданное решение по функциональным возможностям системы. Для принятия решения можно использовать и наблюдения за реально работающей системой, а также использовать интуитивное мышление, накопленный опыт. Однако, такой подход не позволяет всесторонне оценить наблюдаемый объект и может принести большие экономические потери. Кроме того, во многих случаях проведение экспериментов над реальной системой просто недопустимо.

Особенно важно проводить моделирование вновь создаваемых систем, так как появляется возможность еще на стадии проектирования оценить основные параметры и функции объекта. При этом можно сравнить эффективность различных управляющих решений в поисках наиболее подходящего варианта.

Различают два вида моделирования: математическое и имитационное моделирование. Математическое моделирование применяется для создания моделей физических процессов с использованием численных методов. Имитационное моделирование отражает структуру и поведение объекта. Компьютерный эксперимент позволяет установить характеристики, как всей системы, так и ее составных частей.

С помощью имитационного моделирования можно решать различные задачи. Существует различная классификация этих задач. Известна [3] следующая классификация: моделирование динамических систем, дискретно-событийное моделирование, системная динамика и агентное моделирование.

Имитационное моделирование представляет собой статистический эксперимент. Его результаты должны основываться на статистических проверках. Имитационный эксперимент должен удовлетворять следующим требованиям:

1. Наблюдения должны иметь стационарные распределения.
2. Наблюдения подчиняются нормальному распределению.
3. Наблюдения независимы.

Соблюдение этих требований гарантирует корректный сбор наблюдений при использовании имитационной модели. На практике выполнение этих требований вызывает большие трудности. Пути выполнения требований к имитационным моделям хорошо известны, хотя не всегда выполнимы [4].

Для разработки имитационных моделей имеются программные средства: Simulink, GPSS, AnyLogic. Эффективное использование этих инстру-

ментальных средств предполагает глубокие знания предметной области моделирования, знания в программировании, статистике, теории вероятности. В настоящем методическом указании используется язык моделирования GPSS (General Purpose System Simulation), который был создан для дискретно-событийного моделирования.

В рамках курса имитационного моделирования предлагается выполнить две контрольные работы.

Первая из них содержит задание, выполнение которого предполагает написание программы на каком-либо алгоритмическом языке, не ориентированном на имитационное моделирование. Для создания программы необходимы знания, как предметной области моделирования, так и какого-либо языка программирования.

Во второй контрольной работе рассматривается дискретно-событийное моделирование. Выполнение этой работы предполагает знания теории системы массового обслуживания (СМО) и специализированного языка имитационного моделирования (GPSS).

## 1. Определение площади геометрической фигуры

Дана сеточная функция

Точки	1	2	3
x	0	3,6	4,0
y	2,0	0,8	0

Необходимо найти площадь под этой функцией, полагая, что координаты промежуточных точек определяются на основе линейной зависимости.

Используя, полином Лагранжа первой степени построим линейную функцию, между первой и второй точками.

$$y_1 = 2 \frac{x - 3,6}{0 - 3,6} + 0,8 \frac{x - 0}{3,6 - 0} = -0,33x + 2, \quad (1.1)$$

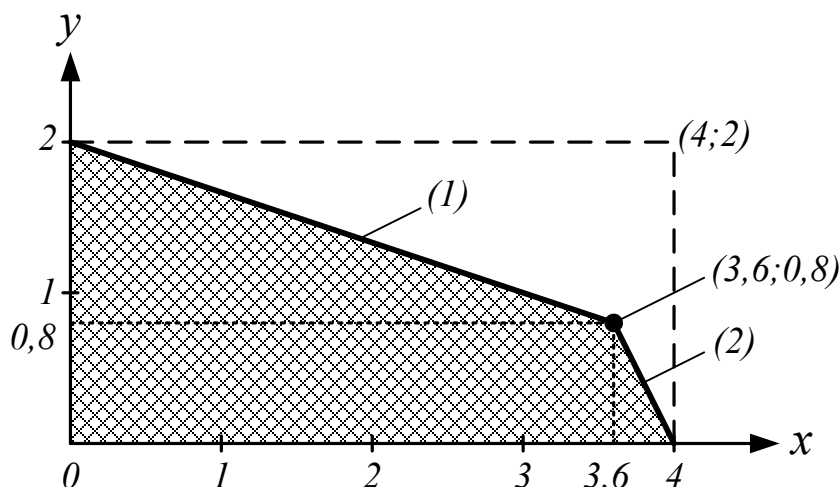
$$y_2 = 0,8 \frac{x - 4}{3,6 - 4} + 0 \frac{x - 3,6}{4 - 3,6} = -2x + 8. \quad (1.2)$$

Представим графики функций  $y_1$  и  $y_2$ .

Из графика и из сеточной функции видно, что значения  $y$  и  $x$  изменяются соответственно от 0 до 2 и от 0 до 4. Поэтому для оценки площади под рассматриваемой ломанной кривой заключим эту площадь в прямоугольник со сторонами 4 и 2. Определение площади обозначенной фигуры основано на предположении, что все точки, входящие в прямоугольник равновероятны. Предположим, была произведена случайная выборка, которая показала, что из

общего количества точек  $n$  в площадь фигуры попало  $m$  точек. Таким образом, с некоторой погрешностью площадь фигуры можно вычислить по формуле

$$\text{площадь фигуры} = m / n \cdot (\text{площадь прямоугольника}) = m / n \cdot 8.$$



При увеличении числа испытаний  $n$  эта площадь будет стремиться к истинному значению.

