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 *Робототехники и комплексная автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ УЧЕБНОЙ ПРАКТИКИ №3

по дисциплине: «Учебно-технологическая практика на C++»

Студент: _____ **Новокшанов Е.А.**
подпись, дата *фамилия, и.о.*

Группа: **РК6-36Б**

Преподаватель: _____ **Берчун Ю.В.**
подпись, дата *фамилия, и.о.*

Москва, 2021 г

Оглавление

Постановка задачи

Требуется разработать программу, реализующую *дискретно-событийное моделирование* системы, рассмотренной в задании 2 домашнего задания №4:

- Смоделируем поведение покупателя в магазине, в котором работают 2 кассы, причём к каждой из них выстраивается отдельная очередь, а квалификация сотрудников немного отличается, поэтому время обслуживания распределено с разными параметрами. Каждая касса будет представлена одноканальным устройством, обращение к которым будем осуществлять по номерам. Очереди также будут идентифицироваться номерами, без введения символьных имён. Моделирование будем проводить в течение 1 часа, в качестве единицы времени будем выбирать секунду.
- Время между приходом покупателей распределено на отрезке [0; R1+G1+B1] ([0; 24]). Время обслуживания на первой кассе распределено на отрезке [R1; R1+G1+B1] ([10;24]). Время обслуживания на второй кассе распределено на отрезке [G1; R1+G1+B1] ([9;24]).
- При принятии решения покупатель сперва проверяет, есть ли свободная касса, и, если есть, направляется к ней.
- Если же обе кассы заняты, то выбирает кассу, очередь к которой в данный момент короче.
- Если же свободны обе кассы, или очередь к ним одинакова, то выбирается первая касса.

Обратите внимание, что все интервалы времени подчиняются законам распределений, носящим непрерывный характер. Поэтому категорически неверными является выбор целочисленных типов данных для моментов и интервалов времени, и тем более инкремент модельного времени с единичным шагом. Нужно реализовать именно переход от события к событию, как это сделано в GPSS и других проблемно-ориентированных системах. Для упрощения можно ограничиться использованием единственного потока случайных чисел для генерации всех необходимых случайных величин. Результатом работы программы должен быть лог-файл, содержащий записи типа:

- «В момент времени 12.345 транзакт с идентификатором 1 вошёл в модель»,
- «В момент времени 123.456 транзакт с идентификатором 123 встал в очередь 1»,
- «В момент времени 234.567 транзакт с идентификатором 234 занял устройство 2»,
- «В момент времени 345.678 транзакт с идентификатором 345 освободил устройство 1»,
- «В момент времени 456.789 транзакт с идентификатором 456 вышел из модели».

Входные и выходные данные

По итогам выполнение программы формируется файл, название которого может быть задано первым аргументом командной строки или изменено в коде программы, содержащий в себе информацию о поведение транзактов в модели.

