Министерство образования Республики Беларусь

Министерство образования и науки Российской Федерации

Межгосударственное образовательное учреждение высшего образования

Белорусско-Российский университет

кафедра “Автоматизированные системы управления”

Отчёт

По лабораторной работе №1

По дисциплине “Имитационное моделирование систем”

На тему “Моделирование потоков заявок в системах массового обслуживания”

Выполнил: ст. гр. ПИР-211

Лобановский И. В.

Проверила: Борчик Е. М.

Могилев, 2024

**Цель работы:** реализация методов и алгоритмов моделирования реализаций псевдослучайных числовых последовательностей с заданным законом распределения.

**Задание:**





**Формулы:**

Формирование псевдослучайных числовых последовательностей z(i) в интервале [a; b]:

(1)

z(i) – псевдослучайное число

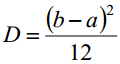
a – начало диапазона псевдослучайных чисел

b – конец диапазона псевдослучайных чисел

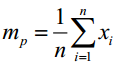
x(i) – случайное число в диапазоне [0;1]

Математическое ожидание:

(2)

Дисперсия:

(3)

Оценка математического ожидания:

(4)

n – размер набора псевдослучайных чисел

*xi* – *i*-е значение псевдослучайной числовой последовательности.

Математическое ожидание показывает среднее значение всех чисел в наборе. Чем больше n — тем точнее значение.

Оно рассчитывается как сумма всех значений последовательности, делённая на количество этих значений.

Закон равномерного распределения псевдослучайных чисел [0,7; 1,8].

Оценка дисперсии:

(5)

Неравенства для определения точности оценки:

(6)

 (7)

ε – заданная точность оценки

Число интервалов для формирования гистограммы:

(8)

k – число интервалов

**Блок-схема программы:**

Метод main:



Рисунок 1 – Блок-схема работы метода main

Метод CalculateExperimentalM:



Рисунок 2 – Блок-схема работы метода CalculateExperimentalM

Метод CalculateExperimentalD:



Рисунок 3 – Блок-схема работы метода CalculateExperimentalD

**Код программы:**

namespace ConsoleApp1

{

public static class Program

{

public static void Main(string[] args)

{

// Определяем переменные для параметров a, b, точности и длины последовательности

double a, b, precision;

int sequenceLength;

// Запрашиваем у пользователя параметр a

Console.Write("Параметр a:");

// Проверяем корректность ввода и продолжаем запрашивать, пока не будет введено корректное значение

while (!double.TryParse(Console.ReadLine(), out a))

{

Console.Write("Ошибка. Повторите ввод: ");

}

// Запрашиваем у пользователя параметр b

Console.Write("Параметр b:");

// Проверяем корректность ввода, причем значение b должно быть больше или равно a

while (!double.TryParse(Console.ReadLine(), out b) || b < a)

{

Console.Write("Ошибка. Повторите ввод: ");

}

// Запрашиваем у пользователя точность

Console.Write("Точность:");

// Проверяем корректность ввода точности

while (!double.TryParse(Console.ReadLine(), out precision))

{

Console.Write("Ошибка. Повторите ввод: ");

}

// Запрашиваем у пользователя количество элементов последовательности

Console.Write("Количество элементов массива:");

// Проверяем корректность ввода, при этом количество должно быть положительным

while (!int.TryParse(Console.ReadLine(), out sequenceLength) || sequenceLength <= 0)

{

Console.Write("Ошибка. Повторите ввод: ");

}

// Создаем массив для хранения элементов последовательности

double[] sequence = new double[sequenceLength];

// Теоретическое математическое ожидание и дисперсия

double theoreticalM = (a + b) / 2; // Теоретическое мат. ожидание

double theoreticalD = Math.Pow(b - a, 2) / 12; // Теоретическая дисперсия

double experimentalM = 0; // Экспериментальное мат. ожидание

double experimentalD = 0; // Экспериментальная дисперсия

int experimentNumber = 0; // Номер эксперимента

bool isResultValid = false; // Флаг для проверки достижения точности

Random random = new(); // Инициализация генератора случайных чисел

// Пока точность не достигнута, продолжаем проводить эксперименты

while (!isResultValid)

{

experimentNumber++;

// Печатаем номер текущего эксперимента

Console.WriteLine($"Номер эксперимента: {experimentNumber}");

// Заполняем массив случайными числами в диапазоне [a, b]

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

{

sequence[i] = (b - a) \* random.NextDouble() + a;

}

// Вычисляем экспериментальное мат. ожидание

experimentalM = CalculateExperimentalM(sequence);

