МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ОДЕССКИЙ НАЦИОНАЛЬНЫЙ МОРСКОЙ УНИВЕРСИТЕТ

Кафедра «Информационные технологии»

Расчетно-графическое задание

по ТПР

на тему «Обработка и анализ результатов коллективных решений»

Выполнила:

студента 3к. 2гр. КСФ

Шипкова В.И.

Проверил:

Рудниченко Н.Д.

Одесса 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4

ПРАКТИЧЕСКАЯ ЧАСТЬ 7

ВЫВОДЫ 10

ЛИТЕРАТУРА 11

ЛИСТИНГ 12

ВВЕДЕНИЕ

Цель работы– определить коллективную оценку объектов с точки зрения их воздействия на некоторую цель или показатель.

Задачи:

1. Изучить теоретическую часть к данной теме;

2. Модифицировать программу, разработанную в первой лабораторной работе, а именно, что бы она запрашивала также число критериев, строила матрицу парных сравнений критериев относительно общей цели и альтернатив относительно каждого критерия;

3. Рассчитать собственные вектора для всех полученных матриц по выражению.

4. Использовать модуль работы с матрицами и векторами, который должен был быть разработан в работе №1.

5. Используя полученные собственные вектора, синтезировать приоритеты на иерархии и получить вектор приоритетов альтернатив относительно цели.

6. Ответить на контрольные вопросы.

7. Составить отчет по проделанной работе.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Методы обработки групповых мнений и принятия коллективных решений есть общее определение для разнообразных методов построения некоторого коллективного мнения по совокупности индивидуальных мнений – голосование, проведение опроса, или проведение экспертизы, т.е. измерение группой людей свойств определенных объектов или признаков объектов, если эти признаки носят субъективный характер или нет измерительных приборов.

Совокупность опрашиваемых участников процесса называются референтнойили экспертной группой, а для оцениваемых объектов принято использовать термин факторы.

Принятие решений с помощью экспертов включает следующие типовые этапы:

1. формирование референтной группы;
2. разработка анкеты для опроса;
3. сбор данных (голосование);
4. обработка и получение коллективного решения (т.е. группового предпочтения);
5. проверка результатов на обоснованность;
6. обсуждение результатов и зацикливание (коррекция анкеты, переформирова ние группы т.д.);
7. оформление решения.
   1. Методы сбора данных

Результатом экспертизы является так называемая совокупность групповых предпочтений, к изображению которой имеются два альтернативных подхода:

ординальное(порядковое) предпочтение – здесь объекты упорядочиваются по рангу или месту в общем ряду совокупности;

кардинальное(числовое) представление – здесь каждому фактору ставится в соответствие вес или число.

Наибольшее распространение получили следующие три метода:

* ранжирование;
* нормирование;
* парные сравнения.

Рассмотрим метод ранжирования сбора для общей ситуации: наличия n факторов Ф1, Ф2, …, Ф*n* и *m* экспертов Э1, Э2, …, Э*m*.

Ранжирование (ординальный подход)

Каждый эксперт упорядочивает факторы по убыванию важности, присваивая им числа 1,2,…,*n*. Если возникает ситуация, когда эксперт не может

различить по важности два или более факторов он может ставить их рядом и приписывать им одинаковые, так называемые связанные ранги. Если, например, не удается различить факторы, занимающие места с *р* по *q*, то им всем приписывается ранг:

*R*св 

*p*  ( *p*  1)  ...  (*q*  1)  *q*

*q*  *p*  1 .

Результатом опроса в случае метода ранжирования является матрица

*R*=(*Rij*),

 *R*11



 *R*1 *j*

 *R*1*m* 

     



 *R*



*R*  *R*

 *i*1

 *Rij*



*im*  .

  

 *R*  *R*

  

 *R* 

 *n*1 *nj nm* 

* 1. Обработка результатов опроса

Основным результатом обработки является вектор весов факторов:

 *w*1 



*w*  



  ,

 *w* 

который строится в два этапа.

 *n* 

1. Построение системы векторов или матрицы *Х*=(*хij*)

 *х*11



 *х*1 *j*

 *х*1*m* 

     





 *х*

*Х*  *i*1

 *х*

 *хij*



*im*  .





*х*

 *n*1

 

 *хnj*

  



 *хnm* 

Это матрица индивидуальных весов, где *j*-ый столбец описывает веса, приписанные *j*-м экспертом всем *n* факторам, причем

*n*

 *хij*

*i* 1

 1,

*j*  1, *m* .