Выходные данные:

```
Время запуска модели:2022-01-14 13:45:31.317987
                        Время          ID                Действие
В момент времени    |0.000       | транзакт с ID1        | вошел в очередь номер1
В момент времени    |0.000       | транзакт с ID1        | зашел в устройство номер1выйдет в  22.249
В момент времени    |21.402      | транзакт с ID2        | вошел в очередь номер2
В момент времени    |21.402      | транзакт с ID2        | зашел в устройство номер2выйдет в  42.316
В момент времени    |22.249      | транзакт с ID1        | вышел из устройства номер1
В момент времени    |35.934      | транзакт с ID3        | вошел в очередь номер1
В момент времени    |35.934      | транзакт с ID3        | зашел в устройство номер1выйдет в  63.321
В момент времени    |42.316      | транзакт с ID2        | вышел из устройства номер2
В момент времени    |44.370      | транзакт с ID4        | вошел в очередь номер2
В момент времени    |44.370      | транзакт с ID4        | зашел в устройство номер2выйдет в  69.565
В момент времени    |63.321      | транзакт с ID3        | вышел из устройства номер1
В момент времени    |67.536      | транзакт с ID5        | вошел в очередь номер1
В момент времени    |67.536      | транзакт с ID5        | зашел в устройство номер1выйдет в  79.497
В момент времени    |69.565      | транзакт с ID4        | вышел из устройства номер2
В момент времени    |79.497      | транзакт с ID5        | вышел из устройства номер1
В момент времени    |81.048      | транзакт с ID6        | вошел в очередь номер1
В момент времени    |81.048      | транзакт с ID6        | зашел в устройство номер1выйдет в  92.468
В момент времени    |92.468      | транзакт с ID6        | вышел из устройства номер1
В момент времени    |99.309      | транзакт с ID7        | вошел в очередь номер1
В момент времени    |99.309      | транзакт с ID7        | зашел в устройство номер1выйдет в  126.614
В момент времени    |116.143     | транзакт с ID8        | вошел в очередь номер2
В момент времени    |116.143     | транзакт с ID8        | зашел в устройство номер2выйдет в  131.863
В момент времени    |126.614     | транзакт с ID7        | вышел из устройства номер1
В момент времени    |131.863     | транзакт с ID8        | вышел из устройства номер2
В момент времени    |136.908     | транзакт с ID9        | вошел в очередь номер1
В момент времени    |136.908     | транзакт с ID9        | зашел в устройство номер1выйдет в  163.502
В момент времени    |144.672     | транзакт с ID10       | вошел в очередь номер2
В момент времени    |144.672     | транзакт с ID10       | зашел в устройство номер2выйдет в  154.273
В момент времени    |154.273     | транзакт с ID10       | вышел из устройства номер2
В момент времени    |160.098     | транзакт с ID11       | вошел в очередь номер2
В момент времени    |160.098     | транзакт с ID11       | зашел в устройство номер2выйдет в  169.893
В момент времени    |163.502     | транзакт с ID9        | вышел из устройства номер1
В момент времени    |169.893     | транзакт с ID11       | вышел из устройства номер2
В момент времени    |170.141     | транзакт с ID12       | вошел в очередь номер1
В момент времени    |170.141     | транзакт с ID12       | зашел в устройство номер1выйдет в  184.675
В момент времени    |184.675     | транзакт с ID12       | вышел из устройства номер1
В момент времени    |187.387     | транзакт с ID13       | вошел в очередь номер1
В момент времени    |187.387     | транзакт с ID13       | зашел в устройство номер1выйдет в  209.985
В момент времени    |209.985     | транзакт с ID13       | вышел из устройства номер1
В момент времени    |211.986     | транзакт с ID14       | вошел в очередь номер1
В момент времени    |211.986     | транзакт с ID14       | зашел в устройство номер1выйдет в  222.124
В момент времени    |212.211     | транзакт с ID15       | вошел в очередь номер2
В момент времени    |212.211     | транзакт с ID15       | зашел в устройство номер2выйдет в  226.792
В момент времени    |221.773     | транзакт с ID16       | вошел в очередь номер1
В момент времени    |222.124     | транзакт с ID16       | зашел в устройство номер1выйдет в  238.826
В момент времени    |222.124     | транзакт с ID14       | вышел из устройства номер1
```

Рисунок 1. Лог-файл

Алгоритмы

В программе предусмотрены 2 цепи: цепь текущих событий (CEC) и цепь будущих событий (FEC), которые будут заполняться и изменяться в будущем. Изначально в цепи будущих событий будут находиться транзакты, имеющие только время генерации. Программа последовательно выполняет 3 этапа (или фазы):

- 1ый этап – фаза ввода: здесь транзакты добавляются в FEC и ожидают своей очереди
- 2ой этап – фаза распределения: здесь транзакты с минимальным временем перемещаются из FEC в CEC и указывается соответствующая очередь.
- 3ый этап – фаза просмотра: здесь транзакты либо меняют свою позицию в модели (перемещается по очереди, переходит на обслуживающее устройство), либо выходит из модели.

