



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»

---

## Отчёт по лабораторной работе №5 по курсу «Моделирование»

Тема Моделирование работы информационного центра

---

Студент Калашков П. А.

---

Группа ИУ7-76Б

---

Оценка (баллы)

---

Преподаватели Рудаков И. В.

---

Целью данной работы является разработка программы с графическим интерфейсом для моделирования процесса обработки 300 запросов клиентов информационным центром и определения вероятности отказа клиенту в обслуживании. Информационный центр работает следующим образом:

1. Клиенты приходят через интервал времени, равный  $10 \pm 2$  мин.
2. Если все три имеющихся оператора заняты, клиенту отказывают в обслуживании. Операторы имеют разную производительность и могут обеспечивать обслуживание среднего запроса пользователя за  $20 \pm 5$  мин.,  $40 \pm 10$  мин. и  $40 \pm 20$  мин. соответственно. Клиенты стремятся занять свободного оператора с максимальной производительностью.
3. Полученные запросы сдаются в приемный накопитель, из которого они выбираются для обработки. На первый компьютер выбираются запросы от первого и второго операторов, на второй компьютер — от третьего оператора. Время обработки на первом и втором компьютерах равны соответственно 15 мин. и 30 мин.

В процессе взаимодействия клиентов и информационного центра предусмотреть: режим нормального обслуживания, когда клиент выбирает одного из свободных операторов с максимальной производительностью, и режим отказа.

## Моделирование функционирования системы

При моделировании функционирования системы эндогенными переменными являются:

- время обслуживания клиента  $i$ -ым оператором, где  $i = \overline{1, 3}$ ;
- время обработки запроса на  $j$ -ом компьютере, где  $j = \overline{1, 2}$ .

Экзогенными переменными являются:

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

Уравнение модели имеет следующий вид:

$$P_{\text{отказа}} = \frac{n_1}{n_0 + n_1} \quad (1)$$

## Схемы модели

На рисунке 1 показана структурная схема модели.

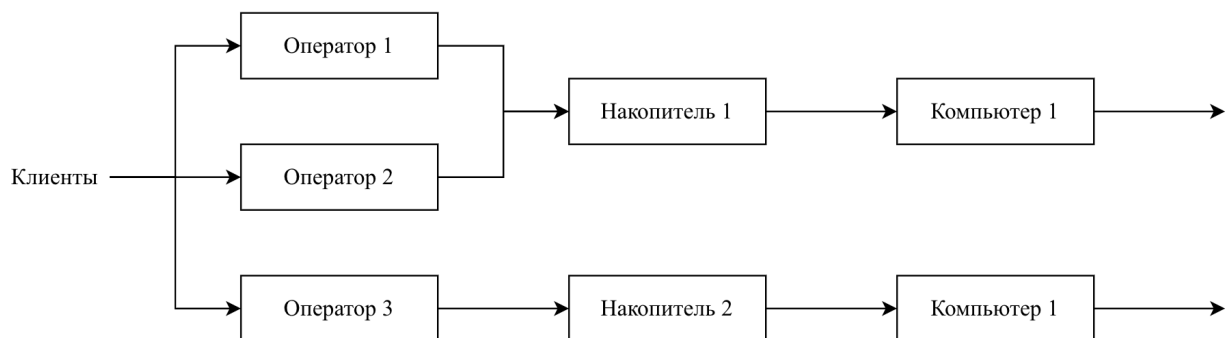


Рисунок 1 – Структурная схема модели информационного центра

На рисунке 2 представлена схема модели в терминах СМО.

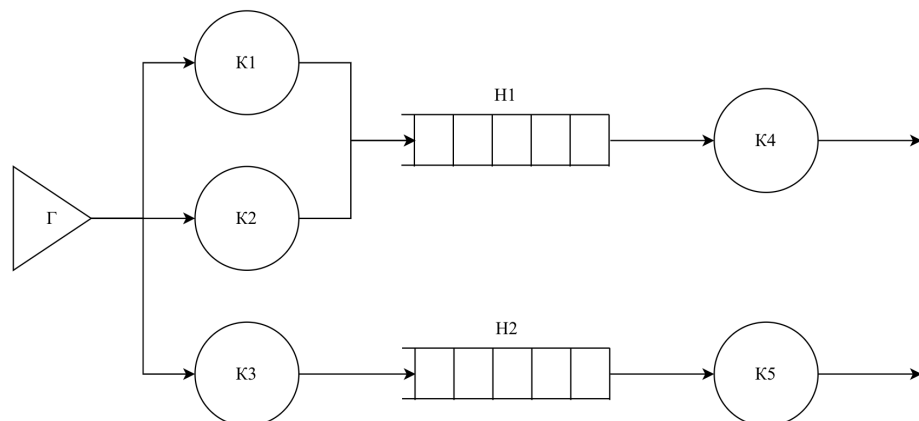


Рисунок 2 – Схема модели в терминах СМО

# Результаты работы

## Детали реализации

В листинге 1 представлена реализация работы генератора клиентов, а в листингах 2–3 — реализация работы оператора. Реализации работы компьютера и самого информационного центра представлены в листингах 4 и ?? соответственно.