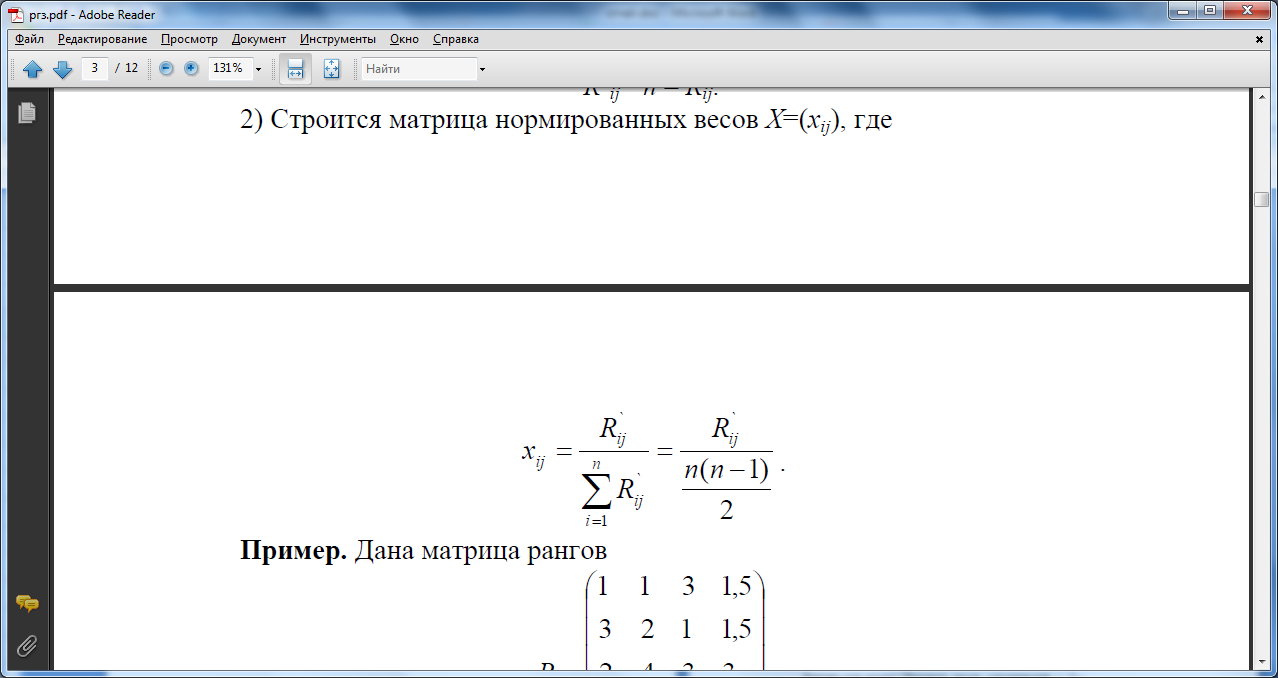
1. Построение групповых весов (при использовании метода ранжирования обычно получают также групповые ранги).
   1. Построение матрицы индивидуальных весов

Здесь в качестве исходной информации используется матрица *R*=(*Rij*).

1. Строится матрица преобразованных рангов *R`*=(*R`ij*), элементы которой вычисляются по следующему правилу:

*R`ij*= *n* – *Rij*.

1. Строится матрица нормированных весов *Х*=(*хij*)



ПРАКТИЧЕСКАЯ ЧАСТЬ

Алгоритм выполнения:

1. Ввод числа альтернатив, определяющих размерность матрицы, как показано на рис.1-4.
2. Последовательный ввод значимостей одного элемента над другим и заполнение матрицы попарных сравнений.
3. Вычисление собственного вектора матрицы попарных сравнений
4. Вычисление результирующего вектора, как показано на рис.5.

