



Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Отчет по заданию практикума по
объектно-ориентированному программированию

Моделирование работы супермаркета

Выполнил:

студент 424 группы

Ногина Софья Олеговна

Москва, 2025

Оглавление:

I.	Уточнение постановки задачи	3
II.	Диаграмма классов	4
III.	Текстовые спецификации интерфейса основных классов системы	4
IV.	Диаграмма объектов.....	6
V.	Инструментальные средства	6
VI.	Описание файловой структуры программы.....	6
VII.	Пользовательский интерфейс.....	6

I. Уточнение постановки задачи

Необходимо разработать имитационную модель обслуживания покупателей супермаркета несколькими ($1 \leq K \leq 7$) кассами. Супермаркет работает круглосуточно.

При моделировании работы супермаркета его покупатели приходят случайным образом: случайной величиной является отрезок времени между последовательным появлением двух покупателей. Эта случайная величина имеет нормальное распределение в интервале, границы которого задаются пользователем (но должны быть больше 1 минуты), причем плотность потока заявок зависит от дня недели (доля прошедших в процессе моделирования дней), времени дня (доля прошедших часов с начала данного дня). Длительность обслуживания каждого покупателя также случайное число в некотором диапазоне, который задается пользователем, но оно не зависит от входного потока заявок. Еще одной случайной величиной является сумма покупки (границы задаются пользователем), причем сумма не зависит от других случайных величин.

Максимально возможная длина очереди у каждой кассы – N человек ($5 \leq N \leq 8$), не считая обслуживаемого покупателя. Очереди формируются по определенному закону – покупатель идет в кассу, у которой минимальная длина очереди. Если у каждой кассы скопилось очередь из N человек, то вновь прибывающие покупатели уходят, и плотность потока заявок уменьшается в два раза, пока не освободится место в одной из очередей – тем самым супермаркет теряет своих потенциальных покупателей.

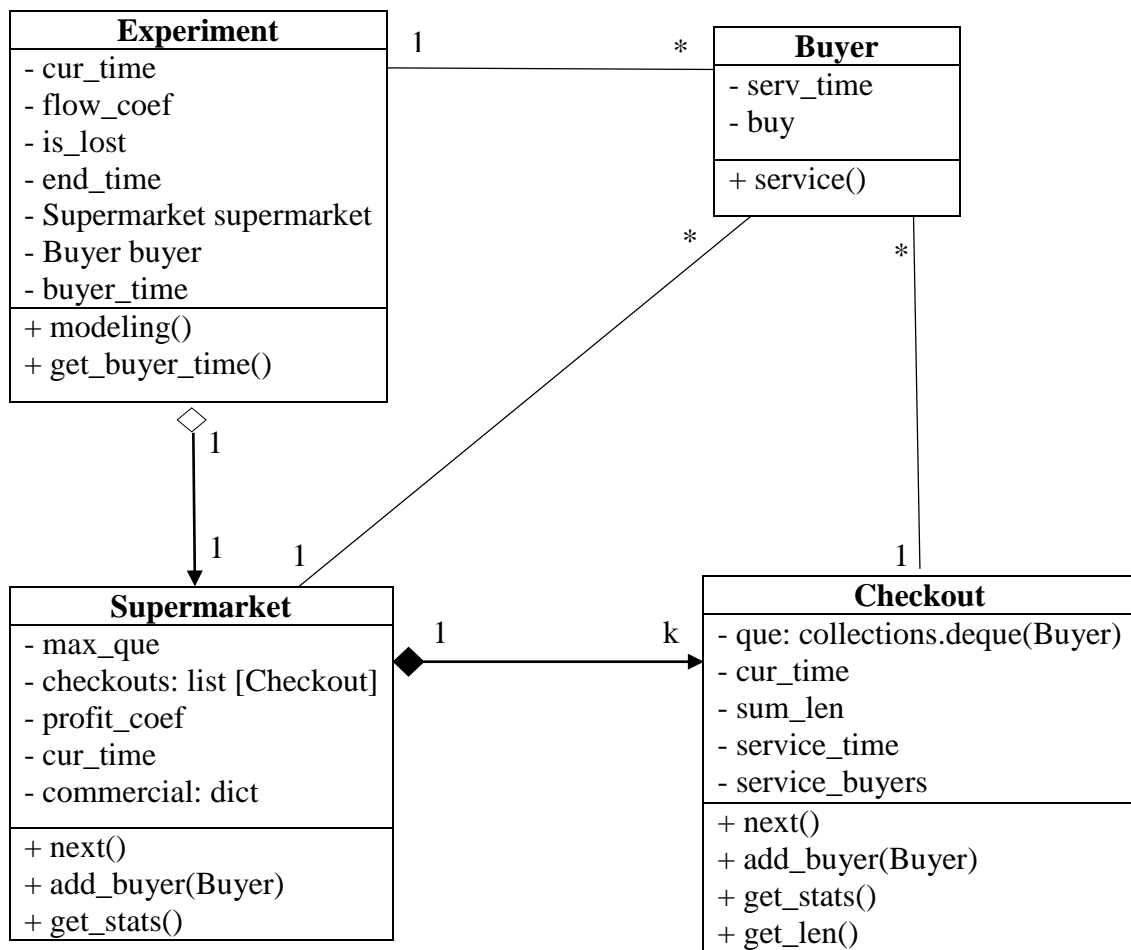
Цель исследования работы супермаркета – определение оптимальных режимов его работы, т.е. режимов, при которых работающие кассы всегда заняты, и увеличиваются прибыли от продаж. Режим работы включает число касс, рекламу и скидки на товары.

