Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Моделирование

ОТЧЕТ

по лабораторной работе № 5

на тему

ПОСТРОЕНИЕ И ИССЛЕДОВАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ  
НЕПРЕРЫВНО-СТОХАСТИЧЕСКОЙ СМО

ВАРИАНТ № 4

Студент: П.В. Сякачёв

Проверила: Ю.О. Герман

МИНСК 2022

# 1. Цель работы

Изучить методы имитационного моделирования поведения непрерывно-стохастической СМО.

# 2. Задание

Произвести имитационное моделирование для системы с одним прибором. Интенсивность поступления заявок λ = 0.1 сек−1, интенсивность обслуживания заявок μ = 2сек-1. Закон распределения вероятностей времен поступления обозначен как F(t), времени обслуживания – как G(t).

По результатам моделирования найти – среднее время обслуживания, среднее время пребывания заявки в системе, среднее число заявок в системе, процент загрузки обслуживающего прибора (канала).

# 3. Ход работы

Для вычисления времени поступления заявок и длительности обслуживания, выразим t из формул F(t) и G(t):

Используя метод Лемера, сгенерируем несколько случайных чисел: 19, 23, 91, 47, 99, 83.

В соответствии, с законами распределения F(t) и G(t), вычислим несколько значений t времён поступления заявок и τ времён обслуживания заявок:

Теперь для наглядности на временных осях мы можем отложить значения времени поступления и времени обслуживания:

*t = 2.10 t = 26.17 t = 72.22 время поступления*

*время обслуживания*

t = 3.12 t = 27.76 t = 74.88

Заметим, что в интервале от 3.12 до 26.17 система простаивает – нет заявок, ничего не обслуживается. Кроме того, возможно появление очереди на обслуживание (в нашем примере очередь не успела сформироваться).

## 3.1 Написание программы

Для реализации имитационной модели СМО был написан класс **Queuing\_system**, вот несколько методов из него:

private void Init()

{

for (int i = 0; i < count; i++)

{

R[i] = new Request();

R[i].coming\_time = F\_Function(random.Lehmer\_Random());

R[i].processing\_time = G\_Function(random.Lehmer\_Random());

}

for (int i = 1; i < count; i++)

{

R[i].coming\_time += R[i - 1].coming\_time;

}

}

private int F\_Function(double F)

{

return (int)((Math.Log(1 - F) / (-l)) \* 100);

}

private int G\_Function(double G)

{

return (int)(m \* Math.Sqrt(-Math.Log(1 - G)) \* 100);

}

Изображенные выше методы необходимы для заполнения времён запросов и выполнения числами, вычисленными согласно законам F(t) и G(t) при помощи сгенерированных случайных чисел.

public void Calculate()

{

time = 0;

free\_time = 0;

int num = 0, requests = 0;

bool is\_free = true;

while (true)

{

request\_count += requests;

if (num < count && R[num].coming\_time == time)

{

requests++;

num++;

}

if (is\_free)

{

if (requests == 0)

free\_time++;

else

{

R[num - requests].starting\_time = time;

is\_free = false;

}

}

else

{

if (R[num - requests].starting\_time +

+ R[num - requests].processing\_time == time)

{

is\_free = true;

requests--;

time--;

}

}

if (num == count && requests == 0)

break;

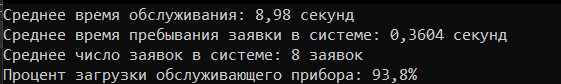
time++;

}

}

Приведённый выше метод – моделирование СМО, в которой имитируются приходящие запросы и их обработка.

Результат выполнения программы – вывод в консоль сведений о процессе моделирования:



# Вывод

В ходе лабораторной работы был а построена имитационная модель непрерывно-стохастической СМО и исследовано её поведение в зависимости от изменения формул законов F(t) и G(t).