### Листинг 1 – Реализация работы генератора клиентов

```
1 class ClientGenerator:
2     def __init__(self, time_value, time_limit, operators, number):
3         self.generator = TimeGenerator(time_value - time_limit,
4                                         time_value + time_limit)
5         self.operators = sorted(operators, key= lambda operator:
6                                   operator.max_time)
7         self.time_next = 0
8         self.number = number
9
10    def generate_client(self, time_prev):
11        self.time_next = time_prev + self.generator.get_interval()
12
13    def choose_operator(self):
14        for operator in self.operators:
15            if operator.is_free():
16                return operator
17        return None
```

### Листинг 2 – Реализация работы оператора (часть 1)

```
1 class Operator:
2     def __init__(self, time_value, time_limit, computer):
3         self.time_generator = TimeGenerator(time_value -
4                                             time_limit, time_value + time_limit)
5         self.computer = computer
6         self.max_time = time_value + time_limit
7         self.time_next = 0
8         self.free = True
9
10    def generate_time(self, prev_time):
11        self.time_next = prev_time +
12                        self.time_generator.get_interval()
13
14    def is_free(self):
15        return self.free
16
17    def set_free(self):
18        self.free = True
```

### Листинг 3 – Реализация работы оператора (часть 2)

```
1  def set_busy(self):
2      self.free = False
3
4  def get_computer(self):
5      return self.computer
```

### Листинг 4 – Реализация работы компьютера

```
1  class Computer:
2      def __init__(self, time_value, time_limit):
3          self.time_generator = TimeGenerator(time_value -
4              time_limit, time_value + time_limit)
5          self.queue = []
6          self.time_next = 0
7          self.free = True
8
9      def generate_time(self, prev_time):
10         self.time_next = prev_time +
11             self.time_generator.get_interval()
12
13     def is_free(self):
14         return self.free
15
16     def set_free(self):
17         self.free = True
18
19     def set_busy(self):
20         self.free = False
21
22     def is_empty(self):
23         if self.queue:
24             return False
25         return True
26
27     def add_request(self):
28         self.queue.append(request)
29
30     def pop_request(self):
31         self.queue.pop(0)
```

Листинг 5 – Реализация работы информационного центра (часть 1)

```
1 class Center:
2     def __init__(self, client_generator):
3         self.client_generator = client_generator
4
5     def service_clients(self):
6         failures = 0
7         self.client_generator.generate_client(0)
8         generated_clients = 1
9         events = [Event(self.client_generator,
10                        self.client_generator.time_next)]
11
12     while generated_clients < self.client_generator.number:
13         events = sort_events(events)
14         event = events.pop(0)
15
16         if isinstance(event.creator, ClientGenerator):
17             operator = self.client_generator.choose_operator()
18             if operator is None:
19                 failures += 1
20             else:
21                 operator.set_busy()
22                 operator.generate_time(event.time)
23                 events.append(Event(operator,
24                                    operator.time_next))
25                 self.client_generator.generate_client(event.time)
26                 generated_clients += 1
27                 events.append(Event(self.client_generator,
28                                    self.client_generator.time_next))
29
30         elif isinstance(event.creator, Operator):
31             operator = event.creator
32             operator.set_free()
33             computer = operator.get_computer()
34             computer.add_request()
35             if computer.is_free() and not computer.is_empty():
36                 computer.pop_request()
37                 computer.set_busy()
38                 computer.generate_time(event.time)
39                 events.append(Event(computer,
40                                    computer.time_next))
```

## Листинг 6 – Реализация работы информационного центра (часть 2)

```
1      elif isinstance(event.creator, Computer):
2          computer = event.creator
3          computer.set_free()
4          if not computer.is_empty():
5              computer.pop_request()
6              computer.set_busy()
7              computer.generate_time(event.time)
8              events.append(Event(computer,
9                                  computer.time_next))
10     return failures
```

## Примеры работы

На рисунке 3 представлен пример работы разработанной программы для описанного информационного центра с обработкой 300 заявок.

Лабораторная работа №5 по курсу "Моделирование", тема:...

Подробнее о программе

Клиенты

Число клиентов: 300

Интервал прибытия (мин.): 10 ± 2

Информационный центр

Время обслуживания (мин.)

Первым оператором 20 ± 5

Вторым оператором 40 ± 10

Третьим оператором 40 ± 20

Время обработки (мин.)

Первым компьютером 15

Вторым компьютером 30

Промоделировать

Результат

Число обслуженных клиентов: 238

Число отказов: 62

Вероятность отказа: 0.20667

Рисунок 3 – Пример работы программы, обработка 300 заявок



## Вывод

В ходе выполнения лабораторной работы была реализована программа с графическим интерфейсом для моделирования процесса обработки 300 запросов клиентов информационным центром и определения вероятности отказа клиенту в обслуживании. Вероятность отказа равна примерно 0.2.