Координаты  $y$  и  $x$  точек прямоугольника можно представить как равномерно распределённые случайные величины с плотностью вероятностей

$$f(x) = 1/4 \quad 0 \leq x \leq 4 \quad (1.3)$$

$$f(y) = 1/2 \quad 0 \leq y \leq 2 \quad (1.4)$$

Обе функции равны нулю вне указанных интервалов. Для получения пары случайных чисел  $y$  и  $x$  запишем выражения  $y = 2R_1$  и  $x = 4R_2$ , где  $R_1$  и  $R_2$  случайные числа из интервала  $(0,1)$ . Истинные случайные числа могут быть сгенерированы с использованием электронных приборов. Однако, так как имитационная модель реализуется на компьютере, такая генерация замедлит работу программы. Поэтому в имитационных моделях используются генераторы случайных чисел, основанных на арифметических вычислениях.

Такие числа не являются случайными. Их называют псевдослучайными, так как они могут быть заранее определены. Так, например, для получения псевдослучайных чисел может быть использована формула из теории целых чисел

$$u_n = (b \cdot u_{n+1} + c) \bmod(m); R_n = \frac{u_n}{m},$$

где  $u_0, b, c, m$  наперёд заданные параметры. Так, например, при  $u_0 = 11, b = 9, c = 5, m = 12$ , псевдослучайные числа  $R_1, R_2, R_3, R_4, R_5$  будут соответственно равны

$$u_1 = (9 \cdot 11 + 5) \bmod(12) = 8; R_1 = \frac{8}{12} = 0,6667;$$

$$u_2 = (9 \cdot 8 + 5) \bmod(12) = 5; R_2 = \frac{5}{12} = 0,4167;$$

$$u_3 = (9 \cdot 5 + 5) \bmod(12) = 2; R_3 = \frac{2}{12} = 0,1666;$$

$$u_4 = (9 \cdot 2 + 5) \bmod(12) = 11; R_4 = \frac{11}{12} = 0,9166;$$

$$u_5 = (9 \cdot 11 + 5) \bmod(12) = 8; R_5 = \frac{8}{12} = 0,6667.$$

На основе анализа результатов расчёта можно сделать вывод, что эта формула не может быть использована для генерации координат точек в прямоугольнике, так как числа начинают повторяться с периодом четыре. Такой период не позволяет обеспечить необходимую случайную выборку координат точек прямоугольника. Поэтому перед использованием выбранного генератора необходимо проводить статистические проверки по выявлению качества генератора, которое, в данном случае, зависит от значения параметров  $u_0, b, c$ .

Между тем следует заметить, в языках высокого уровня присутствуют функции для получения псевдослучайных чисел с большим периодом повторения. При этом разработчик программы должен обеспечить автоматический контроль за началом повторения псевдослучайных чисел, чтобы избежать получения неправильной имитации.

После выбора подходящего генератора псевдослучайных чисел выполняется имитация определения площади фигуры. Суть имитации заключается в следующем.

1. С помощью генератора определяется два числа  $R_1$  и  $R_2$ . Допустим, они соответственно равны 0,6675 и 0,2345. Тогда координаты точки будут равны:  $y = 2 \cdot 0,6675 = 1,335$  и  $x = 4 \cdot 0,2345 = 0,938$ .

2. Запишем (1) и (2) уравнение в таком виде:

$$y + 0,33x = 2, \tag{1.5}$$

$$y + 2x = 8 \tag{1.6}$$

и подставим туда вычисленные значения  $x$  и  $y$

$$1,335 + 0,333 \cdot 0,938 = 1,647,$$

$$1,335 + 2 \cdot 0,938 = 3,211.$$

Таким образом, так как левые части выражений (1.5) и (1.6) оказались меньше правых, делаем заключение, точка попала под ломанную кривую. В этом случае  $n = 1$  и  $m = 1$ . Если хотя бы в одном из выражений (1.5) и (1.6) ле-

вая часть окажется больше правой, делаем заключение, что точка не попала под ломанную кривую и присваиваем  $n = 2$  и  $m = 1$ . В противном случае присваиваем  $n = 2$  и  $m = 2$ .

Описанный процесс вычислений повторялся  $n = 5000$  раз при 10 прогонах. Он показал, что погрешность вычислений площади уменьшается с увеличением числа генерируемых точек и числа прогонов.

Результаты эксперимента представлены в таблице 1.

Таблица 1

Номер прогона	1	2	3	4	5	6	7	8	9	10
$m$	3248	3253	3256	3246	3249	3255	3247	3255	3239	3259
$F$	5.196	5,204	5,209	5,193	5,198	5,208	5,195	5,208	5,182	5,214
$S^2$	1,87E E-05	1,35 E-05	7,19 E-05	5,66 E-05	7,4 E-06	4,73 E-05	3,5 E-04	4,73 E-05	3,5 E-04	1,76 E-04

Ввиду того, что оценки площади имеют разброс, необходимо результаты экспериментов выразить в виде доверительного интервала точного значения.

