Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

Лабораторная работа

по дисциплине: «Моделирование процессов и систем»

по теме: «Системы массового обслуживания»

Выполнил:

Студент группы

Соболь Евгений Валерьевич

Проверил:

доцент кафедры ВМиМ

Максимов Петр Викторович

Пермь

2021 г

**Цель работы:** разработать программный алгоритм, позволяющий проводить моделирование кассового обслуживания покупателей продуктового магазина. Смоделировать различные ситуации, задавая соответствующим образом входные параметры, научиться анализировать результаты.

**Задание к лабораторной работе:**

1. Разработать программный алгоритм, позволяющий проводить моделирование кассового обслуживания покупателей продуктового магазина.
2. Провести моделирование работы магазина несколько раз, самостоятельно определить, сколько именно, чтобы целое число покупателей после осреднения не менялось. Необходимо находить среднее значения по каждой кассе.
3. Смоделировать различные ситуации, задавая соответствующим образом входные параметры, для каждой ситуации предложить разумный вариант ее разрешения. Рассмотреть в работе не менее 3-4 разных жизненных ситуаций.

**Основная часть:**

1. Из формулировки задания определим ***входные данные*** алгоритма моделирование кассового обслуживания:

* Количество касс
* Время обработки заказа одного покупателя
* Плотность потока заявок
* Количество опытов

1. Определим ***выходные данные***, алгоритма моделирования кассового обслуживания:

* Количество человек, обслуженных каждой кассой
* Число потерянных не обслуженных покупателей
* Количество человек, обслуженных каждой кассой в среднем, за заданное количество опытов
* Число потерянных не обслуженных покупателей в среднем за заданное количество опытов
* Среднее число посетителей
* Среднее число покупателей

Схема алгоритма работы программы:

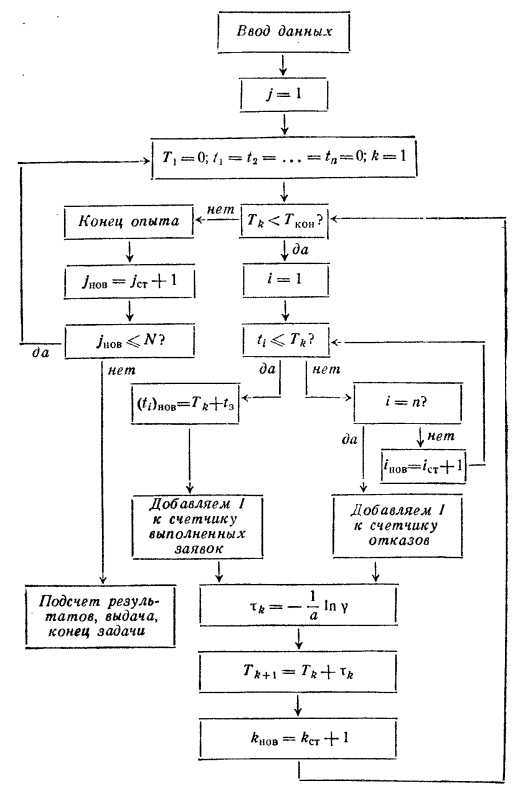


Рисунок 1 - Схема работы алгоритма

Где,  - момент освобождения - й линии ,  - момент поступления первого клиента,  - время окончания расчета,  - время, за которое обслуживается покупатель,  - сгенерированное время поступления нового клиента,  - обслуженных либо не обслуженных посетителей,  – нумерация опытов.

**Описание алгоритма работы программы:**

1. Вводим входные значения
2. Генерируем случайное число от 0 до 1 по нормальному закону распределения .
3. Генерируем время случайного посещения магазина
4. Проверяем, не закончилось ли время моделирования
5. Проверяем, есть ли свободная касса, если есть, то изменяем значение счётчика кассы и записываем новое время окончания, обслуживая посетителя, если нет, увеличиваем счётчик отказов
6. Повторяем шаги b, c, d, e до окончания времени моделирования
7. Повторяем опыт заданное число раз
8. Выводим результаты

**Реализуем систему массового обслуживания на Python:**

1. Импортируем всё необходимое для написания алгоритма, функцию для генерации числа по нормальному закону распределения, функцию логарифма, экспоненту, функцию корня.

from random import normalvariate

from math import log, e, sqrt

1. Создадим функцию генерации числа от 0 до 1 по нормальному закону распределения с помощью библиотеки random:

def normal():

while True:

s = normalvariate(0.5, sqrt(0.5))

if (s > 0) and (s < 1):

return s

1. Создадим функцию tilda, которая будет генерировать время случайной заявки по формуле :

def tilda(a: float, y: float) -> float:

return -(1/a \* log(y, e))