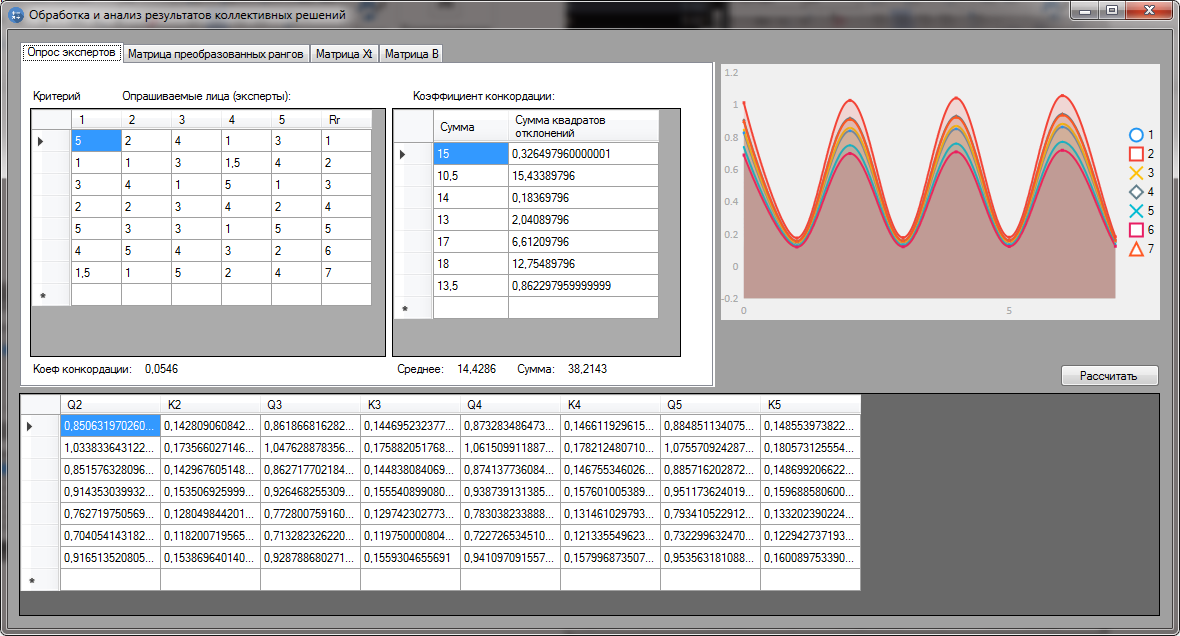


Рисунок 1 – Основное окно программы

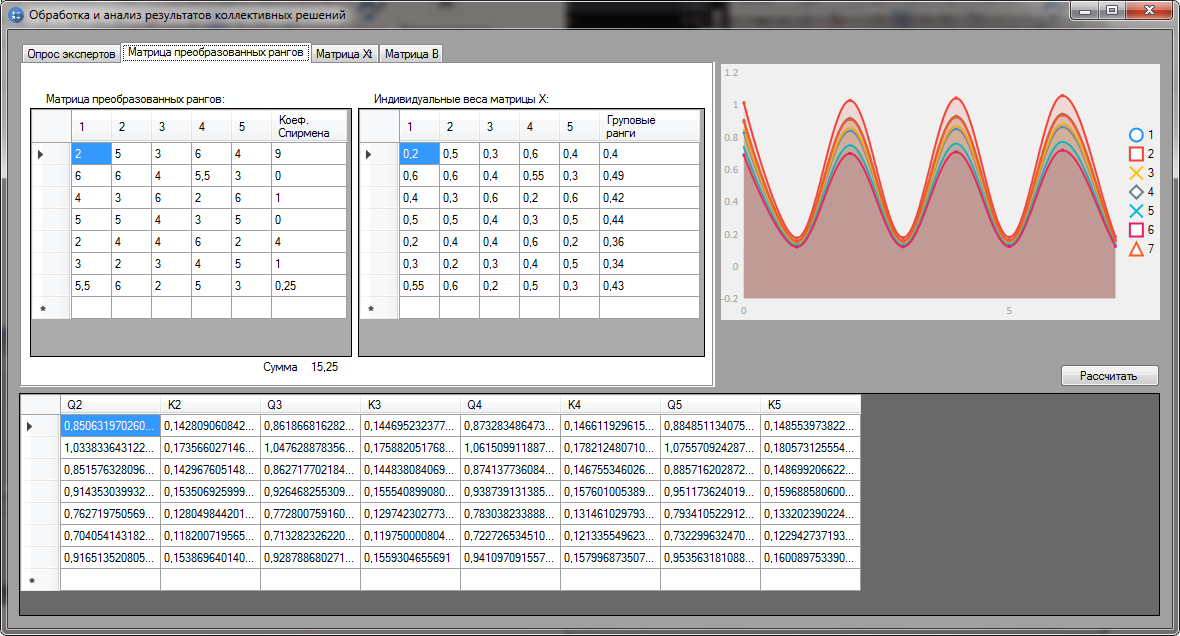


Рисунок 2 – Матрица преобразования рангов

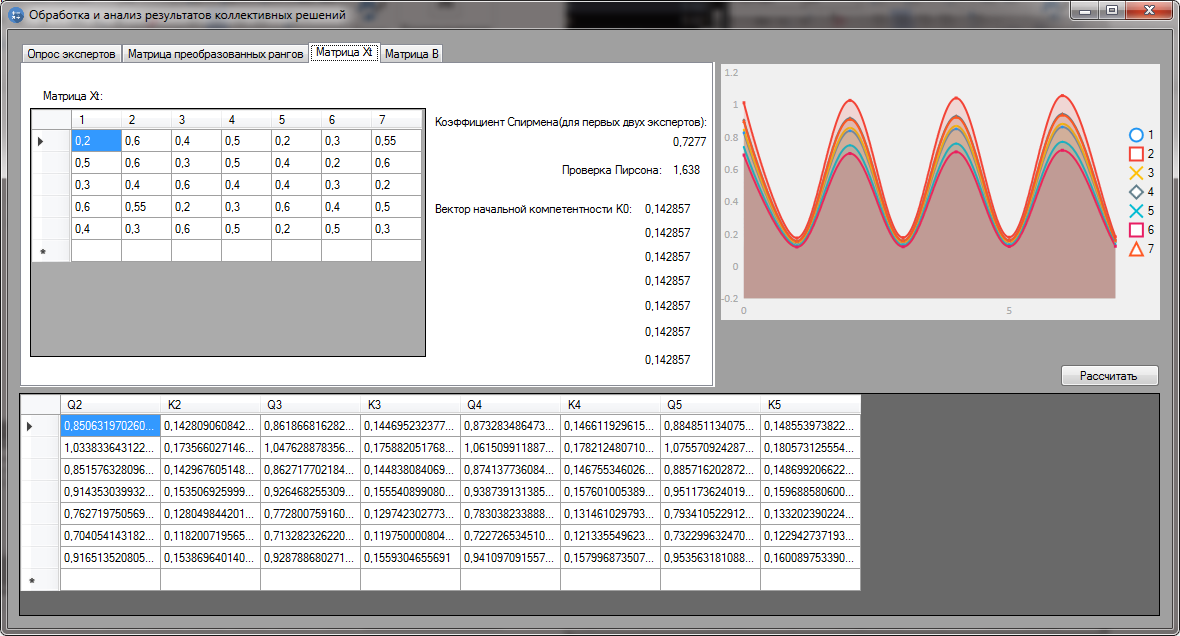


Рисунок 3 – Матрица Xt

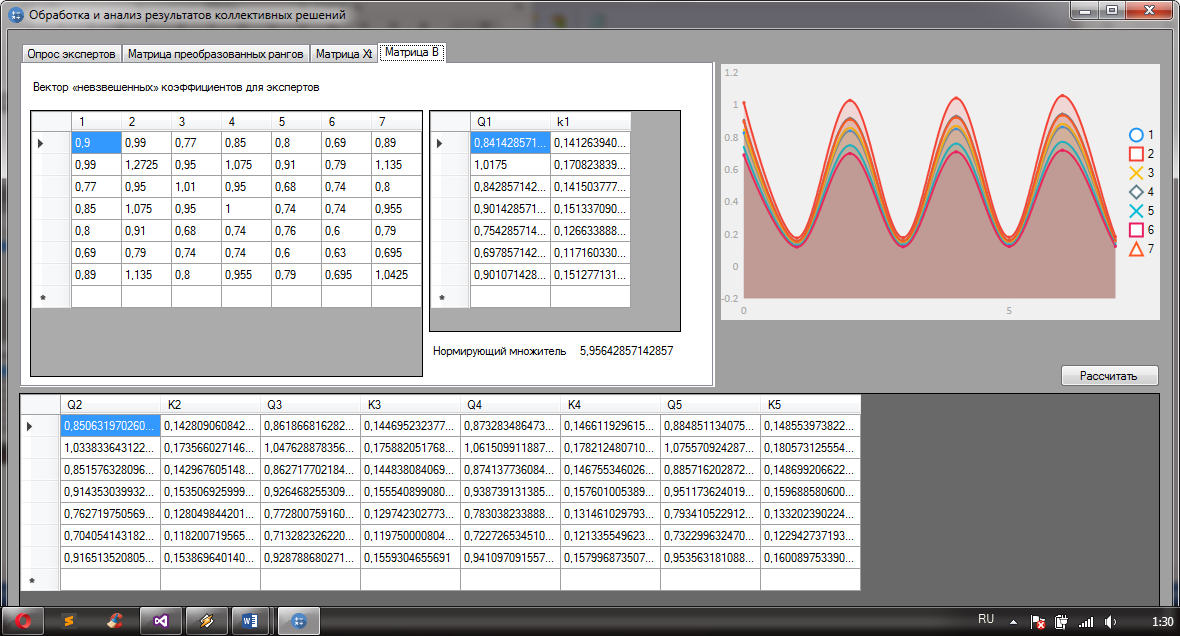


Рисунок 4 – Матрица B

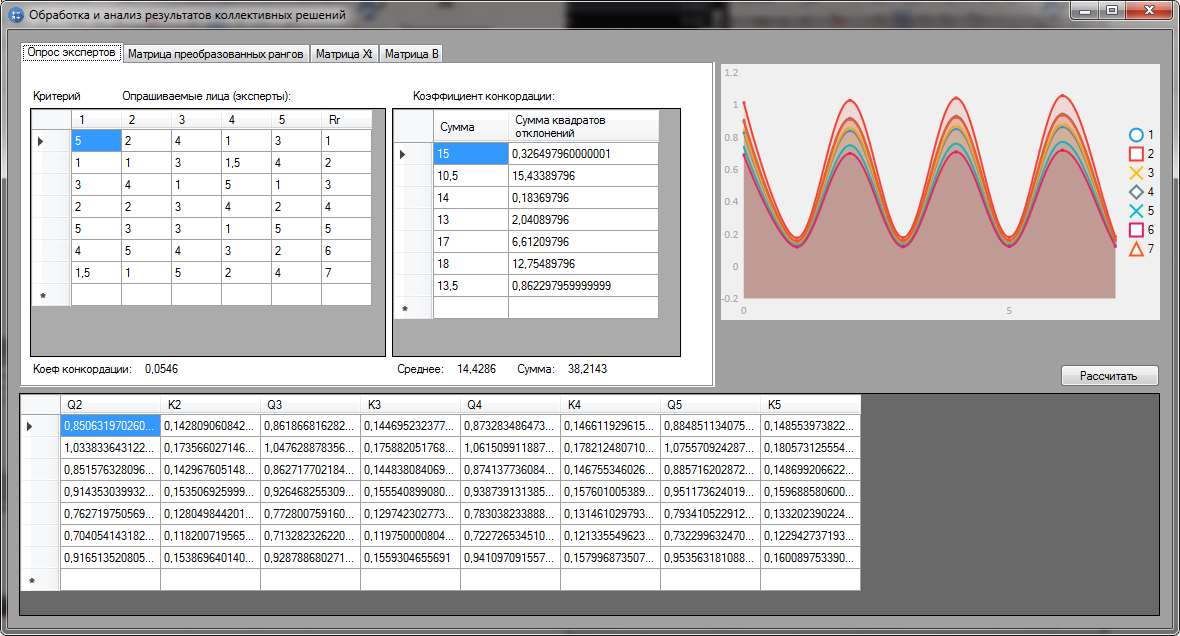


Рисунок 5 - Результат выполнения расчетов программой

ВЫВОДЫ

При выполнении лабораторной работы были применены знания определения коллективной оценки объектов (факторов и пр.) с точки зрения их воздействия на некоторую цель или показатель и применение этих знаний в разработке алгоритма и написания программного продукта на языке С#.