В данном примере точная площадь

$$F = (2 + 0,8) \cdot 3,6/2 + (0 + 0,8) \cdot 0,4/2 = 5,2.$$

Доверительный интервал для  $F$  можно записать в виде

$$\bar{F} - \frac{s}{\sqrt{N}} t_{\alpha/2, N-1} \leq F \leq \bar{F} + \frac{s}{\sqrt{N}} t_{\alpha/2, N-1}, \quad (1.7)$$

где  $t_{\alpha/2, N-1}$  – коэффициент Стьюдента;  $\bar{F}$  – среднее значение площади;  $N$  – количество прогонов;  $\alpha$  – коэффициент значимости;  $s^2$  – дисперсия. Согласно данным таблицы 1 определяем  $\bar{F} = 5,201$ ,  $s^2 = 0,00009$ . При десяти прогонах и  $\alpha = 0,05$ ,  $t_{\alpha/2, N-1} = 2,23$ , доверительный интервал  $5,19 \geq F \leq 5,208$ .

На точность вычислений большое влияние оказывает величина площади прямоугольника, в который вписана геометрическая фигура, площадь которой надо определить. В представленном примере площадь прямоугольника равна  $4 \cdot 2 = 8$ . Если принять стороны прямоугольника 6 и 8, то при объеме вычислений, представленных выше, результаты вычислений имеют вид.

Таблица 2

Номер прогона	1	2	3	4	5	6	7	8	9	10
$m$	506	553	503	527	558	534	494	551	657	649
$F$	4.86	5,31	4,83	5,01	5,36	5,13	4,74	5,29	5,34	5,27
$S^2$	0,066	0,038	0,081	0,01	0,059	0,0002	0,138	0,03	0,054	0,024



Согласно данным таблицы 2, определяем  $\bar{F} = 5,11$ ,  $s^2 = 0,056$ . При десяти прогонах и  $\alpha = 0,05$ ,  $t_{\alpha/2, N-1} = 2,23$ , доверительный интервал  $4,94 \geq F \leq 5,28$ . Сопоставление результатов расчёта позволяет сказать, что при одинаковом объеме имитации во втором случае доверительный интервал в 19 раз больше, чем в первом случае. Это означает, что в первом случае расчёта результат расчёта ближе к истинному значению площади геометрической фигуры.

Оценивая с практической точки зрения полученный доверительный интервал, можно принять решение о необходимости увеличения или уменьшении объема выборки и количестве прогонов.

## 2. Модель системы массового обслуживания

В этом разделе рассматривается дискретно-событийная имитационная модель. Эта модель описывает поведение системы, изменяющееся в заданные моменты времени. Типичным примером являются системы массового обслуживания (СМО), в результате функционирования которых, образуются очереди из клиентов. Эти системы представляют собой совокупность двух объектов: сервис и клиент. Клиент поступает на обслуживание в сервис, а сервис его обслуживает.

В рассматриваемой модели происходит только два события: приход и уход клиента, так все статистические характеристики системы изменяются только в эти моменты времени. В другие моменты времени ничего в системе не происходит.

Таким образом, логику имитационной модели можно представить, рассматривая два события: приход и уход клиента.

В момент прихода клиента в модели необходимо выполнить следующие операции:

1. Сгенерировать и сохранить время прибытия клиента. Оно равняется текущее время плюс промежуток времени между приходами двух последних клиентов.

2. Если сервис свободен:

- 2.1. Начать обслуживание, поступившего клиента, изменить состояние сервиса на рабочее.

- 2.2. Сгенерировать и сохранить время ухода клиента.

3. Если сервис занят, поставить его в очередь и увеличить ее длину на 1.

В момент ухода клиента в модели необходимо выполнить следующие операции:

1. Объявить сервис свободным, если очередь пуста. Пересчитать статистические характеристики системы.

2. Если очередь не является пустой:

2.1. Начать обслуживание клиента из очереди. Уменьшить ее на 1 и пересчитать статистические характеристики системы.

2.2. Сгенерировать и сохранить время ухода клиента. Оно равно текущее время моделирования плюс время обслуживания клиента.

## 2.1 Механика моделирования СМО

Для иллюстрации процесса дискретно-событийного имитационного моделирования рассмотрим следующий пример.

В обслуживающую организацию поступают клиенты по экспоненциальному закону в среднем 6 человек в час. Обслуживание клиентов осуществляет один человек, который выполняет два вида работ. Время обслуживания клиента является дискретным. Первый вид работ занимает в среднем 15 минут, второй вид работ выполняется в среднем за 10 минут. Один из четырех клиентов заказывает первый вид работы.

Необходимо построить имитационную модель при времени её работы 50 минут и определить:

1. Среднюю занятость сервиса.
2. Среднюю длину очереди.
3. Среднее время ожидания клиентов в очереди.

Обозначим  $p$  и  $q$  как случайные величины времени прибытия и обслуживания клиентов, которые можно определить так:

$$p = -10 \ln(R) \text{ мин.} \quad 0 \leq R \leq 1 \quad (2.1)$$

$$q = 15 \text{ мин.} \quad 0 \leq R \leq 0,25 \quad \text{первая работа} \quad (2.2)$$

$$q = 10 \text{ мин.} \quad 0 \leq R \leq 0,75 \quad \text{вторая работа} \quad (2.3)$$

$R$  – равномерно распределённая величина в диапазоне 0–1.

