



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 (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

ОТЧЕТ

по лабораторной работе № 5

по курсу «Моделирование»

на тему: «Моделирование работы информационного центра»

Студент ИУ7-73Б
(Группа)

(Подпись, дата)

Р. Р. Хамзина
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

И. В. Рудаков
(И. О. Фамилия)

2022 г.

СОДЕРЖАНИЕ

1	Задание	3
1.1	Моделирование функционирования системы	3
1.2	Структурная схема модели	4
1.3	Схема модели в терминах СМО	4
2	Реализация	5
2.1	Детали реализации	5
2.2	Полученный результат	9

1 Задание

Реализовать программу 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 ± 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 ± 5 мин., 40 ± 10 мин. и 40 ± 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 возможны: режим нормального обслуживания, когда клиент выбирает одного из свободных операторов с максимальной производительностью, и режим отказа.

1.1 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:

$$P_{\text{отказа}} = \frac{n_1}{n_0 + n_1} \quad (1.1)$$

1.2 Структурная схема модели

На рисунке 1.1 показана структурная схема модели.

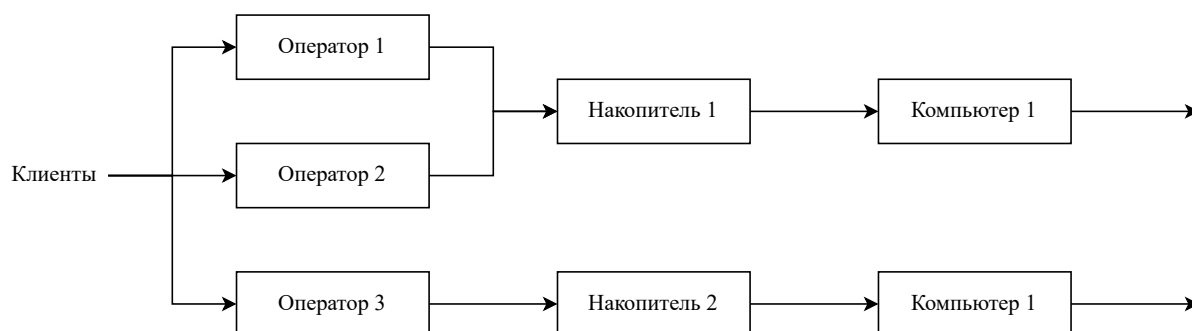


Рисунок 1.1 – Структурная схема модели

1.3 Схема модели в терминах СМО

На рисунке 1.2 представлена схема модели в терминах СМО.

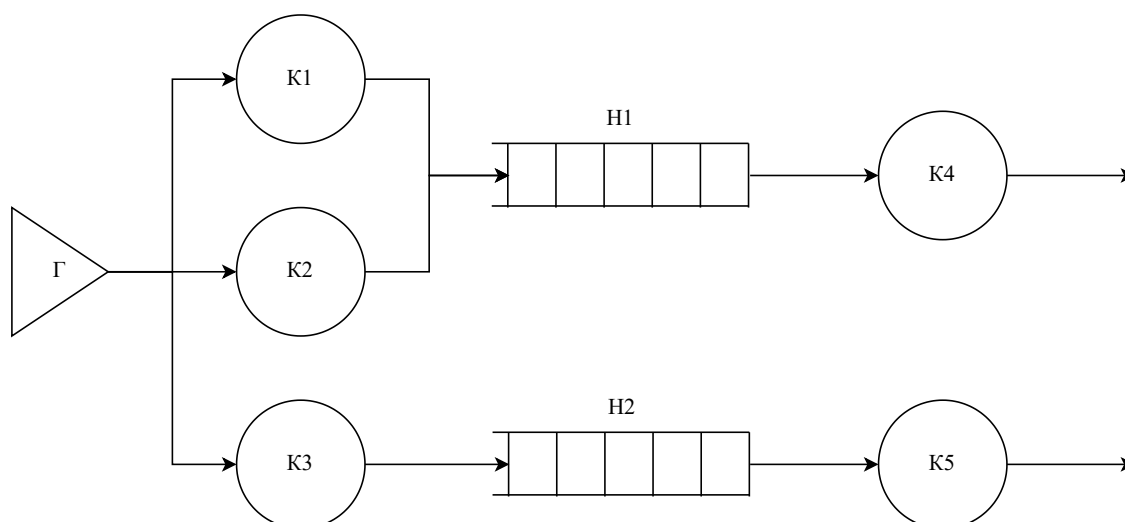


Рисунок 1.2 – Схема модели в терминах СМО

2 Реализация

2.1 Детали реализации

На листинге 2.1 показана реализация класса генератора клиентов.