ЛИТЕРАТУРА

1. <http://www.ecosyn.ru/page0017.html>
2. <http://www.ecosyn.ru/page0012.html>
3. http://victor-safronov.ru/systems-analysis/lectures/zhivickaya/27.html

ЛИСТИНГ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using LiveCharts;

using LiveCharts.Wpf;

namespace Labka\_1

{

public partial class Form1 : Form

{

public object dataGridView2;

public Form1()

{

InitializeComponent();

}

//Обработчик нажатия на кнопку

public void button1\_Click(object sender, EventArgs e)

{

double perv1 = 5; double perv2 = 2; double perv3 = 4; double perv4 = 1; double perv5 = 3; double pervr = 1;

double vto1 = 1; double vto2 = 1; double vto3 = 3; double vto4 = 1.5; double vto5 = 4; double vtor = 2;

double tre1 = 3; double tre2 = 4; double tre3 = 1; double tre4 = 5; double tre5 = 1; double trer = 3;

double che1 = 2; double che2 = 2; double che3 = 3; double che4 = 4; double che5 = 2; double cher = 4;

double pia1 = 5; double pia2 = 3; double pia3 = 3; double pia4 = 1; double pia5 = 5; double piar = 5;

double she1 = 4; double she2 = 5; double she3 = 4; double she4 = 3; double she5 = 2; double sher = 6;

double sed1 = 1.5; double sed2 = 1; double sed3 = 5; double sed4 = 2; double sed5 = 4; double sedr = 7;

dataGridView3.Rows.Add(perv1, perv2, perv3, perv4, perv5, pervr);

dataGridView3.Rows.Add(vto1, vto2, vto3, vto4, vto5, vtor);

dataGridView3.Rows.Add(tre1, tre2, tre3, tre4, tre5, trer);

dataGridView3.Rows.Add(che1, che2, che3, che4, che5, cher);

dataGridView3.Rows.Add(pia1, pia2, pia3, pia4, pia5, piar);

dataGridView3.Rows.Add(she1, she2, she3, she4, she5, sher);

dataGridView3.Rows.Add(sed1, sed2, sed3, sed4, sed5, sedr);

int n = 7;

double perv12 = n - perv1; double perv22 = n - perv2; double perv32 = n - perv3; double perv42 = n - perv4; double perv52 = n - perv5; double pervr2 = Math.Pow(perv1 - perv2, 2);

double vto12 = n - vto1; double vto22 = n - vto2; double vto32 = n - vto3; double vto42 = n - vto4; double vto52 = n - vto5; double vtor2 = Math.Pow(vto1 - vto2, 2);

double tre12 = n - tre1; double tre22 = n - tre2; double tre32 = n - tre3; double tre42 = n - tre4; double tre52 = n - tre5; double trer2 = Math.Pow(tre1 - tre2, 2);

double che12 = n - che1; double che22 = n - che2; double che32 = n - che3; double che42 = n - che4; double che52 = n - che5; double cher2 = Math.Pow(che1 - che2, 2);

double pia12 = n - pia1; double pia22 = n - pia2; double pia32 = n - pia3; double pia42 = n - pia4; double pia52 = n - pia5; double piar2 = Math.Pow(pia1 - pia2, 2);

double she12 = n - she1; double she22 = n - she2; double she32 = n - she3; double she42 = n - she4; double she52 = n - she5; double sher2 = Math.Pow(she1 - she2, 2);

double sed12 = n - sed1; double sed22 = n - sed2; double sed32 = n - sed3; double sed42 = n - sed4; double sed52 = n - sed5; double sedr2 = Math.Pow(sed1 - sed2, 2);

dataGridView5.Rows.Add(perv12, perv22, perv32, perv42, perv52, pervr2);

dataGridView5.Rows.Add(vto12, vto22, vto32, vto42, vto52, vtor2);

dataGridView5.Rows.Add(tre12, tre22, tre32, tre42, tre52, trer2);

dataGridView5.Rows.Add(che12, che22, che32, che42, che52, cher2);

dataGridView5.Rows.Add(pia12, pia22, pia32, pia42, pia52, piar2);

dataGridView5.Rows.Add(she12, she22, she32, she42, she52, sher2);

dataGridView5.Rows.Add(sed12, sed22, sed32, sed42, sed52, sedr2);

double obsum = pervr2 + vtor2 + trer2 + cher2 + piar2 + sher2 + sedr2;

label37.Text = obsum.ToString();

int nd = 10;

double perv13 = perv12 / nd; double perv23 = perv22 / nd; double perv33 = perv32 / nd; double perv43 = perv42 / nd; double perv53 = perv52 / nd; double pervr3 = (perv13 + perv23 + perv33 + perv43 + perv53) / 5;