Обозначим  $T$  – время наблюдения за системой, которое в начале положим равным 0, сервис объявляем свободным. С помощью датчика случайных чисел определяем значение  $R$ . Затем согласно (2.1), (2.2), (2.3) определяем случайное время прихода и обслуживания клиентов.

Для иллюстрации динамики изменения этих величин и образного восприятия сути дискретно-событийного моделирования время прихода и обслуживания клиентов изобразим на временной оси  $T$  на рисунке 2.1, рисунке 2.2. В этой модели (рис. 2.1) первый клиент приходит в систему через 0 минут, второй клиент через 28,3 минуты, третий клиент через 7,34 минуты, четвёртый клиент через 4,88 минуты и пятый клиент через 17,25 минуты. На рисунке 2.2

представлено время обслуживания клиентов, первый клиент обслуживается с 0 до 15 минут, второй клиент обслуживается с 28,3 до 43,3 минут, третий клиент обслуживается с 43,3 до 58,3 минуты. После их обслуживания они покидают систему, а их место занимают клиенты из очереди, если она не пуста. Так на рисунке 2.2 видно, что система находится в простое 15 до 28,3 минуты, так как первый клиент покидает систему в 15 минут, второй клиент поступает на обслуживание в 28,3 минуты.

Рассматривая последовательно приход и уход клиентов можно построить (рис. 2.3, рис. 2.4) графики количества клиентов в очереди и занятости системы в зависимости от модельного времени.

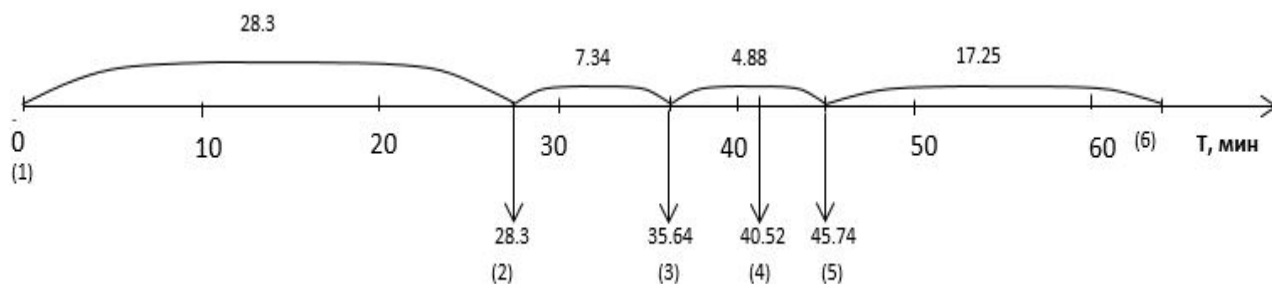


Рис. 2.1 – Приход клиентов:  
(1), (2), (3), (4), (5) – номер клиента

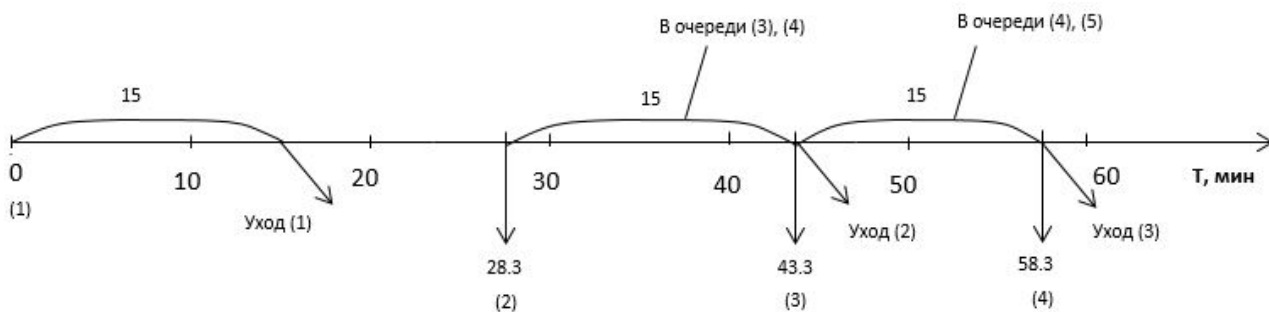


Рис. 2.2 – Обслуживание клиентов:  
(1), (2), (3), (4), (5) – номер клиента

На основе сопоставления этих рисунков построим зависимость (рис. 2.3) и (рис. 2.4) длины очереди и занятости сервиса от времени имитации.

И в заключении с помощью (рис. 2.3) и (рис. 2.4) определяем искомые показатели работы моделируемой системы:

- средняя длина очереди  $21,4/50 = 0,43$  человека;
- средняя занятость сервиса  $(15 + 21,7)/50 = 0,73$ ;
- среднее время ожидания клиентов, которые были в очереди  $21,4/3 = 7,3$  минут;
- среднее время ожидания всех клиентов, которые были в системе обслуживания в очереди  $21,4/5 = 4,28$  минут.