Считается, что затрата 7 тыс. руб. в день на рекламу увеличивает поток покупателей на 10%, а при объявлении скидок на товары каждый процент скидки увеличивает плотность потока на 0.5 %. Известна также средняя прибыль, получаемая при обслуживании каждого покупателя (по умолчанию 9 % от стоимости покупки), и дневная зарплата каждого кассира (по умолчанию 1.5 тыс. руб.).

Период моделирования – неделя (с понедельника 0:0 до воскресенья 23:59), шаг – интервал времени от 10 до 60 минут. Кроме шага, в параметры моделирования включены числа K и N , диапазоны разброса случайных величин – промежутка между последовательным приходом покупателей (от 1 минуты), времени их обслуживания (от 1 минуты), стоимости покупки, а также затраты на рекламу, величину скидки, прибыль от суммы покупки в 1 тыс. руб., зарплату кассира.

Визуализация моделируемого процесса предусматривает показ очередей у каждой кассы, приход и уход покупателей, а также вывод статистической информации, собираемой в ходе моделирования: количество обслуженных и потерянных (потенциальных) покупателей, средняя длина очереди у касс и среднее время ожидания в ней, средняя занятость продавцов или касс, общая прибыль, полученная супермаркетом.

II. Диаграмма классов



III. Текстовые спецификации интерфейса основных классов системы

Эксперимент

```

class Experiment:
    def __init__(self, params):
        self.params = params # Параметры моделирования
        self.cur_time = 0 # Текущее время (таймер в минутах)
        self.flow_coef = [1., 0.5] # Коэффициент потока покупателей (базовый и
        # при переполнении очередей)
        self.is_lost = 0 # Потерян ли покупатель на данный момент
        self.end_time = 7 * 24 * 60 # Конечное время моделирования
        self.lost_buyers = 0 # Число потерянных потенциальных покупателей
        d = {key: params[key] for key in ['Зарплата кассира', 'Затрата на
        рекламу (в 7000 руб.)', 'Величина скидки']} # commercial
        self.supermarket = Supermarket(params['Число касс'][0],
        params['Максимальная длина очереди'][0], params['Прибыль от суммы покупки в 1
        тыс. р.'][0], d)
        self.buyer = Buyer(params['Время обслуживания покупателя'],
        params['Стоимость покупки'])
        self.buyer_time = 0 # Время последнего пришедшего покупателя

    def modeling(self) # Моделирование потока покупателей в каждую минуту

    def get_buyer_time(self) # Получение времени прихода следующего покупателя
  
```

Супермаркет

```
class Supermarket:
    def __init__(self, cnt_checkouts, max_que, profit_procent, commercial):
        self.max_que = max_que # Максимальная длина очереди
        self.checkouts = [Checkout() for _ in range(cnt_checkouts)] # Кассы
        self.profit_coef = profit_procent / 100 # Коэффициент прибыли
        self.profit = 0 # Прибыль в рублях
        self.cur_time = 0 # Текущее время (таймер в минутах)
        self.commercial = commercial # Траты на рекламу, скидки, зп кассирам

    def next(self) # Итерация на следующую минуту

    def add_buyer(self, Bueyr) # Добавление покупателя в кассу с наименьшей
    длиной очереди

    def get_stats(self) # Получение всей собранной статистики
```