1. Создадим вспомогательную функцию для генерации списков нужного размера заполненных нулями:

def create\_list(n: int) -> list:

return [0 for i in range(n)]

1. Создадим основную функцию main, в которой будет основная часть алгоритма, инициализируем необходимые переменные и запросим у пользователя входные значения:

def main():

s\_buyers = 0

result = []

n\_exp = int(input("(>) Количество опытов: "))

n = int(input("(>) Количество касс: "))

processing\_time = int(input("(>) Время обслуживания покупателя: "))

total\_time = int(input("(>) Общее время моделирования: "))

apl\_flow\_density = float(input("(>) Плотность потока заявок: "))

1. Создадим цикл количества опытов, инициализируем список из 2 списков, который будет хранить счётчики количества обслуженных клиентов каждой кассы, счётчик отказов и список со временем занятости касс:

for i\_exp in range(n\_exp):

list\_cashbox = [create\_list(n + 1), create\_list(n)]

time = 0

n\_buyer = 0

1. Внутри предыдущего цикла создадим ещё один цикл, который будет работать до тех пор, пока не закончится время моделирования, а также инициализируем переменную, которая будет отвечать за перемещение по кассам:

while time <= total\_time:

i = 0

1. Создадим цикл, который будет определять, есть ли свободная касса для посетителя и либо принимать его, либо отказывать:

while True:

if list\_cashbox[1][i] <= time:

list\_cashbox[1][i] = time + processing\_time

list\_cashbox[0][i] += 1

print(f"(+) {i} КАССУ ЗАНЯЛ ПОКУПАТЕЛЬ №{n\_buyer}, ОСВОБОДИТСЯ В {list\_cashbox[1][i]}")

break

else:

if i == (n - 1):

list\_cashbox[0][n] += 1

print(f"(-) ПОКУПАТЕЛЮ №{n\_buyer} ОТКАЗАНО В ОБСЛУЖИВАНИИ")

break

else:

i += 1

1. Генерируем время новой заявки в теле цикла, созданного на 7 шаге, увеличиваем счётчик посетителей:

time += tilda(apl\_flow\_density, normal())

n\_buyer += 1

print(f"(!) ПРИШЕЛ ПОКУПАТЕЛЬ № {n\_buyer} - ВРЕМЯ {time}")

1. После окончания времени опыта, условие цикла, созданного в 7 шаге не выполняется, добавляем список с количеством обслуженных кассами клиентов в список с результатами опытов, увеличиваем общую сумму посетителей за опыты:

print(f"(!) ПРИШЕЛ ПОКУПАТЕЛЬ № {n\_buyer} - ВРЕМЯ {time}")

print(f"----------КОНЕЦ {i\_exp + 1} ОПЫТА---------------")

result.append(list\_cashbox[0])

1. Создадим функцию вывода результатов опытов, выведем результаты по каждому опыту и среднее значения данных по опытам, и вызовем её:

def writ\_experience(list\_result: list, n\_exp: int, s\_buyers: int):

list\_mean = []

s\_customer = 0

print(" ---------------РЕЗУЛЬТАТЫ ОПЫТОВ---------------")

for i in range(n\_exp):

length = len(list\_result[i])

print(f" -----РЕЗУЛЬТАТЫ {i + 1} ОПЫТА-----")

for j in range(length - 1):

if i == 0:

list\_mean.append(list\_result[i][j])

else:

list\_mean[j] += list\_result[i][j]

print(f"(<) {j+1} КАССА ОБСЛУЖИЛА {list\_result[i][j]} ЧЕЛ.")

if i == 0:

list\_mean.append(list\_result[i][-1])

else:

list\_mean[-1] += list\_result[i][-1]

print(f"(<) ОТКАЗАННО - {list\_result[i][length-1]} ЧЕЛ.")

print(" ---------------УСРЕДНЁННЫЕ РЕЗУЛЬТАТЫ ОПЫТОВ---------------")

for i in range(len(list\_mean)-1):

print(f"(<) {i + 1} КАССА ОБСЛУЖИЛА В СРЕДНЕМ {list\_mean[i]/n\_exp} ЧЕЛ.")

s\_customer += list\_mean[i]/n\_exp

print(f"(<) ОТКАЗАННО В СРЕДНЕМ - {list\_mean[-1]/n\_exp} ЧЕЛ.")