double vto13 = vto12 / nd; double vto23 = vto22 / nd; double vto33 = vto32 / nd; double vto43 = vto42 / nd; double vto53 = vto52 / nd; double vtor3 = (vto13 + vto23 + vto33 + vto43 + vto53) / 5;

double tre13 = tre12 / nd; double tre23 = tre22 / nd; double tre33 = tre32 / nd; double tre43 = tre42 / nd; double tre53 = tre52 / nd; double trer3 = (tre13 + tre23 + tre33 + tre43 + tre53) / 5;

double che13 = che12 / nd; double che23 = che22 / nd; double che33 = che32 / nd; double che43 = che42 / nd; double che53 = che52 / nd; double cher3 = (che13 + che23 + che33 + che43 + che53) / 5;

double pia13 = pia12 / nd; double pia23 = pia22 / nd; double pia33 = pia32 / nd; double pia43 = pia42 / nd; double pia53 = pia52 / nd; double piar3 = (pia13 + pia23 + pia33 + pia43 + pia53) / 5;

double she13 = she12 / nd; double she23 = she22 / nd; double she33 = she32 / nd; double she43 = she42 / nd; double she53 = she52 / nd; double sher3 = (she13 + she23 + she33 + she43 + she53) / 5;

double sed13 = sed12 / nd; double sed23 = sed22 / nd; double sed33 = sed32 / nd; double sed43 = sed42 / nd; double sed53 = sed52 / nd; double sedr3 = (sed13 + sed23 + sed33 + sed43 + sed53) / 5;

dataGridView6.Rows.Add(perv13, perv23, perv33, perv43, perv53, pervr3);

dataGridView6.Rows.Add(vto13, vto23, vto33, vto43, vto53, vtor3);

dataGridView6.Rows.Add(tre13, tre23, tre33, tre43, tre53, trer3);

dataGridView6.Rows.Add(che13, che23, che33, che43, che53, cher3);

dataGridView6.Rows.Add(pia13, pia23, pia33, pia43, pia53, piar3);

dataGridView6.Rows.Add(she13, she23, she33, she43, she53, sher3);

dataGridView6.Rows.Add(sed13, sed23, sed33, sed43, sed53, sedr3);

dataGridView7.Rows.Add(perv13, vto13, tre13, che13, pia13, she13, sed13);

dataGridView7.Rows.Add(perv23, vto23, tre23, che23, pia23, she23, sed23);

dataGridView7.Rows.Add(perv33, vto33, tre33, che33, pia33, she33, sed33);

dataGridView7.Rows.Add(perv43, vto43, tre43, che43, pia43, she43, sed43);

dataGridView7.Rows.Add(perv53, vto53, tre53, che53, pia53, she53, sed53);

double b1 = perv13 \* perv13 + perv23 \* perv23 + perv33 \* perv33 + perv43 \* perv43 + perv53 \* perv53;

double b2 = perv13 \* vto13 + perv23 \* vto23 + perv33 \* vto33 + perv43 \* vto43 + perv53 \* vto53;

double b3 = perv13 \* tre13 + perv23 \* tre23 + perv33 \* tre33 + perv43 \* tre43 + perv53 \* tre53;

double b4 = perv13 \* che13 + perv23 \* che23 + perv33 \* che33 + perv43 \* che43 + perv53 \* che53;

double b5 = perv13 \* pia13 + perv23 \* pia23 + perv33 \* pia33 + perv43 \* pia43 + perv53 \* pia53;

double b6 = perv13 \* she13 + perv23 \* she23 + perv33 \* she33 + perv43 \* she43 + perv53 \* she53;

double b7 = perv13 \* sed13 + perv23 \* sed23 + perv33 \* sed33 + perv43 \* sed43 + perv53 \* sed53;

////////////////////////////////////////

double b11 = vto13 \* perv13 + vto23 \* perv23 + vto33 \* perv33 + vto43 \* perv43 + vto53 \* perv53;

double b21 = vto13 \* vto13 + vto23 \* vto23 + vto33 \* vto33 + vto43 \* vto43 + vto53 \* vto53;

double b31 = vto13 \* tre13 + vto23 \* tre23 + vto33 \* tre33 + vto43 \* tre43 + vto53 \* tre53;

double b41 = vto13 \* che13 + vto23 \* che23 + vto33 \* che33 + vto43 \* che43 + vto53 \* che53;

double b51 = vto13 \* pia13 + vto23 \* pia23 + vto33 \* pia33 + vto43 \* pia43 + vto53 \* pia53;

double b61 = vto13 \* she13 + vto23 \* she23 + vto33 \* she33 + vto43 \* she43 + vto53 \* she53;

double b71 = vto13 \* sed13 + vto23 \* sed23 + vto33 \* sed33 + vto43 \* sed43 + vto53 \* sed53;

