|  |  |  |
| --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | |
| Факультет компьютерных систем и сетей | | |
| Кафедра электронных вычислительных средств | | |
| Отчет по лабораторной работе №1  по курсу  «Системный анализ и машинное моделирование» | | |
| Выполнил  студент гр. 751003  Стубеда В.Д. |  | Проверил  Мельник Н. И. |
| Минск 2020 | | |

# Цель работы

Изучить формирование равномерно распределенных случайных величин.

# ЗаданиЯ

1. Построить (написать программу) генератор последовательности равномерно распределенных случайных чисел на основе алгоритма Лемера. Предусмотреть при этом возможность ввода параметров a, R0, m с клавиатуры.
2. Для полученной выборки чисел построить гистограмму (20 интервалов), рассчитать значения оценок для математического ожидания , дисперсии () и среднего квадратичного отклонении ( ).
3. Оценить равномерность последовательности по косвенным признакам;
4. Найти длину периода и участка апериодичности. Варьируя значениями параметров a, R0, m, добиться длины периода не менее 50000 чисел.

# Ход работы

Задание №1.

Ниже приведен код генератора последовательности равномерно распределенных случайных чисел на основе алгоритма Лемера:

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

{

r = (r \* a) % m;

x.Add((double) r/m);

}

Задание №2.

Для построения гистограммы находим мнимальное и макисмальное значение последовательности.

Затем высчитываем диапозон варьирования:

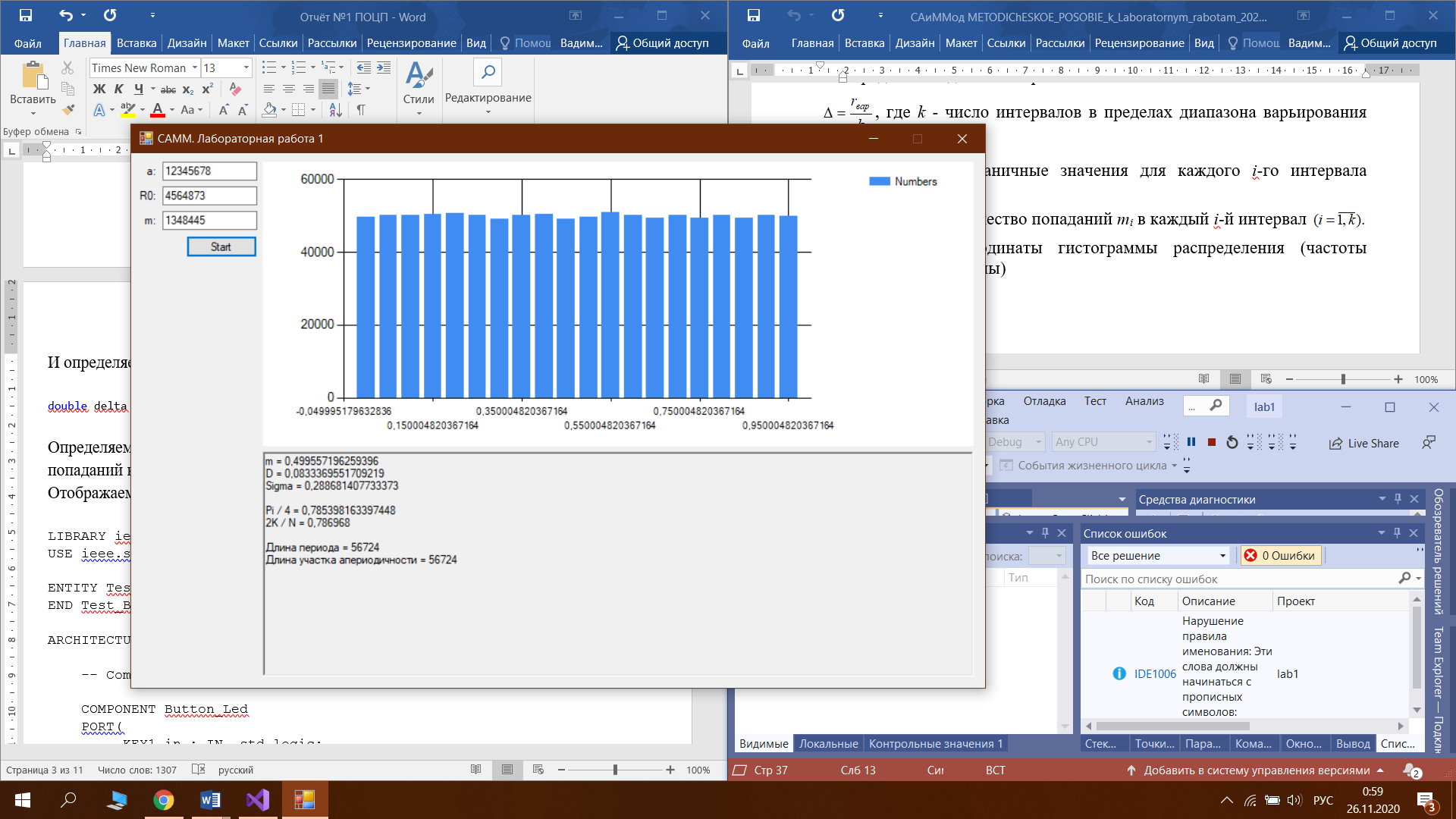
var rVar = xMax - xMin;

И определяем длину интервала:

double delta = rVar/k;

Определяем граничные значения для каждого интервала и фиксируем количество попаданий в них;

Отображаем полученные данные на гистограмме:



Высчитываем математическое ожидание:

var sum = x.Sum();

var matOzh = sum/x.Count;

Высчитываем дисперсию:

var disp = x.Sum(t => Math.Pow(t - matOzh, 2)) / x.Count;

Высчитываем среднее квадратичное отклонение:

Math.Sqrt(disp);

Задание №3.

