**ОТЧЕТ**

по лабораторной работе №1

Моделирование потоков заявок в системах массового обслуживания

Выполнил студент группы АСОИ-122

Силков Е.В.

Проверил: Кушнер А.В.

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

***Порядок выполнения работы***

1. Изучить настоящие методические указания.  
2. Выбрать исходные данные для выполнения работы по таблице 2  
согласно варианту, указанному преподавателем.  
3. Разработать математическую модель и алгоритм формирования  
заданных псевдослучайных числовых последовательностей и вычисления их  
основных статистических характеристик.  
4. Разработать процедуру реализации алгоритма и провести расчетный  
эксперимент.  
5. Оформить отчет.



**Код программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Controls.DataVisualization;

using ZedGraph;

namespace IMS\_01

{

/// <summary>

/// Логика взаимодействия для Lab\_1.xaml

/// </summary>

public partial class Lab\_1 : Page

{

ZedGraphControl zedGraphControl1;

public class Data

{

public int I { get; }

public double X { get; }

public double Z { get; }

public double F { get; }

public Data(int i, double x, double z, double f)

{

I = i;

X = x;

Z = z;

F = f;

}

}

public Lab\_1()

{

InitializeComponent();

zedGraphControl1 = new ZedGraphControl();

}

private void calcButton\_Click(object sender, RoutedEventArgs e)

{

try

{

double a = Convert.ToDouble(tbA.Text);

double b = Convert.ToDouble(tbB.Text);

double p = Convert.ToDouble(tbP.Text);

double c = Convert.ToDouble(tbC.Text);

int n = Convert.ToInt32(tbN.Text);

double[] ax = new double[n];

double[] ay = new double[n];

if (n < 1) throw new Exception("Количество меньше нуля");

double m = (a + b) / 2, d = (b - a) \* (b - a) / 12;

double mp = 0, dp = 0;

double sigma;

int iter = 0;

List<Data> data = new List<Data>();

Random rnd = new Random();

do

{

mp = dp = 0;

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

{

int integer = rnd.Next(0, 101);

double x = (double)integer / 100;

double z = ax[i] = (b - a) \* x + a;

double f = 0;

if (z <= b && z >= a) f = Convert.ToDouble(string.Format("{0:0.000}", 1.0 / (b - a)));

else f = 0;

data.Add(new Data(i + 1, x, z, f));

mp += z; dp += (z - m) \* (z - m);

ay[i] = f;

}

sigma = Math.Sqrt(dp);

mp /= n; dp /= n;

iter++;

}

while (Math.Abs(d - dp) > c && Math.Abs(m - mp) > c);

textBlockN.Text = "Количество итераций: " + iter.ToString();

textBlockM.Text = "m теоретическая" + m.ToString();

textBlockD.Text = "D теоретическая" + d.ToString();

textBlockMp.Text = "m практическая" + mp.ToString();

textBlockDp.Text = "D практическая" + dp.ToString();

dataGrid.ItemsSource = data;

windowsFormsHost.Child = zedGraphControl1;

zedGraphControl1.Width = zedGraphControl1.Height = 400;

zedGraphControl1.GraphPane.CurveList.Clear();

zedGraphControl1.GraphPane.AddCurve("",

ax, ay, System.Drawing.Color.Red, SymbolType.None);

zedGraphControl1.AxisChange();

zedGraphControl1.Invalidate();

}

catch (Exception ex)

{

MessageBox.Show("Возникла ошибка:\n" + ex.Message);

}

}

}

}

Блок-схема программы

f = 0;

f = Convert.ToDouble(string.Format("{0:0.000}", 1.0 / (b - a))) ;

z <= b && z >= a

int integer = rnd.Next(0, 101); double x = (double)integer / 100; double z = ax[i] = (b - a) \* x + a; double f = 0;

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

do … while (Math.Abs(d - dp) > c && Math.Abs(m - mp) > c);

Random rnd = new Random();

m = (a + b) / 2, d = (b - a) \* (b - a) / 12;

Ввод данных

Вывод данных, построение графика

sigma = Math.Sqrt(dp);

mp /= n; dp /= n;

iter++;

data.Add(new Data(i + 1, x, z, f));

mp += z; dp += (z - m) \* (z - m);

ay[i] = f;

Результат программы