// Вычисляем экспериментальную дисперсию

experimentalD = CalculateExperimentalD(sequence, experimentalM);

// Проверяем, достигнута ли заданная точность для мат. ожидания и дисперсии

isResultValid = (Math.Abs(theoreticalM - experimentalM) < precision) && (Math.Abs(theoreticalD - experimentalD) < precision);

// Если точность не достигнута, сообщаем об этом и повторяем эксперимент

if (!isResultValid)

{

Console.WriteLine("Точность не достигнута. Пересчитываем.\n");

}

}

// Сообщаем, что точность достигнута

Console.WriteLine("Точность достигнута.\n");

// Открываем файл для записи результатов

using var streamWriter = new StreamWriter("z.txt", false);

// Печатаем и записываем элементы последовательности в файл

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

{

Console.WriteLine($"z({i + 1})=\t{sequence[i]}");

streamWriter.WriteLine($"{i + 1}\t{sequence[i]}");

}

// Печатаем теоретическое и экспериментальное мат. ожидание и дисперсию

Console.WriteLine($"Теоретическое мат. ожидание: {theoreticalM}" +

$"\nТеоретическая дисперсия: {theoreticalD}\n" +

$"Экспериментальное мат. ожидание: {experimentalM}" +

$"\nЭкспериментальная дисперсия: {experimentalD}");

}

// Метод для вычисления экспериментального мат. ожидания

public static double CalculateExperimentalM(double[] sequence)

{

double experimentalM = 0;

// Суммируем все элементы последовательности

for (int i = 0; i < sequence.Length; i++)

{

experimentalM += sequence[i];

}

// Делим сумму на количество элементов, чтобы получить среднее значение

experimentalM /= sequence.Length;

return experimentalM;

}

// Метод для вычисления экспериментальной дисперсии

public static double CalculateExperimentalD(double[] sequence, double m)

{

double experimentalD = 0;

// Суммируем квадраты разностей каждого элемента последовательности и мат. ожидания

for (int i = 0; i < sequence.Length; i++)

{

experimentalD += Math.Pow(sequence[i] - m, 2);

}

// Делим сумму на количество элементов, чтобы получить дисперсию

experimentalD /= sequence.Length;

return experimentalD;