//////////////////////////////////////////

double b12 = tre13 \* perv13 + tre23 \* perv23 + tre33 \* perv33 + tre43 \* perv43 + tre53 \* perv53;

double b22 = tre13 \* vto13 + tre23 \* vto23 + tre33 \* vto33 + tre43 \* vto43 + tre53 \* vto53;

double b32 = tre13 \* tre13 + tre23 \* tre23 + tre33 \* tre33 + tre43 \* tre43 + tre53 \* tre53;

double b42 = tre13 \* che13 + tre23 \* che23 + tre33 \* che33 + tre43 \* che43 + tre53 \* che53;

double b52 = tre13 \* pia13 + tre23 \* pia23 + tre33 \* pia33 + tre43 \* pia43 + tre53 \* pia53;

double b62 = tre13 \* she13 + tre23 \* she23 + tre33 \* she33 + tre43 \* she43 + tre53 \* she53;

double b72 = tre13 \* sed13 + tre23 \* sed23 + tre33 \* sed33 + tre43 \* sed43 + tre53 \* sed53;

///////////////////////////////////////////

double b13 = che13 \* perv13 + che23 \* perv23 + che33 \* perv33 + che43 \* perv43 + che53 \* perv53;

double b23 = che13 \* vto13 + che23 \* vto23 + che33 \* vto33 + che43 \* vto43 + che53 \* vto53;

double b33 = che13 \* tre13 + che23 \* tre23 + che33 \* tre33 + che43 \* tre43 + che53 \* tre53;

double b43 = che13 \* che13 + che23 \* che23 + che33 \* che33 + che43 \* che43 + che53 \* che53;

double b53 = che13 \* pia13 + che23 \* pia23 + che33 \* pia33 + che43 \* pia43 + che53 \* pia53;

double b63 = che13 \* she13 + che23 \* she23 + che33 \* she33 + che43 \* she43 + che53 \* she53;

double b73 = che13 \* sed13 + che23 \* sed23 + che33 \* sed33 + che43 \* sed43 + che53 \* sed53;

/////////////////////////////

double b14 = pia13 \* perv13 + pia23 \* perv23 + pia33 \* perv33 + pia43 \* perv43 + pia53 \* perv53;

double b24 = pia13 \* vto13 + pia23 \* vto23 + pia33 \* vto33 + pia43 \* vto43 + pia53 \* vto53;

double b34 = pia13 \* tre13 + pia23 \* tre23 + pia33 \* tre33 + pia43 \* tre43 + pia53 \* tre53;

double b44 = pia13 \* che13 + pia23 \* che23 + pia33 \* che33 + pia43 \* che43 + pia53 \* che53;

double b54 = pia13 \* pia13 + pia23 \* pia23 + pia33 \* pia33 + pia43 \* pia43 + pia53 \* pia53;

double b64 = pia13 \* she13 + pia23 \* she23 + pia33 \* she33 + pia43 \* she43 + pia53 \* she53;

double b74 = pia13 \* sed13 + pia23 \* sed23 + pia33 \* sed33 + pia43 \* sed43 + pia53 \* sed53;

////////////////////////////////

double b15 = she13 \* perv13 + she23 \* perv23 + she33 \* perv33 + she43 \* perv43 + she53 \* perv53;

double b25 = she13 \* vto13 + she23 \* vto23 + she33 \* vto33 + she43 \* vto43 + she53 \* vto53;

double b35 = she13 \* tre13 + she23 \* tre23 + she33 \* tre33 + she43 \* tre43 + she53 \* tre53;

double b45 = she13 \* che13 + she23 \* che23 + she33 \* che33 + she43 \* che43 + she53 \* che53;

double b55 = she13 \* pia13 + she23 \* pia23 + she33 \* pia33 + she43 \* pia43 + she53 \* pia53;

double b65 = she13 \* she13 + she23 \* she23 + she33 \* she33 + she43 \* she43 + she53 \* she53;

double b75 = she13 \* sed13 + she23 \* sed23 + she33 \* sed33 + she43 \* sed43 + she53 \* sed53;

///////////////////////////////////////

double b16 = sed13 \* perv13 + sed23 \* perv23 + sed33 \* perv33 + sed43 \* perv43 + sed53 \* perv53;

double b26 = sed13 \* vto13 + sed23 \* vto23 + sed33 \* vto33 + sed43 \* vto43 + sed53 \* vto53;

double b36 = sed13 \* tre13 + sed23 \* tre23 + sed33 \* tre33 + sed43 \* tre43 + sed53 \* tre53;

double b46 = sed13 \* che13 + sed23 \* che23 + sed33 \* che33 + sed43 \* che43 + sed53 \* che53;

double b56 = sed13 \* pia13 + sed23 \* pia23 + sed33 \* pia33 + sed43 \* pia43 + sed53 \* pia53;

double b66 = sed13 \* she13 + sed23 \* she23 + sed33 \* she33 + sed43 \* she43 + sed53 \* she53;

double b76 = sed13 \* sed13 + sed23 \* sed23 + sed33 \* sed33 + sed43 \* sed43 + sed53 \* sed53;

dataGridView8.Rows.Add(b1, b2, b3, b4, b5, b6, b7);

dataGridView8.Rows.Add(b11, b21, b31, b41, b51, b61, b71);

dataGridView8.Rows.Add(b12, b22, b32, b42, b52, b62, b72);

dataGridView8.Rows.Add(b13, b23, b33, b43, b53, b63, b73);

dataGridView8.Rows.Add(b14, b24, b34, b44, b54, b64, b74);

dataGridView8.Rows.Add(b15, b25, b35, b45, b55, b65, b75);

dataGridView8.Rows.Add(b16, b26, b36, b46, b56, b66, b76);