Приложение 1. Содержание файлов исходного кода

Pract3.py

```
from Transact import *
#from prt import *

from Queue_h import *
from Device import *
from copy import *
from datetime import datetime
def funcSort(x):
    #M = [x.get_t_e]
    return int(1000*max(x.get_T_e(), x.get_T_g()))
def main():
    print("Ввещдите время моделирования:")

    osCommandString = "notepad.exe file.txt"
    current_datetime = datetime.now()
    f.write("Время запуска модели:" + str(current_datetime) + "\n")
    f.write("\t\t\tВремя" + "\t\tID" + "\t\t   Действие\n")
    #log = mylogger("res.txt", True)
    #print = log.printml()
    global_time = 0.0
    t_end = int(input())
    t = 0.0
    i = 1
    FEC = []
    while(t < t_end):

        FEC.append(Transact(i, t, -1.0, 0))
        i+=1
        t += Rand(0, 28)

    for j in range(0,len(FEC)):
        (FEC[j].get_T_g())

    j = 0
    D1 = Device(1, 10, 28)
    D2 = Device(2, 9, 28)
    Q1 = Queue_to_device(1, D1)
    Q2 = Queue_to_device(2, D2)
    while(j < len(FEC)):

        if(FEC[j].get_stat() == 0):
            global_time = FEC[j].get_T_g()
            if (global_time > t_end):
                f.write("End model")
                return 0
            if(Q1.get_lenght()+D1.get_stat() <= Q2.get_lenght() +
D2.get_stat()):
                FEC[j].set_stat(1)
                FEC[j].set_Q_number(1)
                out = "В момент времени   |" + str("%.3lf"%global_time) +
"\t| транзакт с ID" + str(FEC[j].get_id()) + "\t| вошел в очередь номер" +
"1" + "\n"
                f.write(out)
                Q1.add_transact_to_Q(FEC[j])
                #print(Q1.get_lenght())
                if(D1.get_stat() == 0 and Q1.get_lenght() == 1):
                    D1.set_T_g(global_time)
                    Q1.go_to_device()

                    #D1.add_transact_to_D(FEC[j])
```

```
                    Tr_in_D = copy(D1.go_out_of_D())
                    #k = 0
                   # while(Tr_in_D.get_T_e() > FEC[j].get_T_g() or
(Tr_in_D.get_T_e() > FEC[j].get_T_e() and FEC[j].get_stat()==2)):
                       # k+=1
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key = lambda trans: funcSort(trans))
                    #print(1)
                    #FEC.insert(k-1, Tr_in_D)

            else:
                FEC[j].set_stat(1)
                FEC[j].set_Q_number(2)
                out = "В момент времени   |" + str("%.3lf"%global_time) +
"\t| транзакт с ID" + str(FEC[j].get_id()) + "\t| вошел в очередь номер" +
'2' + "\n"
                f.write(out)
                Q2.add_transact_to_Q(FEC[j])
                #print(Q2.get_lenght())
                if (D2.get_stat() == 0 and Q2.get_lenght() == 1):
                    D2.set_T_g(global_time)
                    Q2.go_to_device()

                    #D2.add_transact_to_D(FEC[j])
                    Tr_in_D = copy(D2.go_out_of_D())
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key=lambda trans: funcSort(trans))
        elif(FEC[j].get_stat() == 1):
            global_time = FEC[j].get_T_g()
            if (global_time > t_end):
                f.write("End model")
                return 0
            if(FEC[j].set_Q_number()==1):
                if (D1.get_stat() == 0 and Q1.get_lenght() == 1):
                    D1.set_T_g(global_time)
                    Q1.go_to_device()

                    #D1.add_transact_to_D(FEC[j])
                    Tr_in_D = copy(D1.go_out_of_D())
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key=lambda trans: funcSort(trans))
            else:
                if (D2.get_stat() == 0 and Q2.get_lenght() == 1):
                    D2.set_T_g(global_time)
                    Q2.go_to_device()

                    #D2.add_transact_to_D(FEC[j])
                    Tr_in_D = copy(D2.go_out_of_D())
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key=lambda trans: funcSort(trans))
        elif(FEC[j].get_stat() == 2):
            global_time = FEC[j].get_T_e()
            if (global_time > t_end):
                f.write("End model")
                return 0
            if(FEC[j].get_Q_num() == 1):
                D1.set_stat(0)
                D1.set_T_g(global_time)
                if(Q1.get_lenght()>0):
                    D1.set_T_g(global_time)
                    Q1.go_to_device()

                    # D2.add_transact_to_D(FEC[j])
```

```
                    Tr_in_D = D1.go_out_of_D()
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key=lambda trans: funcSort(trans))
            elif(FEC[j].get_Q_num() == 2):
                D2.set_stat(0)
                D2.set_T_g(global_time)
                if (Q2.get_lenght() > 0):
                    D2.set_T_g(global_time)
                    Q2.go_to_device()

                    # D2.add_transact_to_D(FEC[j])
                    Tr_in_D = D2.go_out_of_D()
                    FEC.append(Tr_in_D)
                    FEC = sorted(FEC, key=lambda trans: funcSort(trans))
            FEC[j].end()
        j += 1
    f.close()
    #os.startfile("Результат_моделирования.txt")
    #os.system(osCommandString)
    #subprocess.call(['notepad.exe', "Результат_моделирования.txt"])
    return 0
if __name__ == '__main__':
    main()
```

