Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №3**

по дисциплине: «Компьютерные системы моделирования»

Тема: " Имитационное моделирование. Моделирование системы массового обслуживания "

Часть 1.

|  |
| --- |
| Выполнил:  студент группы  Б.ПИН.РИС-21.06  Леонов А.М. |
| Проверила:  старший преподаватель кафедры ПО  Корнеева Е.И. |

Тверь 2024

**Оглавление**

**[Цель работы:](#_Toc161857274)** [3](#_Toc161857274)

[**Задание:** 3](#_Toc161857275)

[**Код программы на python:** 4](#_Toc161857276)

[**Ссылки на GitHub:** 8](#_Toc161857277)

[**Решение задач:** 8](#_Toc161857278)

[**Задача 1:** 8](#_Toc161857279)

[**Задача 2:** 9](#_Toc161857280)

[**Задача 3:** 9](#_Toc161857281)

[**Задача 4:** 9](#_Toc161857282)

[**Задача 5:** 9](#_Toc161857283)

[**Задача 6:** 10](#_Toc161857284)

[**Вывод по работе** 10](#_Toc161857285)

# **Цель работы:**

Для задач в задании необходимо составить расчеты характеристик СМО. Задачи 1, 2, 4 относятся к одноканальным СМО. Задачи 3, 5, 6 относятся к многоканальным СМО. Разработать программу на python для вывода характеристик СМО.

# **Задание:**

**Задача 1.** Одноканальная СМО с отказами представляет собой одну телефонную линию.

Заявка (вызов), пришедшая в момент, когда линия занята, получает отказ. Все потоки событий простейшие. Интенсивность потока λ=0,95 вызова в минуту. Средняя продолжительность разговора – 1 минута. Определить вероятность отказа, вероятность обслуживания, отношение числа обслуженных к числу необслуженных заявок.

**Задача 2.** Автозаправочная станция представляет собой СМО с одним каналом обслуживания и одной колонкой. Площадка при АЗС допускает пребывание в очереди на заправку не более трех автомобилей одновременно. Если в очереди уже находится три автомобиля, очередной автомобиль, прибывший к станции, в очередь не становится, а проезжает мимо. Поток автомобилей, прибывающих для заправки, имеет интенсивность = 0,7 автомобиля в минуту.

Процесс заправки продолжается в среднем 1,25 мин. Все потоки простейшие. Определите вероятностные характеристики СМО в стационарном режиме.

**Задача 3.** В инструментальном отделении сборочного цеха работают три кладовщика. В среднем за 1 мин. за инструментом приходят 0,8 рабочего. Обслуживание одного рабочего занимает у кладовщика 1 мин. Очередь не имеет ограничения. Известно, что поток рабочих за инструментом - пуассоновский, а время обслуживания подчинено экспоненциальному закону распределения. Определите вероятностные характеристики СМО.

**Задача 4.** Пост диагностики автомобилей представляет собой одноканальную СМО с отказами. Заявка на диагностику, поступившая в момент, когда пост занят, получает отказ.

Интенсивность потока заявок на диагностику = 0,5 автомобиля в час. Средняя продолжительность диагностики = 1,2 часа. Все потоки событий в системе простейшие.

Определите в установившемся режиме вероятностные характеристики системы.

**Задача 5.** Рассматривается работа АЗС, на которой имеется три заправочные колонки.

Заправка одной машины длится в среднем 3 мин. В среднем на АЗС каждую минуту прибывает машина, нуждающаяся в заправке бензином. Число мест в очереди не ограничено.

Все машины, вставшие в очередь на заправку, дожидаются своей очереди. Все потоки в системе простейшие. Определите вероятностные характеристики работы АЗС в стационарном режиме.

**Задача 6.** На станцию технического обслуживания (СТО) автомобилей каждые три часа подъезжает в среднем одна машина. Станция имеет 2 поста обслуживания. Очередь автомобилей, ожидающих обслуживания ограничена пятью автомобилями. Среднее время обслуживания одной машины - 2 часа. Все потоки в системе простейшие. Определите вероятностные характеристики станции технического обслуживания автомобилей.