Касса

```
class Checkout:
    def __init__(self):
        self.que = collections.deque() # Очередь из покупателей
        self.cur_time = 0 # Текущее время (таймер в минутах)
        self.sum_len = 0 # Суммарная длина очереди
        self.service_time = 0 # Суммарное время обслуживания (в минутах)
        self.service_buyers = 0 # Суммарное число обслуженных покупателей

    def next(self) # Итерация на следующую минуту

    def add_buyer(self, Bueyr) # Добавление покупателя в очередь

    def get_stats(self) # Получение собранной статистики: число обслуженных
    покупателей, средние: длина очереди, время ожидания покупателя, занятость касс

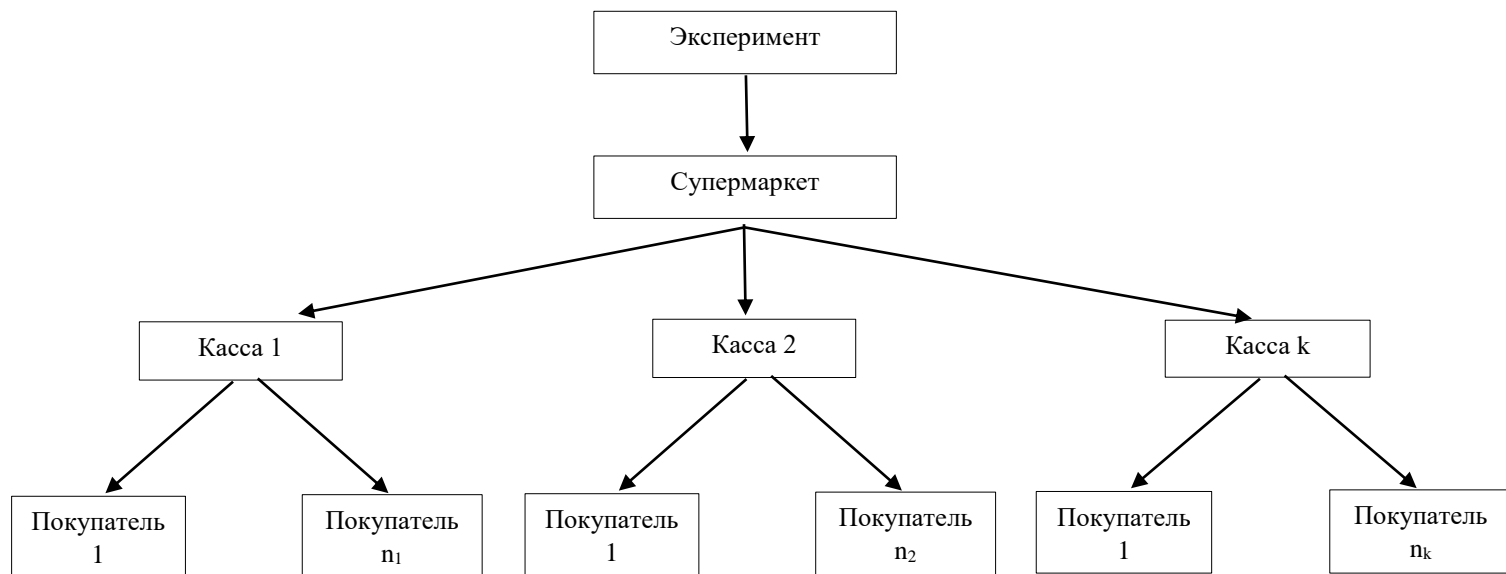
    def get_len(self) # Получение длины очереди
```

Покупатель

```
class Buyer:
    def __init__(self, limits_service, limits_buy):
        # Длительность обслуживания
        self.serv_time = random.randint(limits_service[0], limits_service[1])
        self.buy = random.randint(limits_buy[0], limits_buy[1]) # Сумма покупки

    def service(self) # Обслуживание покупателя: уменьшение длительности
    обслуживания. Если обслуживание завершено – возвращение суммы покупки
```

IV. Диаграмма объектов



V. Инструментальные средства

- Язык программирования – python 3.11.5;
- Среда разработки – PyCharm Community Edition 2023.3.4;
- Используемые библиотеки – tkinter, random, collections, numpy.

VI. Описание файловой структуры программы

- run.py – запуск программы;
- modeling.py – классы, описанные в спецификации интерфейса;
- gui.py – классы для создания пользовательского интерфейса.

