**МИНОБРНАУКИ РОССИИ**

**ФГБОУ ВО «ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра вычислительной техники

Отчет по лабораторной работе № 4

«Моделирование непрерывных случайных воздействий»

по дисциплине «Основы моделирования систем»

Выполнил студент группы 220681:

Шевердин Д.О.

Проверил:

доц. Семенчев Е.А.

Тула 2020

# ЦЕЛЬ РАБОТЫ

Освоение методов получения реализаций непрерывной случайной величины на ЭВМ. Имитация случайных воздействий по заданному закону на ЭВМ.

# ЗАДАНИЕ

1. Составить схему алгоритма и программу для имитации случайной величины , имеющей экспоненциальное распределение.
2. Составить схему алгоритма и написать программу вычисления вероятности события

где – номер студента по списку группы (27), -случайная величина

1. Используя 10000 реализаций , вывести на печать 30 первых значений случайной величины, гистограмму распределения и значение частоты попадания в указанный выше интервал.
2. Сравнить эту частоту с теоретической вероятностью.

# ХОД РАБОТЫ

Для имитации экспоненциального распределения воспользуемся мультипликативным конгруэнтным методом:

где некоторые константы.

Алгоритм получения числа с помощью мультипликативного конгруэнтного метода (от 0 до 1):

1. Задать константы и начальное значение
2. Вычислить
3. Вернуть

Алгоритм получения экспоненциально распределенного числа:

1. Задать интенсивность
2. Сгенерировать число из
3. Вернуть .

Вычислить вероятность события можно по следующей формуле:

# ТЕСТИРОВАНИЕ

На рисунках 1, 2 представлены результаты работы программы. Код программы представлен в «Приложении».

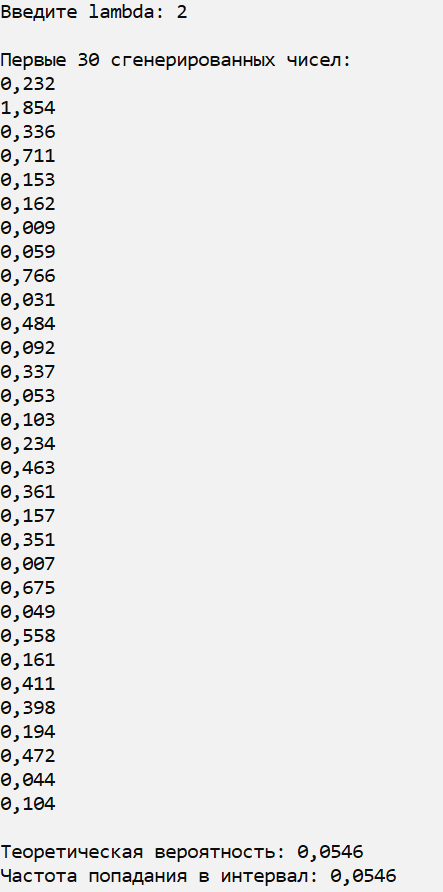
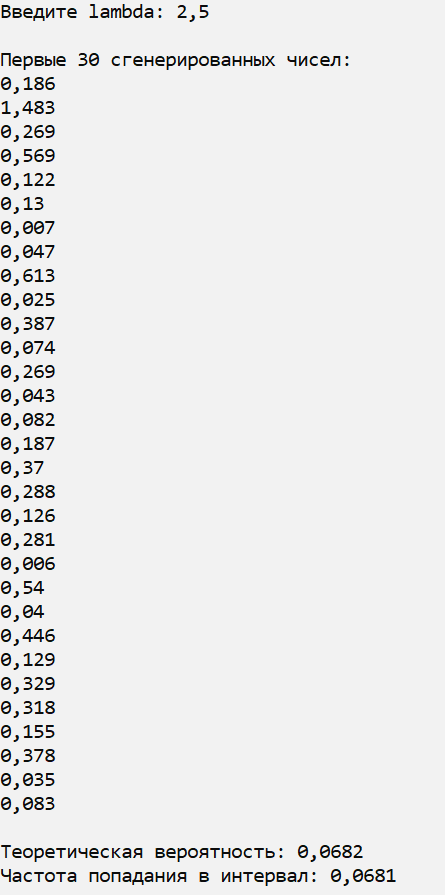
 

Рисунок 1 – Тестирование программы Рисунок 2 – Тестирование программы

Экспоненциальное распределение для первого теста имеет следующую гистограмму (Рисунок 3).



Рисунок 3 – Гистограмма распределения

# ВЫВОДЫ

Были освоены методы получения непрерывной случайной величины на ЭВМ. По полученным данным можно сделать вывод, что при увеличении количества испытаний частота стремится к теоретически вычисленной вероятности.

# Приложение

## Testing.cs

using System;

namespace Random

{

public class Testing

{

private double l;

private double E { get; set; }

private double LeftBound { get; set; }

private double RightBound { get; set; }

public Testing(double l, double k = 27.00)

{

this.l = l;

E = 1/l;

LeftBound = E - (1 / k);

RightBound = E + (1 / k);

}

public double TheoreticalProbability()

{

if (LeftBound + RightBound < 0)

{

return 0;

}

else if (LeftBound < 0)

{

return 1-Math.Pow(Math.E, -1.00 \* l \* RightBound);

}

else

return Math.Pow(Math.E, -1.00 \* l \* LeftBound) - Math.Pow(Math.E, -1.00 \* l \* RightBound);

}

public bool BetweenBoundaries(double num)

=> LeftBound <= num && num <= RightBound;

}

}

## Program.cs

using System;

namespace Random

{

class Program

{

static void Main()

{

try

{

Console.Write("Введите lambda: ");

var l = double.Parse(Console.ReadLine());

int count = 0;

double amount = 10000;

var random = new Multiplicative();

var testing = new Testing(l);

Console.WriteLine("\nПервые 30 сгенерированных чисел:");

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

{

double num = random.Next(l);

// if (i <= 30)

Console.WriteLine(Math.Round(num, 3));

if (testing.BetweenBoundaries(num))

count++;

}

Console.WriteLine($"\nТеоретическая вероятность: {Math.Round(testing.TheoreticalProbability(), 4)}");

Console.WriteLine($"Частота попадания в интервал: {Math.Round(count / amount, 4)}");

}

catch (Exception)

{

Console.WriteLine("Введите данные корректно!");

}

}

}

}

## Multiplicative.cs

using System;

namespace Random

{

class Multiplicative

{

private int a;

private int m;

private double Xn;

public Multiplicative(int x0 = 11, int t = 43, int m = 6075)

{

a = 8 \* t + 3;

this.m = m;

Xn = x0;

}

public double GenerateNumber()

{

Xn = (a \* Xn) % m;

return Xn / m;

}

public double Next(double l)

=> (-1.00/l)\*Math.Log(GenerateNumber());

}