print(f"(<) СРЕДНЕЕ ЧИСЛО ПОСЕТИТЕЛЕЙ: {s\_buyers/n\_exp}")

print(f"(<) CРЕДНЕЕ ЧИСЛО ПОКУПАТЕЛЕЙ: {s\_customer}")

Вызов функции вывода результатов идёт в самом конце функции main.

writ\_experience(result, n\_exp, s\_buyers)

**КОНЕЦ ПРОГРАММЫ**

**Моделирование ситуаций:**

***Ситуация 1:*** Просмотрев камеры в магазине, мы выяснили, что к кассе подходит покупатель раз в 62,5 секунды, на обслуживание кассир в среднем тратит 70 секунд, в магазине находится 5 кассовых аппаратов, необходимо определить, сколько в среднем обслуживает каждая касса человек, за 12 часовой рабочий день.

Забиваем входные данные:

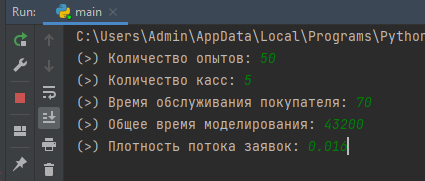


Рисунок 2 - Экран ввода ситуации 1

Запускаем моделирование:

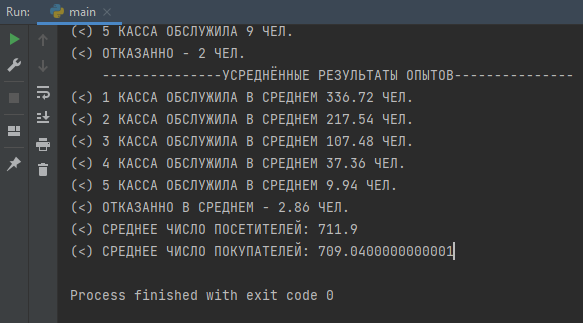


Рисунок 3 - Экран вывода ситуации 1

Получаем, что за день в магазине было примерно 712 посетителей, из них 709 человек (99,57%) были обслужены моментально без очереди, 3 человека (0,42%) не были обслужены, так как по правилам образование очередей запрещено, при этом основная нагрузка легла на 3 первые кассы, 93,33% покупателей обслуживались на них. 1 касса обслужила примерно 337 покупателей (47%), 2 касса обслужила 217 покупателей (30,47%), 3 касса обслужила 107 покупателей (15,02%), 4 касса обслужила 37 покупателей (5,19%), 5 касса обслужила всего 10 покупателей (1,4%).

Вывод: система при заданной нагрузке в текущей ситуации работает отлично, при сильной необходимости можно будет сократить количество работников и касс до 3, что приведёт к проценту отказов примерно в 7%. Что достаточно несущественно скажется на бизнесе.

***Ситуация 2****:* Бизнесмен хочет открыть цветочную лавку на 7-8 марта (по 12 часов в день), в предыдущем году, в среднем был посетитель каждые 312,5 секунд, продавец, в среднем, обслуживал покупателя за 150 секунд. Необходимо определить, сколько продавцов необходимо нанять, чтобы без очереди обслужить посетителей, так как конкуренция большая и ждать посетители не будут, продавца можно нанять на 12 часовую смену за 4000 рублей. Средний чек покупателя составляет 500 рублей.

Предположим, что мы наймём 1 продавца:

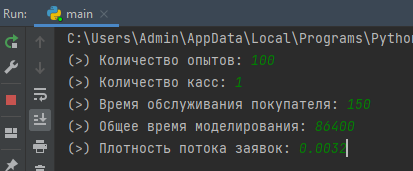


Рисунок 4 - Экран ввода ситуации 2 (опыт 1)

Запускаем моделирование:

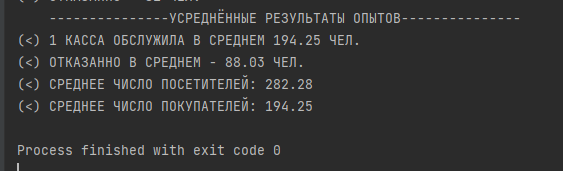


Рисунок 5 - - Экран вывода ситуации 2 (опыт 1)

Получаем, что всего цветочная лавка обслужила 194 покупателя (68,79%), не смогла обслужить 88 посетителей (31,21%). Получается, что прибыль составила 97000 рублей, упущенная прибыль 44000 рублей. Из этого следует, что необходимо нанять большее количество продавцов.

