Государственное образовательное учреждение высшего профессионального образования «Московский Государственный Технический Университет имени Н. Э. Баумана»

ОТЧЕТ

По лабораторной работе №5 «Система массового обслуживания»

По курсу «Очередь заявок»

Студент: Белова А.А.

Группа: ИУ7-75

Преподаватель: Рудаков И.В.

Задание

В информационный центр приходят клиенты через интервал времени 10±3 минуты. Если все три имеющихся оператора заняты, клиенту отказывают в обслуживании. Операторы имеют разную производительность и могут обеспечивать обслуживание среднего запроса пользователя за 20±5; 40±10; 40±20. Клиенты стремятся занять свободного оператора с максимальной производительностью. Полученные запросы сдаются в накопитель. Откуда выбираются на обработку. На первый компьютер запросы от 1 и 2-ого операторов, на второй – запросы от 3-его. Время обработки запросов первым и 2-м компьютером равны соответственно 15 и 30 мин. Промоделировать процесс обработки 300 запросов.

Вероятность отказа $P = (C_{om\kappa})/(C_{om\kappa} + C_{obca})$

Реализация

```
op0 = Operator(op0_m, op0_d)
op1 = Operator(op0_m, op0_d)
op2 = Operator(op0 m, op0 d)
comp0 = Computer(comp0 m)
comp1 = Computer(comp0 m)
stuff = [client gen, op0, op1, op2, comp0, comp1]
current time = \overline{0}
lost = \overline{0}
success = 0
start = True
while client gen.max req > 0:
    # Find next event
    print(client_gen.max_req)
    current time, index = client gen.time to start, 0
    for i, device in enumerate(stuff):
        if 0 < device.time_to_start < current_time:</pre>
            current time = device.time to start
            index = i
    if index == 0:
        req = client gen.get from queue()
        client gen.gen new time()
        result = op0.add to queue(current time)
        if result:
            continue
        result = op1.add_to_queue(current_time)
        if result:
            continue
        result = op2.add to queue(current time)
        if result:
            continue
        lost += 1
    elif index == 1:
        res = op0.get from queue()
        if res == 0:
            continue
        res = comp0.add to queue(current time)
        if res == False:
            lost += 1
    elif index == 2:
        res = op1.get from queue()
        if res == 0:
            continue
        res = comp0.add_to_queue(current_time)
        if res == False:
             lost += 1
    elif index == 3:
        res = op2.get from queue()
        if res == 0:
            continue
        res = comp1.add_to_queue(current_time)
        if res == False:
            lost += 1
    elif index == 4:
        success += comp0.get from queue()
```

Интерфейс и результаты работы программы:

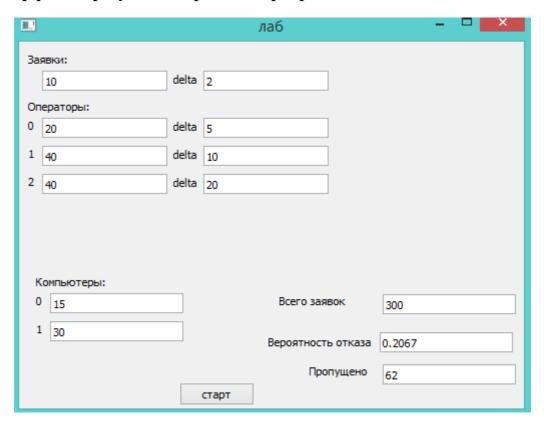


Рисунок 1. Пример работы программы