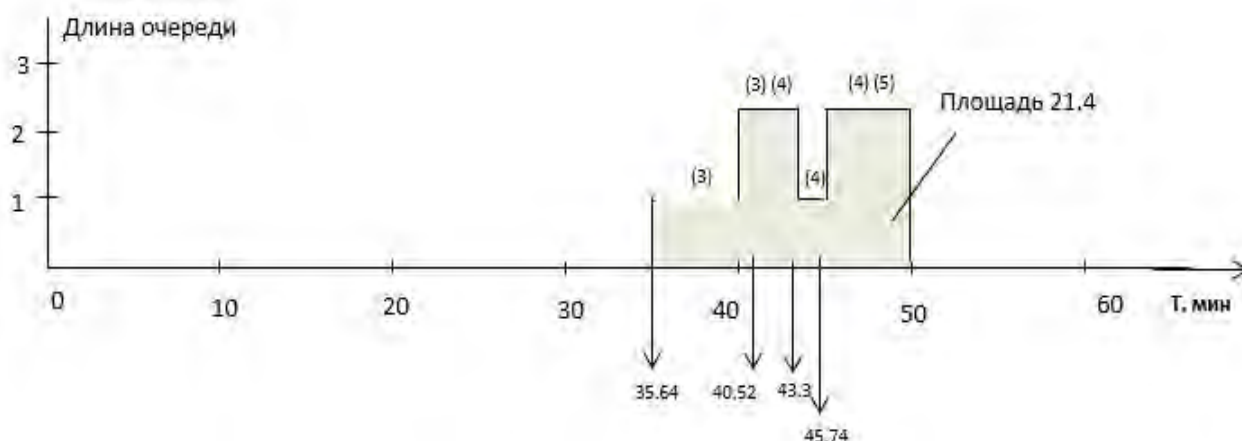


Рис. 2.3 – Длина очереди:  
(1), (2), (3), (4), (5) – номер клиента

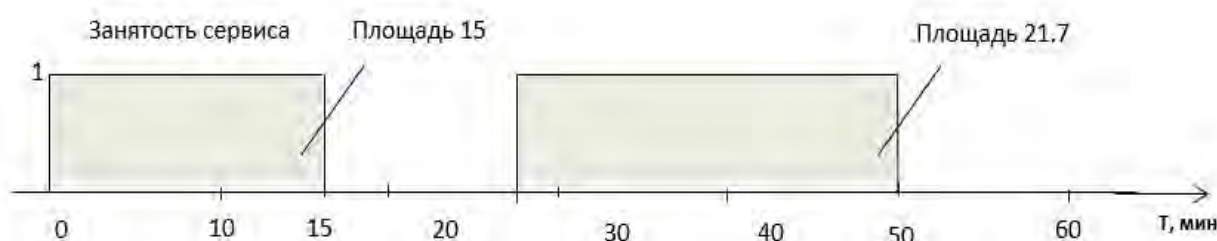


Рис. 2.4 – Занятость сервиса

Как видно из представленного примера, моделирование системы СМО связано с выполнением большого количества вычислений и поэтому предполагает написание приложения для реализации на компьютере. При этом очевидно, что написание и сопровождение такого рода программных продуктов требует значительных затрат.

В настоящее время имеются специальные программные продукты, решающие проблему по автоматизации создания приложений для имитационного моделирования систем. В данном методическом указании представлены примеры использования языка программирования GPSS.

## 2.2 Краткое описание языка GPSS

Язык GPSS используется при решении задач, в которых рассматриваются процессы с дискретными событиями. Такие процессы наблюдаются, например, при функционировании СМО.

В основе языка заложено понятие транзакта, который представляет собой формальный объект, перемещающийся по системе. Каждый транзакт обладает совокупностью параметров, принимающие различные значения в зависимости от места нахождения его в системе.

Язык GPSS – язык интерпретируемого типа. При выполнении программы происходит пошаговое выполнение операторов, называемые блоками. Комплекс программ реализующих функционирование блоков называется симулятором.

В GPSS присутствуют следующие объекты: устройство, память, очередь, ячейки, таблица.

Устройство имитирует единицу оборудования, который обрабатывает один транзакт.

Память имитирует единицу оборудования, в котором может обрабатываться несколько транзактов.

Очередь имитирует задержку транзакта перед устройством или памятью.

Таблица обеспечивает накопление статистики о параметре модели.

Ячейки используются для накопления и хранения входных и выходных параметров.

Для написания программы на GPSS необходимо знание формализмов языка и предметной логики моделируемой системы. Для иллюстрации работы блоков языка рассмотрим три примера моделирования с пошаговым комментированием программного кода и результатов решения.

### 2.3 Работа кассы с двумя кассирами

Необходимо построить имитационную модель работы кассы по продаже билетов при следующих параметрах работы. Посетители приходят в кассу через 10–30 секунд. Знакомятся с помещением 0–15 секунд и занимают очередь. В кассе работают два кассира. Каждый кассир затрачивает на обслуживание посетителя одинаковое время около 15–25 секунд. Касса работает 5 часов.

Код программы имеет следующий вид.

	<b>Код программы</b>	<b>Комментарий</b>
10	GENERATE 20, 10	;приход клиентов, генерирование транзактов (клиентов) в интервале 10–30 секунд.
20	TRANSFER .5, , PROM	;выбор кассира клиентом с вероятностью 50%.
30	ADVANCE 15, 15	;знакомство клиента с помещением перед кассой. Задержка транзакта (клиента) в помещении перед кассой на интервал времени 0–30 секунд.
40	QUEUE OCH	;транзакт (клиент) задерживается в очереди OCH. Поставить клиента в очередь.