}

}

}

**Скриншот работы программы:**

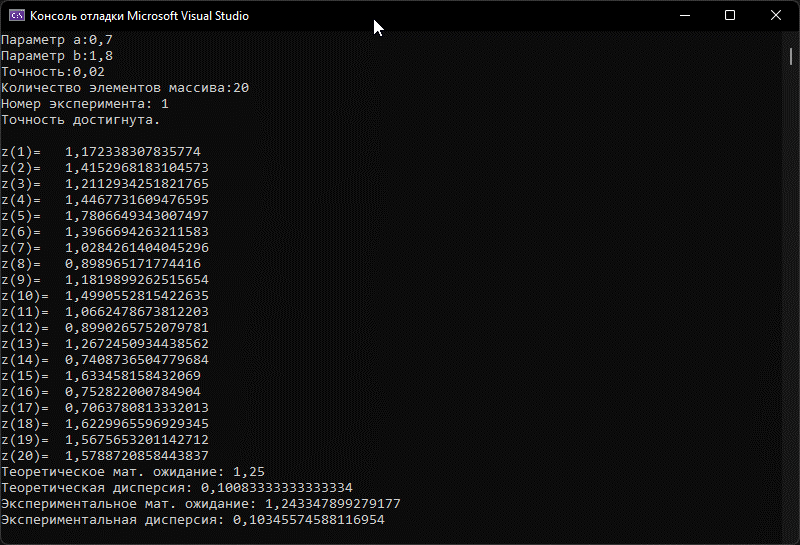


Рисунок 4 – Скриншот консоли с результатом работы программы

**Построение гистограммы в Excel:**

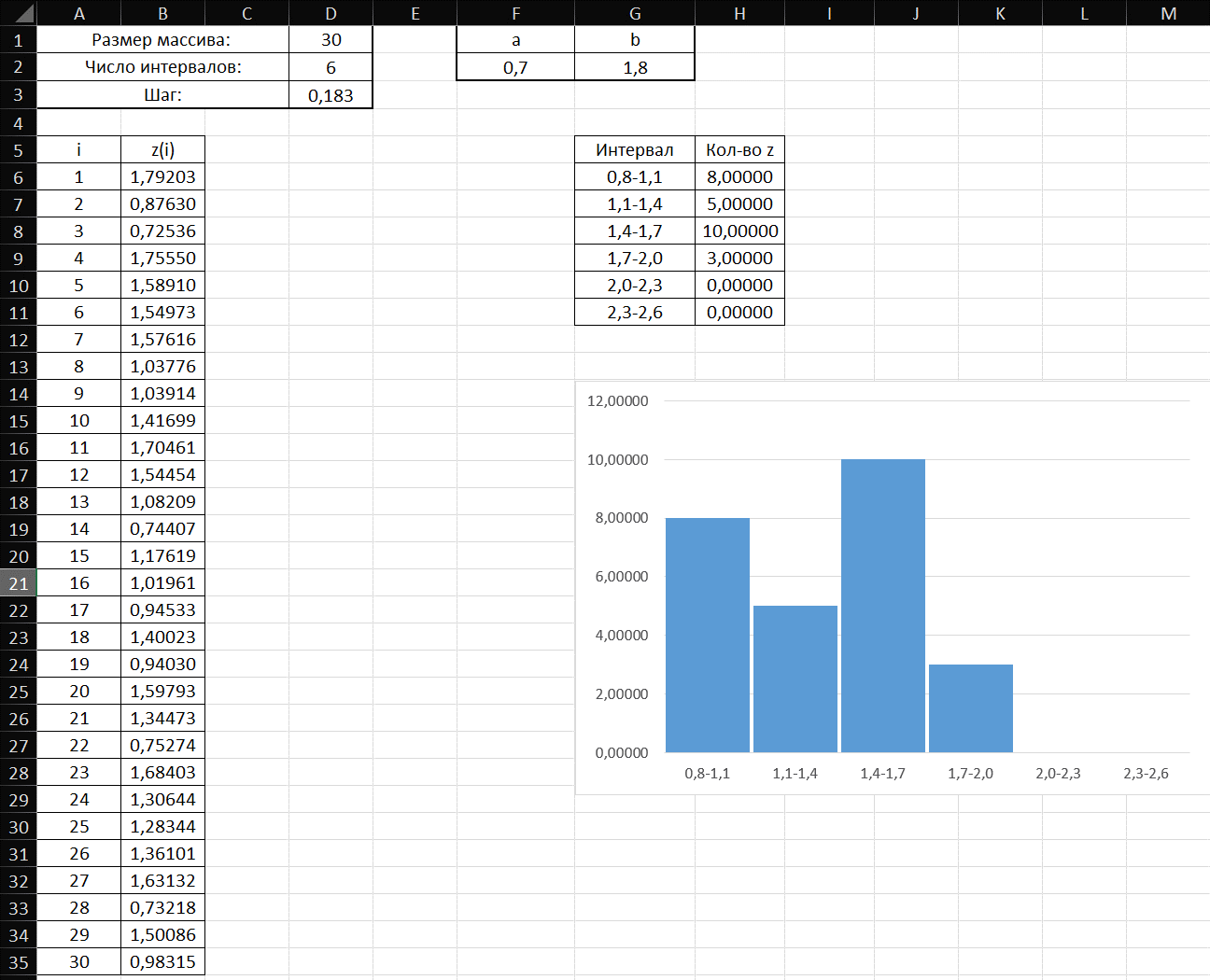


Рисунок 6 – Построение гистограммы распределения для 30 чисел

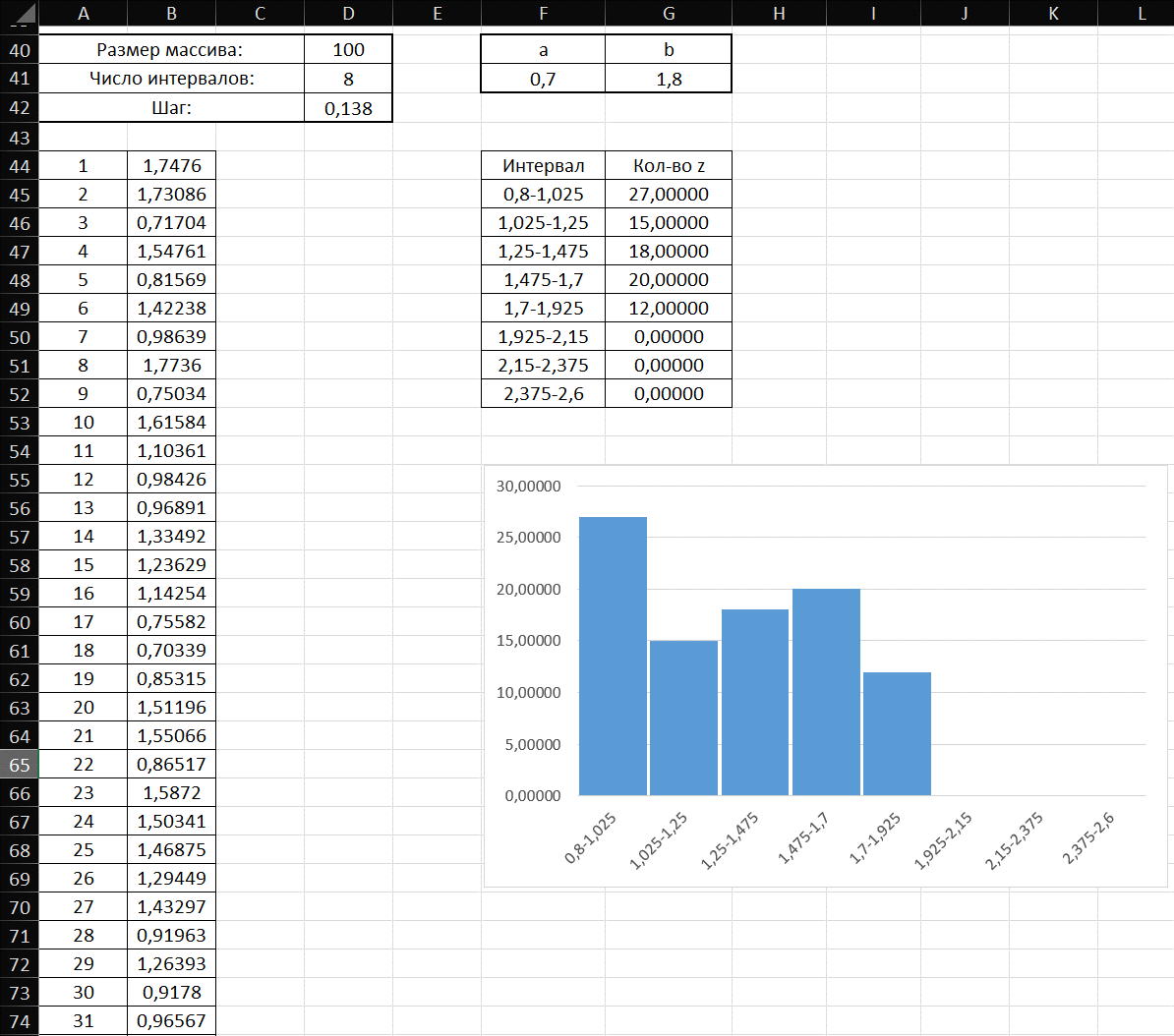


Рисунок 7 – Построение гистограммы распределения для 100 чисел

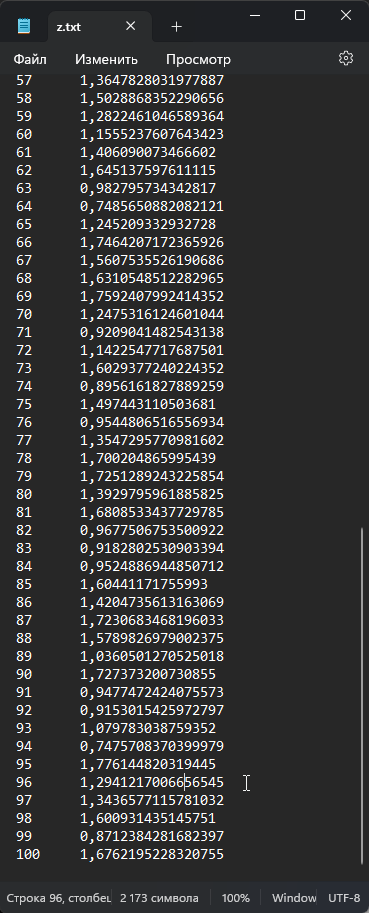


Рисунок 8 – Массив из 100 псевдослучайных чисел, распределенных по равномерному закону в диапазоне от 0,7 до 1,8

## Вывод:

Был проведён расчётный эксперимент математической модели, основанной на алгоритме формирования псевдослучайной математической последовательности чисел равномерного закона распределения в диапазоне [0,7; 1,8] с исходным значением точности 0,02. Было выполнено 2 эксперимента, в ходе которых были получены:

* псевдослучайная математическая последовательность чисел размера 30, оценка математического ожидания 1,23 и оценка дисперсии 0,110
* псевдослучайная математическая последовательность чисел размера 100, оценка математического ожидания 1,23 и оценка дисперсии 0,119

В обоих экспериментах получены величины, соответствующие теоретическим значениям с учётом заданной точности.

## Результаты тестирования:





