



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Программное обеспечение ЭВМ и информационные технологии»

---

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ по курсу «Моделирование»

«Моделирование работы информационного центра»

Студент:	<u>ИУ7-73Б</u> (группа)	_____ (подпись, дата)	<u>М. Д. Маслова</u> (И. О. Фамилия)
Преподаватель:		_____ (подпись, дата)	<u>И. В. Рудаков</u> (И. О. Фамилия)

2022 г.

## СОДЕРЖАНИЕ

<b>1</b>	<b>Задание . . . . .</b>	<b>3</b>
<b>2</b>	<b>Теоретическая часть . . . . .</b>	<b>4</b>
2.1	Схемы модели . . . . .	4
2.2	Переменные и уравнение имитационной модели . . . . .	5
<b>3</b>	<b>Практическая часть . . . . .</b>	<b>6</b>
3.1	Текст программы . . . . .	6
3.2	Полученный результат . . . . .	8

## 1 Задание

В информационный центр приходят клиенты через интервалы времени  $10 \pm 2$  минуты. Если все три имеющихся оператора заняты, клиенту отказывают в обслуживании. Операторы имеют разную производительность и могут обеспечивать обслуживание среднее запросы за  $20 \pm 5$ ,  $40 \pm 10$ ,  $40 \pm 20$  минут. Клиенты стремятся занять свободного оператора с максимальной производительностью.

Полученные запросы сдаются в приемные накопители, откуда они выбираются для обработки. На первый компьютер — запросы от первого и второго операторов, на второй компьютер — от третьего оператора. Время обработки на первом и втором компьютере равны соответственно 15 и 30 минутам.

Смоделировать процесс обработки 300 запросов. Определить вероятность отказа.

## 2 Теоретическая часть

### 2.1 Схемы модели

На рисунке 2.1 представлена структурная схема модели.

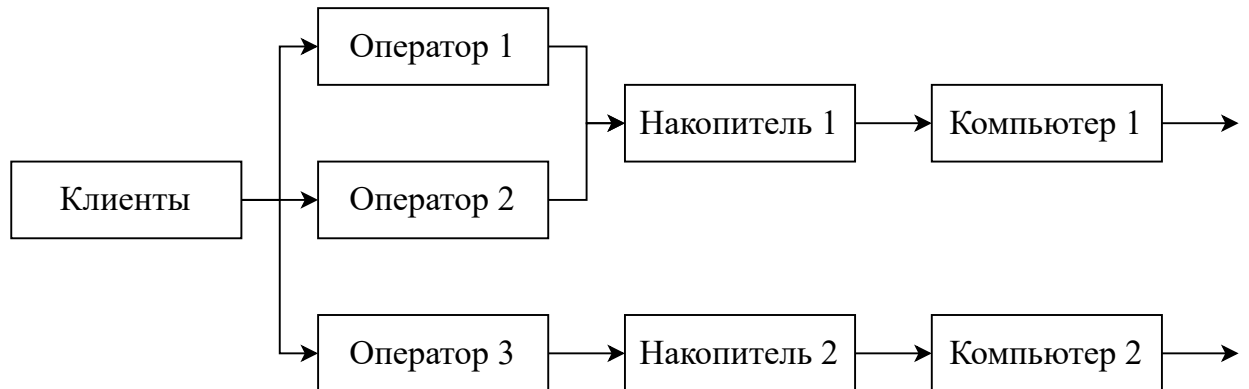


Рисунок 2.1 – Структурная схема модели

В процессе взаимодействия клиентов с информационным центром возможно два режима работы:

- режим нормального обслуживания, когда клиент выбирает одного из свободных операторов, отдавая предпочтение тому, у кого максимальная производительность;
- режим отказа клиенту в обслуживании, когда все операторы заняты.

На рисунке 2.2 представлена схема модели в терминах систем массового обслуживания.

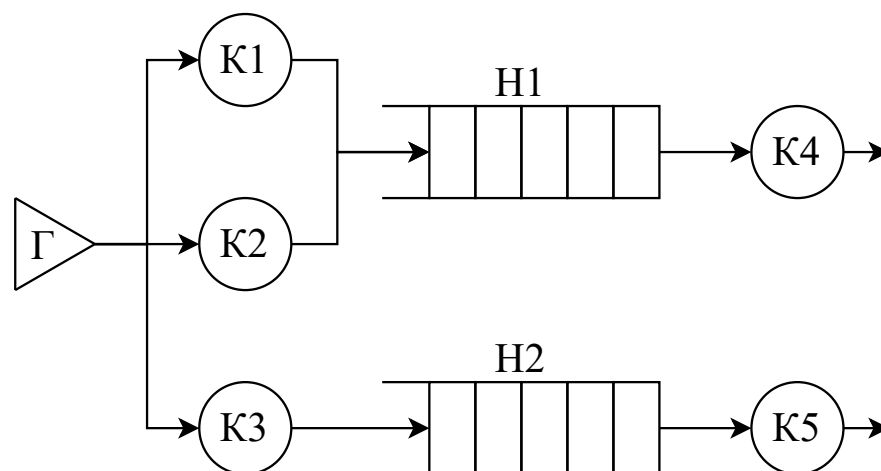


Рисунок 2.2 – Схема модели в терминах СМО

## 2.2 Переменные и уравнение имитационной модели

Эндогенные переменные:

- время обработки задания  $i$ -ым оператором;
- время решения задания на  $j$ -ом компьютере.

Экзогенные переменные:

- $n_0$  — число обслуженных клиентов;
- $n_1$  — число клиентов, получивших отказ.

Вероятность отказа рассчитывается по формуле 2.1, которая описывает уравнение модели:

$$P_{\text{отказа}} = \frac{n_0}{n_0 + n_1}. \quad (2.1)$$

## 3 Практическая часть

### 3.1 Текст программы

На листинге 3.1-3.2 представлены реализации генератора и канала обслуживания соответственно.

Листинг 3.1 — Реализация генератора

```
1  class Generator:
2      def __init__(self, distribution, receivers):
3          self.distribution = distribution
4          self.receivers = receivers
5          self.nextEvent = Event(-1, self)
6
7      def GenerateNextEvent(self, curTime):
8          self.nextEvent.Time = curTime + self.distribution.Generate()
9
10     def TransmitRequest(self):
11         for receiver in self.receivers:
12             if receiver.TakeRequest(self.nextEvent.time):
13                 return True
14         return False
```

Листинг 3.2 — Реализация канала обслуживания

```
1  class Processor(Generator):
2      def __init__(self, generator: Generator, memory: Memory):
3          super().__init__(generator.distribution, generator.receivers)
4          self.nextEvent.eventBlock = self
5          self.memory = memory
6          self.aviable = True
7
8      def TakeRequest(self, curTime) -> bool:
9          if self.aviable:
10             self.aviable = False
11             self.GenerateNextEvent(curTime)
12             return True
13         return self.memory.InsertRequest()
14
15     def EndProcess(self, curTime):
16         self.TransmitRequest()
17         if not self.memory.IsEmpty():
18             self.memory.RemoveRequest()
19             self.GenerateNextEvent(curTime)
20         else:
21             self.aviable = True
22             self.NextEvent.Time = -1
```

На листинге 3.3 реализация моделирования работы информационного центра.

Листинг 3.3 — Реализация моделирования работы информационного центра

```
1  class EventModel:
2      def __init__(self, generator: Generator, operators: list[Processor]
3          , computers: list[Processor], requestsNum=1000):
4          self.generator = generator
5          self.operators = operators
6          self.computers = computers
7          self.blocks = [self.generator] + self.operators + self.computers
8          self.requestsNum = requestsNum
9
10     def run(self):
11         self.generator.GenerateNextEvent(0)
12         events = [block.NextEvent for block in self.blocks]
13
14         generatedRequests = 1
15         denials = 0
16
17         while generatedRequests < self.requestsNum:
18             curTime = events[0].Time
19             for event in events[1:]:
20                 if not event.Time < 0 and event.Time < curTime:
21                     curTime = event.Time
22
23             for block in self.blocks:
24                 if abs(block.NextEvent.Time - curTime) < EPS:
25                     if not isinstance(block.NextEvent.EventBlock,
26                         ↪ Processor):
27                         generatedRequests += 1
28                         if not block.TransmitRequest():
29                             denials += 1
30                             block.GenerateNextEvent(curTime)
31                     else:
32                         block.EndProcess(curTime)
33
34         return denials / generatedRequests
```

## 3.2 Полученный результат

На рисунке 3.1 приведен примеры работы программы на значениях, данных в условии.

Маслова Марина ИУ7-73Б Лабораторная работа №5

Интервал прихода	Время обслуживания	Время обработки
<b>Клиенты</b> 10 - + ± 2 - + минут	<b>Оператор 1</b> 20 - + ± 5 - + минут	<b>Компьютер 1</b> 15 - + минут
	<b>Оператор 2</b> 40 - + ± 10 - + минут	<b>Компьютер 2</b> 30 - + минут
	<b>Оператор 3</b> 40 - + ± 20 - + минут	
<b>Количество заявок</b> 300 - +	<b>Смоделировать</b>	<b>Вероятность отказа</b> 0.213 - +

Автор: Маслова Марина ИУ7-73Б

Рисунок 3.1 – Пример работы программы