50	SEIZE KASS	;занятие устройства (кассы) KASS, если оно свободно.
60	DEPART OCH	;выход транзакта (клиента) из очереди OCH.
70	ADVANCE 20,5	;задержка транзакта (клиента) в устройстве (кассе) KASS в интервал времени 15–25 с. Покупка билета.
80	RELEASE KASS	;освобождение устройства (кассы) KASS. Обслуживание клиента закончено.
90	TERMINATE	;уничтожение транзакта. Клиент покидает устройство (кассу).
100	PROM QUEUE OCH1	;поставить транзакта (клиента) в очередь OCH1
110	SEIZE KASS1	;занятие второго устройства (кассы) KASS1, если оно свободно. Обращение ко второму кассиру.
120	DEPART OCH1	;выход из очереди OCH1.
130	ADVANCE 20,5	;задержка транзакта (клиента) в устройстве (кассе) KASS1 в интервал времени 15–25 с. Покупка билета.
140	RELEASE KASS	;освобождение устройства (кассы) KASS1. Обслуживание клиента закончено.
150	TERMINATE	;уничтожение транзакта. Клиент покидает устройство (кассу).
160	GENERATE 18000	;время работы устройства (кассы) в секундах.
170	TERMINATE 1	;закрытие кассы. 1 – это уничтожение одного цикла работы кассы.
START 1		;старт программы 1 – один цикл работы устройства (кассы).

Если вместо 1 поставить, например, 3, то это означает моделирование работы кассы на протяжении 54000 секунд.

После запуска программы на выполнение получаем следующий результат.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	18000.000	17	2	0

START TIME – начало моделирования.  
 END TIME – конец моделирования.  
 BLOCKS – количество блоков в коде.  
 FACILITIES – количество устройств.

**STORAGES** – количество оборудований, обрабатывающих количество транзактов больше одного.

NAME	VALUE
KASS	10003.000
KASS1	10001.000
OCH	10002.000
OCH1	10000.000
PROM	10.000

**NAME** – названия устройств, очередей и начала следующей очереди.

**VALUE** – значение имён.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	909	0	0
	2	TRANSFER	909	0	0
	3	ADVANCE	443	1	0
	4	QUEUE	442	0	0
	5	SEIZE	442	0	0
	6	DEPART	442	0	0
	7	ADVANCE	442	1	0
	8	RELEASE	441	0	0
	9	TERMINATE	441	0	0
PROM	10	QUEUE	466	0	0
	11	SEIZE	466	0	0
	12	DEPART	466	0	0
	13	ADVANCE	466	0	0
	14	RELEASE	466	0	0
	15	TERMINATE	466	0	0
	16	GENERATE	1	0	0
	17	TERMINATE	1	0	0

**LABEL** – присвоенное имя блока.

**LOC** – номер блока.

**BLOCK TYPE** – названия блока

**ENTRY COUNT** – количество транзактов, побывавших в блоке.

**CURRENT COUNT** – количество транзактов на момент окончания моделирования.

**RETRY** – количество транзактов, ожидающих выполнения специальных условий.

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASS1	466	0.517	19.952	1	0	0	0	0	0
KASS	442	0.492	20.053	1	909	0	0	0	0

**FACILITY** – имя устройства, присвоенное в программе.

**ENTRIES** – количество раз было занято устройство.

**UTIL.** – средняя загрузка устройства.

**AVE. TIME** – среднее время транзакта в устройстве.

AVAIL. – доступность устройства на момент окончания моделирования.  
 OWNER – номер транзакта в устройстве.  
 PEND – количество транзактов в очереди при блоке PREEMPT.  
 INTER – количество транзактов в очереди после прерывания.  
 RETRY – количество транзактов в очереди в зависимости от состояния устройства.  
 DELAY – количество транзактов в очереди при блоке SEIZE и PREEMPT.

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
OCH1	2	0	466	339	0.048	1.855	6.808	0
OCH	3	0	442	287	0.109	4.433	12.642	0

QUEUE – название очереди.  
 MAX – максимальная длина очереди.  
 CONT – длина очереди на момент окончания моделирования.  
 ENTRY – общее количество входов в очередь.  
 ENTRY(0) – количество нулевых входов.  
 AVE.CONT. – средняя длина очереди.  
 AVE.TIME – среднее время нахождения в очереди.

Изменяя входные данные, можно получить различные статистические характеристики моделируемой системы и определить нужный режим работы предприятия. На практике присутствуют предприятия с различными режимами и графиками работы. Вместе с тем логика функционирования предприятий одинакова. При этом при моделировании системы необходимо учитывать уникальные особенности конкретного предприятия. С помощью языка GPSS эти особенности можно смоделировать, используя различные сочетания блоков. В следующем примере представим модель предприятия с двумя видами работ и одним работником (в предыдущем примере был один вид работ и два работника).

## 2.4 Работа парикмахерской с одним мастером

Необходимо построить имитационную модель работы парикмахерской по обслуживанию клиентов при следующих параметрах работы. В обслуживающую организацию поступают клиенты по экспоненциальному закону в среднем 2 человека в час (математическое ожидание 30 минут). Обслуживание клиентов осуществляет один человек, который выполняет два вида работ. Время обслуживания клиента является дискретным. Первый вид работы занимает в среднем



20 минут, второй вид работы выполняется в среднем за 30 минут. Один из четырех клиентов заказывает первый вид работы.