Queue_h.

```
from Device import *
from copy import *
class Queue_to_device:
    def __init__(self, i, D):
        self.Q = []
        self.Id = i
        self.D = D
    def add_transact_to_Q(self, new_Tr):
        self.Q.append(new_Tr)
    def go_to_device(self):
        self.D.add_transact_to_D(self.Q[0])
        self.Q.pop(0)
    def get_lenght(self):
        return len(self.Q)
    def get_ID(self):
        return self.Id
```

Device.py

```
from Transact import *
from copy import *
class Device:
    def __init__(self, id, t1, t2):
        self.Tr = Transact(0, 0, 0, 0)
        self.Id = id
        self.t1 = t1
        self.t2 = t2
        self.T_g = 0.0
        self.stat = 0
    def add_transact_to_D(self, transact):

        self.Tr = copy(transact)
        self.stat = 1
        self.Tr.set_stat(2)
        out = "В момент времени   |" + str("%.3lf"%self.T_g) + "\t| транзакт
c ID" + str(self.Tr.get_id()) +  "\t| зашел в устройство номер" +
str(self.Id)
        f.write(out)
    def set_T_g(self, t):
```

```
        self.T_g = t
    def go_out_of_D(self):
        #self.stat = 0
        self.Tr.set_T_e(self.T_g + Rand(self.t1, self.t2))
        out = "выйдет в  " + str("%.3lf"%self.Tr.get_T_e()) + "\n"
        f.write(out)

        #self.Tr.set_stat()
        return self.Tr
    def get_stat(self):
        return self.stat
    def set_stat(self, s):
        self.stat = s
```

Transact.py

```
from random import *
from copy import *
def Rand(t1, t2):
    return uniform(t1, t2)
f = open('Результат_моделирования.txt', 'w')
class Transact:
    def __init__(self, id, T_g, T_e, stat):
        self.id = id
        self.T_g = T_g
        self.T_e = T_e
        self.stat = stat
        self.Q_number = 0
    def set_Q_number(self, n):
        self.Q_number = n
    def set_id(self, i):
        self.id = i
    def set_T_g(self, t):
        self.T_g = t
    def set_T_e(self, t):
        self.T_e = t
    def set_stat(self, s):
        self.stat = s
    def get_id(self):
        return self.id
    def get_T_g(self):
        return self.T_g
    def get_T_e(self):
        return self.T_e
    def get_stat(self):
        return self.stat
    def get_Q_num(self):
        return self.Q_number
    def end(self):
        out = "В момент времени   |" + str("%.3lf"%self.T_e) + "\t| транзакт
с ID" + str(self.id) + "\t| вышел из устройства номер" + str(self.Q_number) +
"\n"
        f.write(out)
```

Полный листинг программы можно увидеть по ссылке: https://github.com/NovokshanovE/NewRepo/tree/master