Листинг 2.1 – Моделирование работы генератора клиентов

```
1 class ClientGenerator:
2     def __init__(self, time_value, time_limit, operators,
3         number):
4         self.time_generator =
5             TimeGenerator(time_value - time_limit,
6                 time_value + time_limit)
7         self.operators = self.__sort_operators(operators)
8         self.time_next = 0
9         self.number = number
10
11     def generate_client(self, time_prev):
12         self.time_next = time_prev + \
13             self.time_generator.get_time_interval()
14
15     def choose_operator(self):
16         for operator in self.operators:
17             if operator.is_free():
18                 return operator
19
20         return None
21
22     def __sort_operators(self, operators):
23         return sorted(operators, key= lambda operator:
24             operator.max_time)
```

На листинге 2.2 представлена реализация класса оператора.

Листинг 2.2 – Моделирование работы оператора

```
1 class Operator:
2     def __init__(self, time_value, time_limit, computer):
3         self.time_generator =
4             TimeGenerator(time_value - time_limit,
5                             time_value + time_limit)
6         self.computer = computer
7         self.max_time = time_value + time_limit
8         self.time_next = 0
9         self.free = True
10
11     def generate_time(self, prev_time):
12         self.time_next = prev_time + \
13             self.time_generator.get_time_interval()
14
15     def is_free(self):
16         return self.free
17
18     def set_free(self):
19         self.free = True
20
21     def set_busy(self):
22         self.free = False
23
24     def get_computer(self):
25         return self.computer
```

На листинге 2.3 показана реализация класса компьютера.

Листинг 2.3 – Моделирование работы компьютера

```
1 class Computer:
2     def __init__(self, time_value, time_limit):
3         self.time_generator =
4             TimeGenerator(time_value - time_limit,
5                             time_value + time_limit)
6         self.queue = []
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_time_interval()
13
14     def is_free(self):
15         return self.free
16
17     def set_free(self):
18         self.free = True
19
20     def set_busy(self):
21         self.free = False
22
23     def queue_empty(self):
24         if self.queue:
25             return False
26         return True
27
28     def append_request(self):
29         self.queue.append(Constants.request)
30
31     def pop_request(self):
32         self.queue.pop(0)

```

На листинге 2.4 представлена реализация класса информационного центра.

Листинг 2.4 – Моделирование работы информационного центра

```

1 class Center:
2     def __init__(self, client_generator):
3         self.client_generator = client_generator
4
5     def service_clients(self):
6         failures_number = 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 =
18             self.client_generator.choose_operator()
19
20         if operator is None:
21             failures_number += 1
22         else:
23             operator.set_busy()
24             operator.generate_time(event.time)
25             events.append(Event(operator,
26                                 operator.time_next))
27
28             self.client_generator.generate_client(event.time)
29             generated_clients += 1
30             events.append(Event(self.client_generator,
31                                 self.client_generator.time_next))
32         elif isinstance(event.creator, Operator):
33             operator = event.creator
34             operator.set_free()
35
36             computer = operator.get_computer()
37             computer.append_request()
38
39             if computer.is_free() and not
40                 computer.queue_empty():
41                 computer.pop_request()
42                 computer.set_busy()
43                 computer.generate_time(event.time)
44                 events.append(Event(computer,
45                                     computer.time_next))
46         elif isinstance(event.creator, Computer):
47             computer = event.creator
48             computer.set_free()
49
50             if not computer.queue_empty():
51                 computer.pop_request()
52                 computer.set_busy()
53                 computer.generate_time(event.time)
54                 events.append(Event(computer,
55                                     computer.time_next))
56
57         return failures_number

```


2.2 Полученный результат

На рисунке 2.1 показана страница программы для определения вероятности отказа клиенту при моделировании процесса обработки запросов клиентов информационным центром с заданными в условии параметрами.

Моделирование работы информационного центра

О программе

Клиенты

Число клиентов: 300

Интервал прибытия (мин.): 10 ± 2

Информационный центр

Время обслуживания

первым оператором (мин.) 20 ± 5

вторым оператором (мин.) 40 ± 10

третьим оператором (мин.) 40 ± 20

Время обработки

первым компьютером (мин.) 15

вторым компьютером (мин.) 30

Промоделировать

Результат

Число обслуженных клиентов: 233

Число отказов: 67

Вероятность отказа: 0.22333

Рисунок 2.1 – Моделирование работы информационного центра