Необходимо построить имитационную модель при времени её работы 60 минут и определить статистические характеристики моделируемой системы.

Код программы имеет следующий вид.

Код программы	Комментарий
10 GENERATE (EXPONENTIAL(1,0,30))	;Приход клиентов. Генерирование времени прихода по экспоненциальному закону.
20 ADVANCE 5,0	; Знакомство с парикмахерской на интервале времени 0–5 минут.
30 QUEUE OCH	;Встать в очередь
40 SEIZE MAS	;Обращение к мастеру. Занятие устройства MAS, если оно свободно
50 DEPART OCH	;Выход из очереди
60 TRANSFER .75,,prom	;Выбор с вероятностью 0.75, второй вид работ
70 ADVANCE 20,0	;Выполнение первого вида работ интервал времени 0–20 минут.
80 RELEASE MAS	;Освобождение устройства (мастера) MAS.
90 TERMINATE	;Уничтожение транзакта.
100 PROM ADVANCE 30,0	;Выполнение второго вида работ в интервал времени 0–30 минут.
110 RELEASE MAS	;Освобождение устройства (мастера) MAS.
120 TERMINATE	;Уничтожение транзакта.
130 GENERATE 480	;Время работы парикмахерской в Минутах.
140 TERMINATE 1	;Закрытие парикмахерской.
START 1	;Старт программы.

После запуска программы на выполнение получаем следующий результат.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	14	1	0

NAME	VALUE
KASS	10001.000
OCH	10000.000
PROM	10.000

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
-------	-----	-------	------	-------	-------	---------	-------	-------

	1	GENERATE	15	0	0
	2	ADVANCE	15	0	0
	3	QUEUE	15	0	0
	4	SEIZE	15	0	0
	5	DEPART	15	0	0
	6	TRANSFER	15	0	0
	7	ADVANCE	3	0	0
	8	RELEASE	3	0	0
	9	TERMINATE	3	0	0
PROM	10	ADVANCE	12	1	0
	11	RELEASE	11	0	0
	12	TERMINATE	11	0	0
	13	GENERATE	1	0	0
	14	TERMINATE	1	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
KASS	15	0.821	26.257	1	16	0	0	0
0								
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	
RETRY								
OCH	3	0	15	4	0.781	25.003	34.095	
0								

В рассмотренном примере для моделирования нескольких видов работ используется TRANSFER, который моделирует выполнение двух работ на сервисе.

В следующем примере рассмотрим работу предприятия, в котором возможен отказ клиента от обслуживания.

## 2.5 Работа парикмахерской с несколькими мастерами при отказе от обслуживания

Необходимо построить имитационную модель работы парикмахерской по обслуживанию клиентов при следующих параметрах работы. В обслуживающую организацию поступают клиенты по экспоненциальному закону в среднем 2 человека в час (математическое ожидание 30 минут). Обслуживание клиентов осуществляют четыре мастера. Время обслуживания клиента является дискретным и составляет от 14 до 19 минут. Клиент отказывается от обслуживания, если в очереди находится более двух человек

Необходимо построить имитационную модель при времени ее работы 480 минут и определить статистические характеристики системы.

Код программы имеет следующий вид:

```

10 KRES    STORAGE      4
20        GENERATE     TWT
30        TEST L       Q1, 3, ОТКАЗ
40        QUEUE       1
50        ENTER       KRES

```

```

60          DEPART          1
70          ADVANCE        19,5
80          LEAVE          KRES
90          TERMINATE
100 ОТКАЗ  TERMINATE
110          GENERATE       480
120          TERMINATE      1
START 1

```

Представленная программа отличается от предыдущих следующим:

- используется объект память KRES, которая описывается картой STORAGE. Карта не является объектом, так как транзакты в неё не входят. Она просто описывает объекты. В данном примере в памяти находятся 4 мастера по обслуживанию клиентов:

- используется блок ENTER, в котором проверяется достаточно ли места для вхождения транзакта в память KRES;

- используется блок LEAVE, в котором освобождается место памяти KRES;

- используется блок TEST, в котором проверяется условие отказа от обслуживания. При этом символом L обозначается условие меньше, символом Q1 обозначается текущая длина очереди номер 1, цифра 3 обозначает критическую длину очереди, символ ОТКАЗ предназначен для удаления транзакта из системы.

После запуска программы на выполнение получаем следующий результат.

```

START TIME          END TIME  BLOCKS  FACILITIES  STORAGEES
          0.000          480.000    11         0         1

          NAME          VALUE
          KRES          10000.000
          ОТКАЗ          9.000

LABEL          LOC  BLOCK TYPE          ENTRY COUNT  CURRENT  COUNT  RETRY
          1  GENERATE          18         0         0
          2  TEST          18         0         0
          3  QUEUE          18         0         0
          4  ENTER          18         0         0
          5  DEPART          18         0         0
          6  ADVANCE          18         0         0
          7  LEAVE          18         0         0
          8  TERMINATE          18         0         0
ОТКАЗ          9  TERMINATE          0         0         0
          10 GENERATE          1         0         0
          11 TERMINATE          1         0         0

QUEUE          MAX CONT.  ENTRY  ENTRY (0)  AVE. CONT.  AVE. TIME  AVE. (-0)  RETRY
          1          1      0      18      18      0.000      0.000      0.000  0

STORAGE          CAP.  REM.  MIN.  MAX.  ENTRIES  AVL.  AVE. C.  UTIL.  RETRY  DELAY
KRES          4      4      0      3      18      1      0.720  0.180  0      0