# **Код программы на python:**

Программа представляет из себя командный интерфейс с методами для вычисления СМО. Реализованы одноканальная с отказами, одноканальная с ограниченной и неограниченной очередью, многоканальная с отказами, многоканальная с ограниченной и неограниченной очередью.

import math

class Calculat():

    #Одноканальная с отказами в обслуживании

    def Single\_channel\_failures(self, l, tobsl):

        mu = 1/tobsl

        tprost = 1 / l

        roobs = l / (l + mu)

        rootk = 1 - roobs

        a = l \* roobs

        return round(mu, 2), round(tprost,2), round(roobs,2), round(rootk,2), round(a,2)

    # Одноканальная с ограниченной и неограниченной очередью

    def Single\_channel\_limited\_or\_unlimited(self, l, tobsl, n, f):

        mu = 1/tobsl

        ro = l / (l + mu)

        if not f: #f - показатель с ограниченной очередью или нет

            if ro == 1:

                r\_0 = 1/(n+2)

                lq = (n\*(n+1))/(2\*(n+2))

            else:

                r\_0 = (1-ro)/(1-ro\*\*(n+2))

                lq = ro\*\*2 \* ((1-ro\*\*n\*(n-n\*ro+1))/(1-ro)\*\*2)

            rootkaz = ro\*\*(n+1)\*r\_0

            roobsl = 1 - rootkaz

            a = l\*roobsl

            tq = lq/l

            lsmo = 1 + lq

            if ro == 1:

                tsmo = (n+1)/(2\*mu)

            else:

                tsmo = lsmo/l

            return round(roobsl,2), round(rootkaz,2), round(lq,2), round(tq,2), round(lsmo,2), round(tsmo,2),round(r\_0, 2)

        else:

            if ro >= 1:

                print('Режим не явлеятся стационарным')

                return False

            lq = (ro\*\*2)/(1-ro)

            tq = lq/l

            lsmo = ro/(1-ro)

            tsmo = ro/l

            return round(lq,2),round(tq,2),round(lsmo,2),round(tsmo,2)

    # Многоканальная с отказами в обслуживании

    def Multi\_channel\_failures(self, l, tobsl, n):

        mu = 1/tobsl

        ro = l/mu

        ro\_0 = 0

        for i in range(0, n, 1):

            ro\_0 += (ro\*\*i)/(math.factorial(i))

        ro\_0 = ro\_0\*\*-1

        rootkaz = ro\_0 \* (ro\*\*n)/math.factorial(n)

        roobsl = 1 - rootkaz

        a = l\*roobsl

        k\_ = l/mu

        return round(ro\_0,2), round(rootkaz,2), round(roobsl,2),round(a,2), round(k\_,2)

    #Многоканальная с ограниченной очередью

    def Multi\_channel\_limited(self, l, tobsl, n, m):

        mu = 1/tobsl

        ro = l/mu

        ro\_0 = 0

        if ro/n == 1:

            for k in range(0, n, 1):

                ro\_0 += (ro\*\*k)/math.factorial(k) + (m \* (ro\*\*(n+1)))/(n \* math.factorial(n))

            ro\_0 = ro\_0\*\*-1

            lq = (ro\*\*(n+1)\*m\*(m+1)\*ro\_0)/(n\*math.factorial(n)\*2)

        else:

            for k in range(0, n, 1):

                ro\_0 += (ro\*\*k)/math.factorial(k) + (ro\*\*(n+1))/(math.factorial(n)\*(n-ro))\*(1 - (ro/n)\*\*m)

            ro\_0 = ro\_0\*\*-1

            lq = (ro\*\*(n+1))/(n\*math.factorial(n)) \* (1 - (ro/n)\*\*(m) \* (m+1-m\*ro/n))/((1 - ro/n)\*\*2)

        rootkaz = (ro\*\*(n+m))/(n\*\*(m)\*math.factorial(n)) \* ro\_0

        tq = lq/l

        tsmo = tq + (1 - rootkaz)/mu

        k\_ = (l/mu)\*(1 - rootkaz)

        return round(ro\_0,2), round(rootkaz,2), round(lq,2),round(tq,2),round(tsmo,2), round(k\_,2)