Предположим, что мы наймём 2 продавцов:

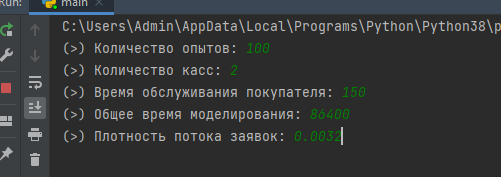


Рисунок 6 - Экран ввода ситуации 2 (опыт 2)

Запускаем моделирование:

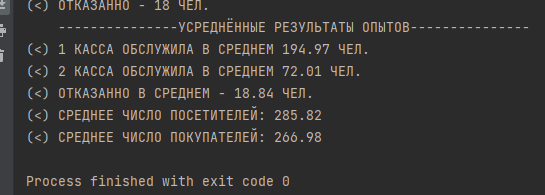


Рисунок 7 - - Экран вывода ситуации 2 (опыт 2)

Получаем, что цветочная лавка обслужила 267 покупателей (93,35%), не смогла обслужить 19 посетителей (6,65%). Получается, что прибыль составила 133500 рублей, упущенная прибыль составила 9500 рублей. Из этого следует, что по прежнему выгодно нанять ещё одного продавца.

Предположим, что мы наймём 3 продавцов:

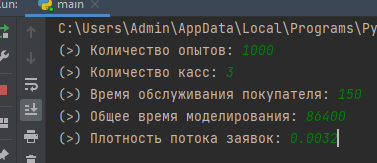


Рисунок 8 - Экран ввода ситуации 2 (опыт 3)

Запустим моделирование:

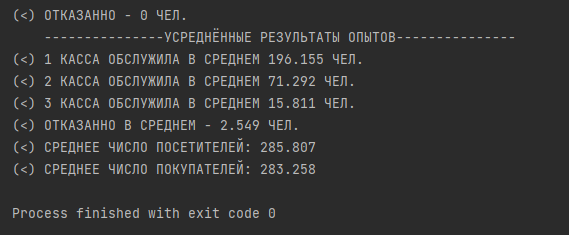


Рисунок 9 - Экран вывода ситуации 2 (опыт 3)

Получаем, что всего цветочная лавка обслужила 283 покупателя (98,95%), не смогла обслужить 3 посетителей (1,05%). Получается, что прибыль составила 141500 рублей, упущенная прибыль 1500 рублей. Из этого следует, что максимальную прибыль бизнесмен получит, если наймёт 3 продавцов, нанимать 4 не рентабельно, так как возможная прибыль от его работы меньше размера оплаты его труда.

***Ситуация 3:*** В новом микрорайоне хотят поставить аппарат по продаже воды, в среднем в подобном микрорайоне воду покупают раз в 1250 секунд, среднее время покупки составляет 300 секунд. Автомат работает круглые сутки, необходимо посмотреть, сколько покупателей он обслужит в среднем за сутки, а скольких не сможет обслужить, и они пойдут в другое место. Есть ли нужда в ещё одном автомате, если стоимость его обслуживания в день равна 700 рублей, а средний чек у покупателей 50 рублей.

Забиваем входные данные:

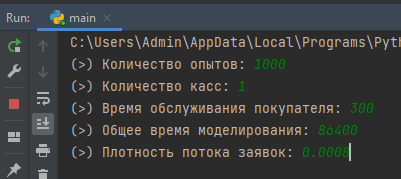


Рисунок 10 - Экран ввода ситуации 3

Запускаем моделирование:

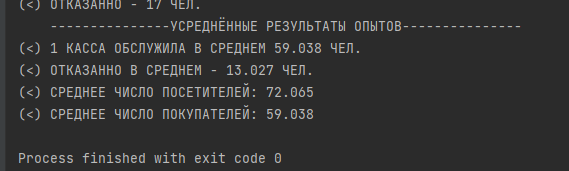


Рисунок 11 - - Экран вывода ситуации 3

Получаем, что автомат обслужил 59 покупателей (81,94%), не смог обслужить 13 покупателей (18,06%). Прибыль за день составила 2950 рублей. Упущенная прибыль составила 650 рублей, что является меньше чем стоимость обслуживания в день второго автомата, поэтому установка второго автомата не рентабельна.

**Заключение:**

В данной лабораторной работе были достигнуты следующие результаты:

* разработан программный алгоритм, позволяющий проводить моделирование кассового обслуживания покупателей продуктового магазина;
* реализован данный алгоритм на языке программирования Python;
* смоделированы различные ситуации, задавая соответствующим образом входные параметры написанной программе;
* проанализированы результаты моделирования.