```

STORAGE	– имя объекта память.
CAP.	– ёмкость памяти.
REM.	– количество свободных ячеек памяти на момент окончания моделирования.
MIN. MAX.	– минимальное и максимальное количество занятых ячеек в течении моделирования.
ENTRIES	– общее количество входов.
AVL.	– занятость памяти на момент окончания моделирования.
AVE.C.	– среднее количество занятых ячеек памяти.
UTIL.	– средняя загрузка памяти в течении моделирования.

Представленные примеры иллюстрируют принципиальные моменты решения задач по имитационному моделированию. Для более глубокого изучения и решения конкретных задач предлагается использовать уже имеющиеся решения [1–4].

### 3. Варианты контрольных заданий

#### 3.1 Задание 1

В соответствии с вариантом задания предлагается:

1. Выбрать из таблицы 5 сетевую функцию.
2. Построить полином первой степени.
3. Определить с помощью имитационного моделирования площадь геометрической фигуры, ограниченной полученным полиномом и осью  $x$ .

Таблица 5

№ задания	Сетевая функция						
	x	1	2	3	4	5	6
1	x	1	2	3	4	5	6
	y	3	8	11	18	27	38
2	x	1	2	3	4	5	6
	y	2	8	18	32	50	72
3	x	1	2	3	4	5	6
	y	4	10	20	34	52	74
4	x	1	2	3	4	5	6
	y	6	9	14	21	30	41
5	x	1	2	3	4	5	6
	y	9	12	17	24	33	44
6	x	1	2	3	4	5	6
	y	11	14	19	26	35	46
7	x	1	2	3	4	5	6
	y	13	16	21	28	37	48

Окончание табл. 5

8	x	1	2	3	4	5	6
	y	10	16	26	40	58	80
9	x	1	2	3	4	5	6
	y	12	18	28	42	60	82
10	x	1	2	3	4	5	6
	y	15	21	31	45	63	85
11	x	1	2	3	4	5	6
	y	2	16	54	128	250	432
12	x	1	2	3	4	5	6
	y	3	17	55	129	251	433
13	x	1	2	3	4	5	6
	y	7	21	59	133	255	437
14	x	1	2	3	4	5	6
	y	20	10	6.6	5	4	3.3
15	x	1	2	3	4	5	6
	y	21	11	7.6	6	5	4.3
16	x	1	2	3	4	5	6
	y	24	14	10.6	9	8	7.3
17	x	1	2	3	4	5	6
	y	40	20	13.3	10	8	6.6
18	x	1	2	3	4	5	6
	y	42	22	15.2	12	10	8.6
19	x	1	2	3	4	5	6
	y	47	27	20.2	17	15	13.6
20	x	1	2	3	4	5	6
	y	100	25	11.1	6.2	4	2.8

### 3.2 Задание 2

В супермаркете имеется  $M$  касс. Посетители занимают места в очереди кассы. Как правило, очереди к различным кассам примерно одинаковые. Было замечено, что при длине очереди больше  $N$  покупатель уходит из супермаркета без покупки. Время прихода покупателей и время их обслуживания имеет экспоненциальную зависимость. Среднее время прихода равно  $T$ , а среднее время обслуживания равно  $Z$ . Для привлечения покупателей перед супермаркетом построена парковка на  $X$  машин. В течение суток супермаркет работает  $S$  часов.

В соответствии с вариантом задания предлагается построить имитационную модель работы супермаркета и определить статистические характеристики системы согласно исходным данным, представленным в таблице 6.

Таблица 6

№ задания	М Число касс (штук)	Н Длина очереди (человек)	Т Время прихода покупателя (минута)	Z Время обслуживания покупателя (минута)	Х Мест на парковке (штук)	S Время работы супермаркета (минута)
1	3	5	20	10	4	480
2	6	7	25	15	7	960
3	5	10	16	26	40	580
4	7	12	18	8	25	60
5	6	15	21	31	45	180
6	1	2	16	5	28	250
7	2	3	17	15	12	240
8	3	7	21	9	13	300
9	4	2	10	6	5	960
10	5	11	11	7	6	120
11	1	24	14	10	9	720
12	3	4	20	13.	10	660
13	5	2	22	15	12	580
14	8	7	27	20	17	120
15	3	7	21	9	13	300
16	4	2	10	6	8	960
17	5	9	11	7	6	120
18	1	6	14	10	5	720
19	3	4	20	13.	10	660
20	5	7	23	14	3	580
21	8	5	20	10	16	180

### Библиографический список

1. Таха, Х. А. Введение в исследование операций / Х. А. Таха. – В 2-х кн. – М. : Изд. дом, 2005.
2. Карпов, Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic / Ю. Карпов. – СПб,-БХВ-Петербург, 2005.
3. НОУ «ИНТУИТ»: <http://www.intuit.ru/studies/coursees>
4. Имитационное моделирование: <http://www.gpss.ru/index-h.html>