    #Многоканальная с неограниченной очередью

    def Multi\_channel\_unlimited(self, l, tobsl, n):

        mu = 1/tobsl

        ro = l/mu

        if(ro/n>= 1):

            print('Режим нестационнарный!')

            return False

        ro\_0 = 0

        for k in range(0, n, 1):

            ro\_0 += (ro\*\*k)/(math.factorial(k))+(ro\*\*(n+1))/(math.factorial(n)\*(n-ro))

        ro\_0 = ro\_0\*\*-1

        roq = (ro\*\*(n+1)\*ro\_0)/(math.factorial(n)\*(n-ro))

        lq = n \* roq / (n - ro)

        tq = lq / l

        lsmo = lq  + ro

        tsmo = tq + tobsl

        return round(ro\_0,2), round(roq,2), round(lq,2),round(tq,2),round(lsmo,2), round(tsmo,2)

if \_\_name\_\_ == '\_\_main\_\_':

    ca = Calculat()

    cmd = -2564

    while cmd != 3:

        print('Выберите канальность:\n1. Одноканальность \n2. Многоканальность\n3. Выход')

        cmd = int(input('Для продолжение введите число:'))

        match cmd:

            case 1:

                print('Выберите вид СМО: \n1. С отказами в обслуживании\n2. С ограниченной очередью\n3. С неограниченной очередью')

                cmd1 = int(input('Для продолжение введите число:'))

                match cmd1:

                    case 1:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        res = ca.Single\_channel\_failures(l, tobsl)

                        print(f"Вероятность обслуживания-{res[3]}, вероятность отказа-{res[2]}, интенсивность обслуживания-{res[0]}, время обслуживания-{tobsl}, время простоя-{res[1]}, абсолютная пропускная способность-{res[4]}")

                    case 2:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        n = int(input('Введите максимум элементов в очереди: '))

                        res = ca.Single\_channel\_limited\_or\_unlimited(l, tobsl, n, False)

                        print(f"P\_0-{res[5]}, вер отказа-{res[1]}, ср длина оч-{res[2]}, ср время ож в оч-{res[3]}, относит пропускная способность-{res[0]}, ср время нахождения в СМО-{res[4]}, число в СМО-{res[3]}")

                    case 3:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        res = ca.Single\_channel\_limited\_or\_unlimited(l, tobsl, 0, True)

                        if not res:

                            continue

                        print(f"ср число заявок в оч-{res[0]}, ср число заявок в смо-{res[2]}, ср время ож в оч-{res[1]}, ср время преб в смо-{res[3]}")

            case 2:

                print('Выберите вид СМО: \n1. С отказами в обслуживании\n2. С ограниченной очередью\n3. С неограниченной очередью')

                cmd1 = int(input('Для продолжение введите число:'))

                match cmd1:

                    case 1:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        n = int(input('Введите колличество каналов: '))

                        res = ca.Multi\_channel\_failures(l, tobsl, n)

                        print(f"Р\_0-{res[0]}, вер отказа-{res[1]}, вер обслуж-{res[2]}, абс проп способность-{res[3]}, ср число зан каналов-{res[4]}")

                    case 2:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        n = int(input('Введите колличество каналов: '))

                        m = int(input('Введите максимум в очереди: '))

                        res = ca.Multi\_channel\_limited(l, tobsl, n, m)

                        print(f"ro\_0-{res[0]}, вероятность отказа-{res[1]}, ср длина оч-{res[2]}, ср время ож в оч-{res[3]}, ср время ож в СМО-{res[4]}, ср число занятых каналов-{res[5]}")

                    case 3:

                        l = float(input('Введите интенсивноть потока: '))

                        tobsl = float(input('Введите время обслуживания: '))

                        n = int(input('Введите колличество каналов: '))

                        res = ca.Multi\_channel\_unlimited(l, tobsl, n)

                        if not res:

                            continue

                        print(f"ro\_0-{res[0]}, вер образования оч-{res[1]}, ср длина оч-{res[2]}, ср время ож в оч-{res[3]}, ср число в СМО-{res[4]}, ср время в СМО-{res[5]}")

            case 3:

                break

            case \_:

                print('Введено не верное число')

                continue

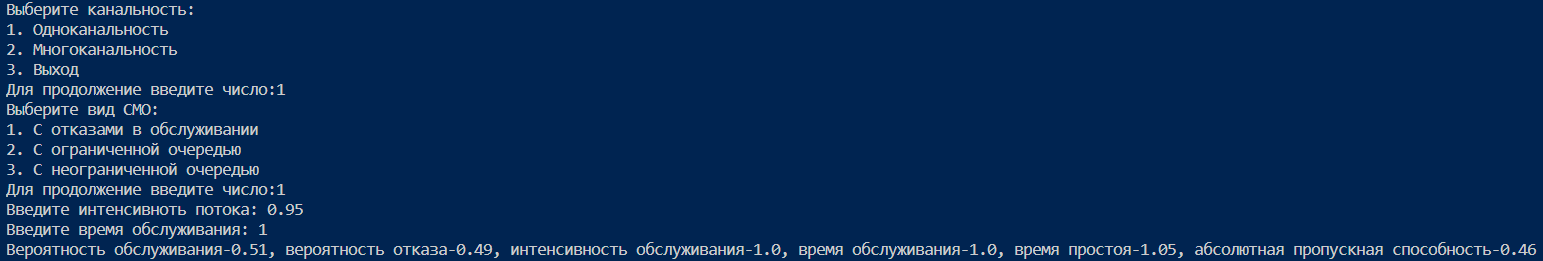
# **Ссылки на GitHub:**

Github- <https://github.com/akkafe1ix/ComputerSimulationLab3Part1>

# **Решение задач:**

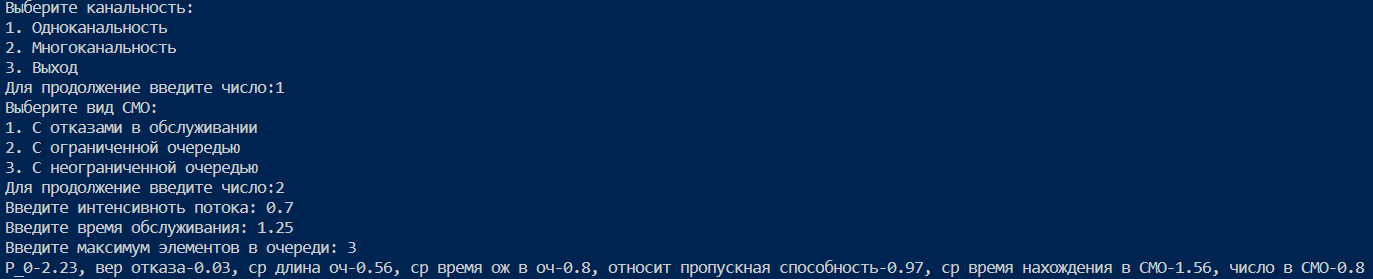
## **Задача 1:**

Это одноканальная СМО с отказами.