Ниже приведен код оценки равномерности последовательности по косвенным признакам:

var innerCount = 0;

for (int i = 0; i + 1 < x.Count; i += 2)

{

if (x[i]\*x[i] + x[i + 1]\*x[i + 1] < 1)

{

innerCount++;

}

}

innerCount = innerCount \* 2 / x.Count;

Задание №4.

Ниже приведен код поиска периода:

var xv = x.Last();

var pos = new int[2];

bool isFirst = false;

const double epsilon = double.Epsilon;

for(int i=0; i<x.Count; i++)

{

if (Math.Abs(x[i] - xv) < epsilon)

{

if (isFirst)

{

pos[1] = i;

break;

}

isFirst = true;

pos[0] = i;

}

}

var period = pos[1] - pos[0];

Ниже приведен код поиска участка апериодичности:

var aperiod = 0;

for (int i = 0; i + period < x.Count; i++)

{

if (Math.Abs(x[i] - x[i + period]) < epsilon)

{

aperiod = i;

break;

}

}

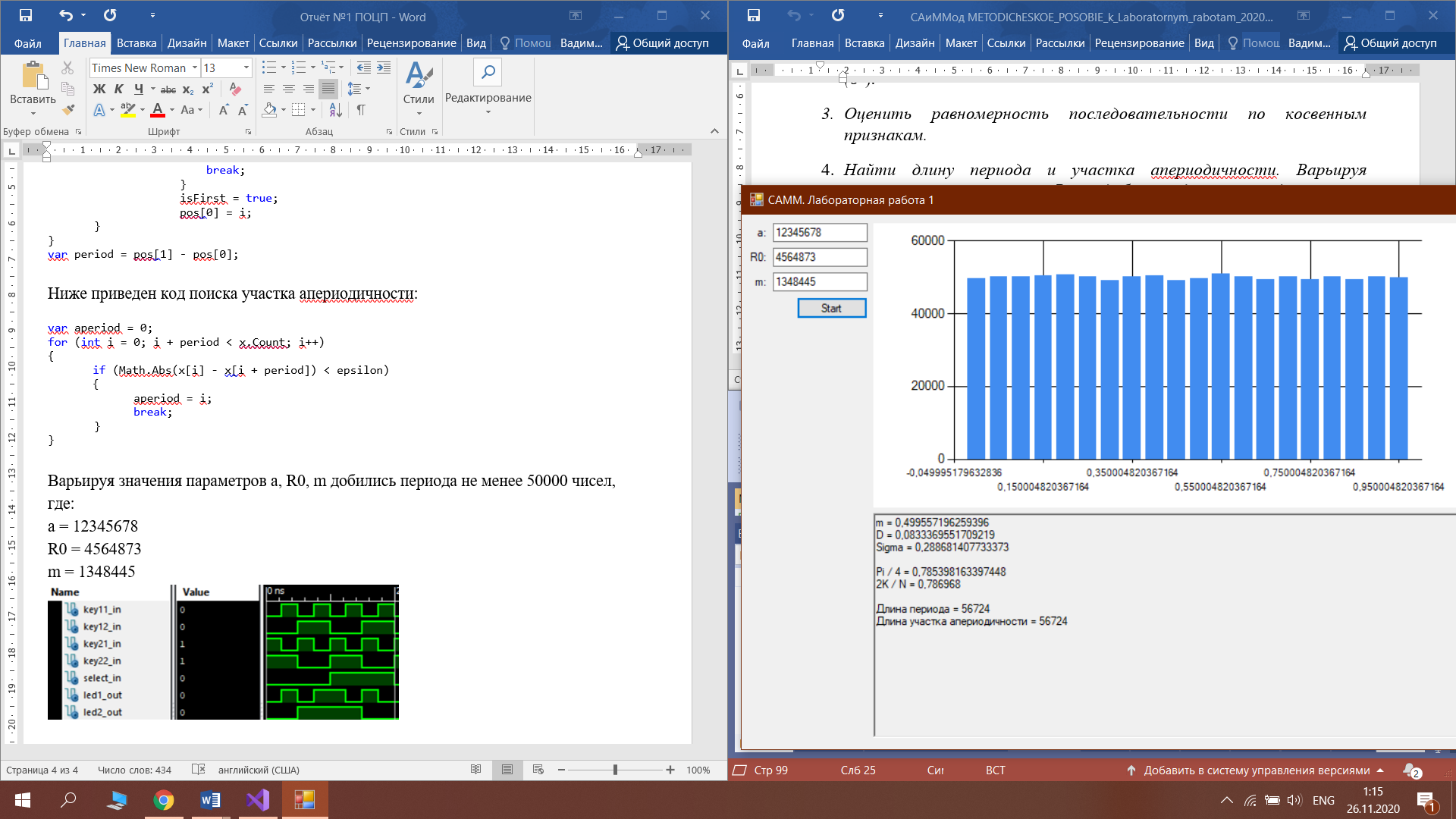
Варьируя значения параметров a, R0, m добились периода не менее 50000 чисел,

где:

a = 12345678

R0 = 4564873

m = 1348445



# Вывод

В ходе выполнения данной лабораторной работы мною было изучено формирование равномерно распределенных случайных числе с помощью алгоритма Лемера, поиск оценок и проверка качества последовательностей РРСЧ.