VII. Пользовательский интерфейс

В начале работы программы появляется окно задания параметров

Параметры моделирования			
Шаг моделирования (мин.)	<input type="text" value="10"/>		
Число касс	<input type="text" value="1"/>		
Максимальная длина очереди	<input type="text" value="5"/>		
Промежуток между приходом двух покупателей (от 1 мин.)	от <input type="text" value="1"/>	до <input type="text" value="7"/>	
Время обслуживания покупателя (от 1 мин.)	от <input type="text" value="1"/>	до <input type="text" value="7"/>	
Стоимость покупки (от 0 руб.)	от <input type="text" value="30"/>	до <input type="text" value="9000"/>	
Прибыль от суммы покупки в 1 тыс. р. (от 0 до 100 %)	<input type="text" value="9"/>		
Зарплата кассира (от 0 руб.)	<input type="text" value="1500"/>		
	Затрата на рекламу (от 0 руб.)	Величина скидки (от 0 до 100 %)	
Понедельник	<input type="text" value="0"/>	<input type="text" value="0"/>	
Вторник	<input type="text" value="0"/>	<input type="text" value="0"/>	
Среда	<input type="text" value="0"/>	<input type="text" value="0"/>	
Четверг	<input type="text" value="0"/>	<input type="text" value="0"/>	
Пятница	<input type="text" value="0"/>	<input type="text" value="0"/>	
Суббота	<input type="text" value="0"/>	<input type="text" value="0"/>	
Воскресенье	<input type="text" value="0"/>	<input type="text" value="0"/>	
<input type="button" value="Начать моделирование"/> <input type="button" value="Завершить работу"/>			

При неправильном задании параметров ячейки подсвечиваются красным:

The screenshot shows the 'Параметры моделирования' (Modeling Parameters) window. It contains various input fields for modeling parameters. Several fields are highlighted in red, indicating errors: 'Максимальная длина очереди' (Maximum queue length) is 5, 'Промежуток между приходом двух покупателей' (Interval between two customers) is 'ыавыа', 'Время обслуживания покупателя' (Service time) is 0, 'Стоимость покупки' (Purchase cost) is 9000, 'Прибыль от суммы покупки' (Profit from the sum of purchases) is 9, 'Зарплата кассира' (Cashier salary) is 1500, 'Затрата на рекламу' (Advertising cost) is 0, and 'Величина скидки' (Discount amount) is 101. An error dialog box is open in the top right corner with the message: 'Исправьте значения, выделенные красным' (Correct the values highlighted in red). The dialog has an 'ОК' button.

При нажатии на кнопку «Завершить работу» окно задания параметров закрывается.

При нажатии на кнопку «Начать моделирование» при правильно введенных данных открывается окно моделирования:

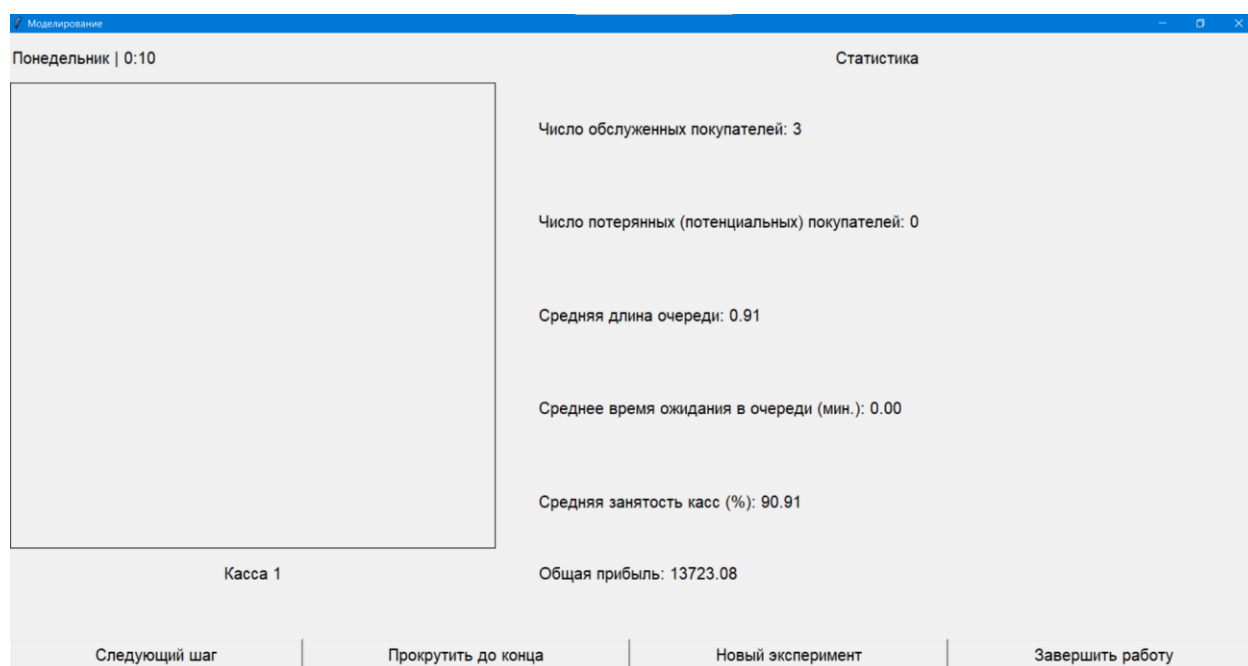
The screenshot shows the 'Моделирование' (Modeling) window. It displays the results of a simulation. On the left, there is a visual representation of the queue system with a single cashier labeled 'Касса 1'. On the right, under the 'Статистика' (Statistics) tab, the following data is shown: 'Число обслуженных покупателей' (Number of served customers): 0, 'Число потерянных (потенциальных) покупателей' (Number of lost (potential) customers): 0, 'Средняя длина очереди' (Average queue length): 1.00, 'Среднее время ожидания в очереди (мин.):' (Average waiting time in queue (min.)): 0.00, 'Средняя занятость касс (%)' (Average cashier occupancy (%)): 100.00, and 'Общая прибыль: 0.00' (Total profit: 0.00). At the bottom, there are four buttons: 'Следующий шаг' (Next step), 'Прокрутить до конца' (Scroll to end), 'Новый эксперимент' (New experiment), and 'Завершить работу' (End work).

В левом верхнем углу отображается текущее время: день недели, час, минута.

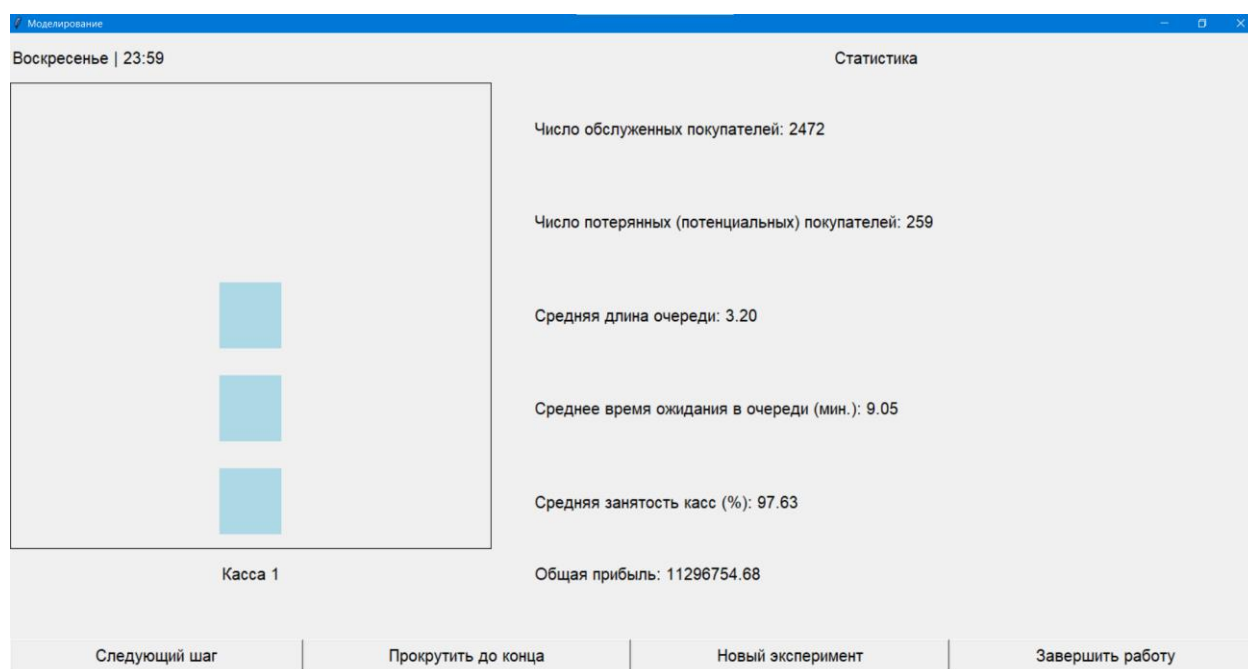
В правой части окна отображается собранная с начала эксперимента статистика.

В левой части отображаются кассы, в которых голубыми квадратами показываются покупатели, находящиеся в соответствующе очереди.

При нажатии на кнопку «Следующий шаг» отображаются изменения, произошедшие за шаг моделирования:



При нажатии на кнопку «Прокрутить до конца» отображается состояние супермаркета на конец эксперимента:



При нажатии на кнопку «Новый эксперимент» открывается окно задания параметров, где по умолчанию выводятся параметры, введенные пользователем в последний раз.

При нажатии на кнопку «Завершить работу» окно моделирования закрывается.