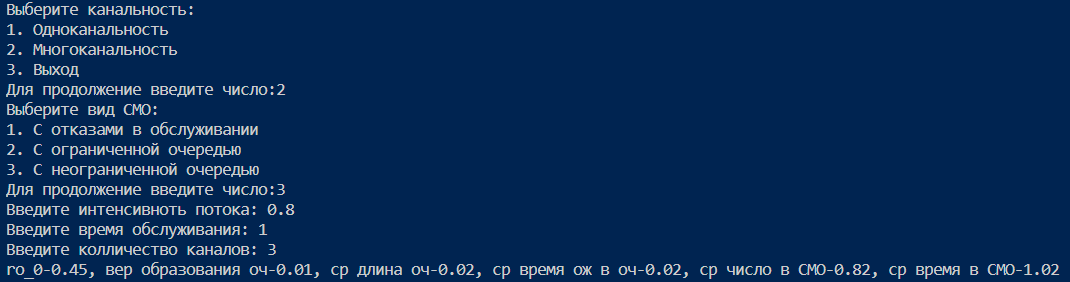
## **Задача 2:**

Это одноканальная СМО с ограниченной очередью.



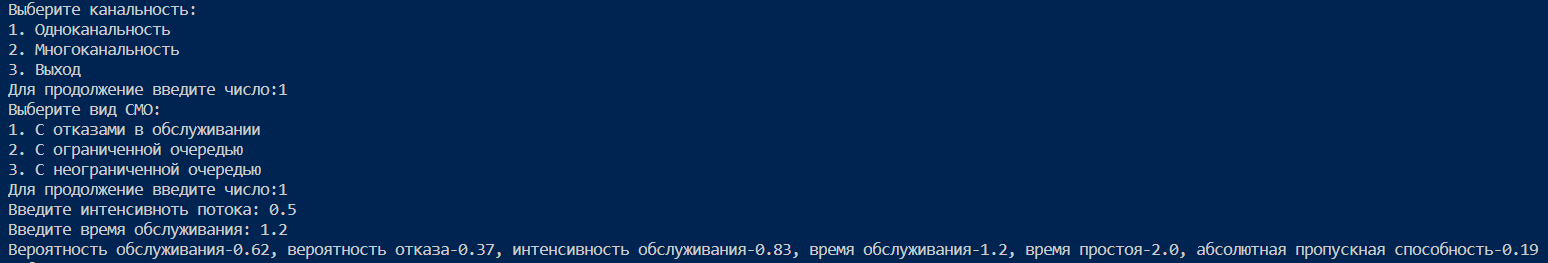
## **Задача 3:**

Это многоканальная СМО с неограниченной очередью.



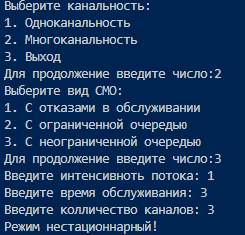
## **Задача 4:**

Это одноканальная СМО с отказами.



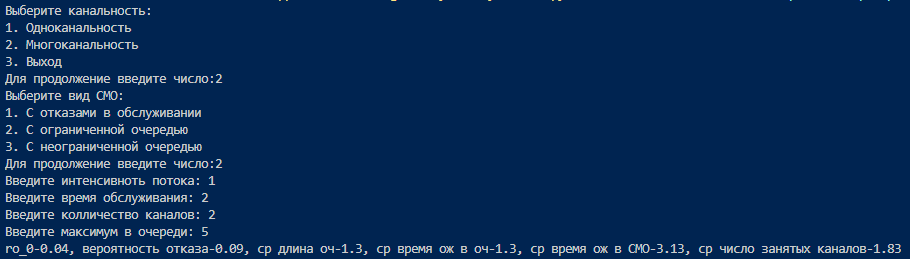
## **Задача 5:**

Это многоканальная СМО с неограниченной очередью.



## **Задача 6:**

Это многоканальная СМО с ограниченной очередью.



# **Вывод по работе**

Разобрались в видах СМО, научились решать задачи с использованием СМО в различных видах деятельности с различными входными данными.

На деле увидели расчётные характеристики такие как: Вероятность отказа, средняя длина очереди, среднее время в ожидании, среднее число занятых каналов и т.д.