/////

double chi = 1;

double slo = 7;

double chislo1 = chi / slo;

double Q1 = (chislo1 \* b1) + (chislo1 \* b2) + (chislo1 \* b3) + (chislo1 \* b4) + (chislo1 \* b5) + (chislo1 \* b6) + (chislo1 \* b7);

double Q2 = (chislo1 \* b11) + (chislo1 \* b21) + (chislo1 \* b31) + (chislo1 \* b41) + (chislo1 \* b51) + (chislo1 \* b61) + (chislo1 \* b71);

double Q3 = (chislo1 \* b12) + (chislo1 \* b22) + (chislo1 \* b32) + (chislo1 \* b42) + (chislo1 \* b52) + (chislo1 \* b62) + (chislo1 \* b72);

double Q4 = (chislo1 \* b13) + (chislo1 \* b23) + (chislo1 \* b33) + (chislo1 \* b43) + (chislo1 \* b53) + (chislo1 \* b63) + (chislo1 \* b73);

double Q5 = (chislo1 \* b14) + (chislo1 \* b24) + (chislo1 \* b34) + (chislo1 \* b44) + (chislo1 \* b54) + (chislo1 \* b64) + (chislo1 \* b74);

double Q6 = (chislo1 \* b15) + (chislo1 \* b25) + (chislo1 \* b35) + (chislo1 \* b45) + (chislo1 \* b55) + (chislo1 \* b65) + (chislo1 \* b75);

double Q7 = (chislo1 \* b16) + (chislo1 \* b26) + (chislo1 \* b36) + (chislo1 \* b46) + (chislo1 \* b56) + (chislo1 \* b66) + (chislo1 \* b76);

double sumkk = Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7;

double kz1 = Q1 / sumkk;

double kz2 = Q2 / sumkk;

double kz3 = Q3 / sumkk;

double kz4 = Q4 / sumkk;

double kz5 = Q5 / sumkk;

double kz6 = Q6 / sumkk;

double kz7 = Q7 / sumkk;

dataGridView9.Rows.Add(Q1, kz1);

dataGridView9.Rows.Add(Q2, kz2);

dataGridView9.Rows.Add(Q3, kz3);

dataGridView9.Rows.Add(Q4, kz4);

dataGridView9.Rows.Add(Q5, kz5);

dataGridView9.Rows.Add(Q6, kz6);

dataGridView9.Rows.Add(Q7, kz7);

label53.Text = sumkk.ToString();

double chislo = 0.142857;

label44.Text = chislo.ToString();

label45.Text = chislo.ToString();

label46.Text = chislo.ToString();

label47.Text = chislo.ToString();

label48.Text = chislo.ToString();

label49.Text = chislo.ToString();

label50.Text = chislo.ToString();

try

{

String s1 = dataGridView3[0, 0].Value.ToString();

double as1 = Convert.ToDouble(s1);

// label8.Text = AS.ToString();

//MessageBox.Show(s);

String s2 = dataGridView3[1, 0].Value.ToString();

double as2 = Convert.ToDouble(s2);

String s3 = dataGridView3[2, 0].Value.ToString();

double as3 = Convert.ToDouble(s3);

String s4 = dataGridView3[3, 0].Value.ToString();

double as4 = Convert.ToDouble(s4);

String s5 = dataGridView3[4, 0].Value.ToString();

double as5 = Convert.ToDouble(s5);

/////////////////////////////////////////////////

String d1 = dataGridView3[0, 1].Value.ToString();

double ad1 = Convert.ToDouble(d1);

String d2 = dataGridView3[1, 1].Value.ToString();

double ad2 = Convert.ToDouble(d2);

String d3 = dataGridView3[2, 1].Value.ToString();

double ad3 = Convert.ToDouble(d3);

String d4 = dataGridView3[3, 1].Value.ToString();

double ad4 = Convert.ToDouble(d4);

String d5 = dataGridView3[4, 1].Value.ToString();

double ad5 = Convert.ToDouble(d5);

/////////////////////////////////////////////////

String f1 = dataGridView3[0, 2].Value.ToString();

double af1 = Convert.ToDouble(f1);

String f2 = dataGridView3[1, 2].Value.ToString();

double af2 = Convert.ToDouble(f2);

String f3 = dataGridView3[2, 2].Value.ToString();

double af3 = Convert.ToDouble(f3);

String f4 = dataGridView3[3, 2].Value.ToString();

double af4 = Convert.ToDouble(f4);

String f5 = dataGridView3[4, 2].Value.ToString();

double af5 = Convert.ToDouble(f5);

/////////////////////////////////////////////////

String g1 = dataGridView3[0, 3].Value.ToString();

double ag1 = Convert.ToDouble(g1);

String g2 = dataGridView3[1, 3].Value.ToString();

double ag2 = Convert.ToDouble(g2);

String g3 = dataGridView3[2, 3].Value.ToString();

double ag3 = Convert.ToDouble(g3);

String g4 = dataGridView3[3, 3].Value.ToString();

double ag4 = Convert.ToDouble(g4);

String g5 = dataGridView3[4, 3].Value.ToString();

double ag5 = Convert.ToDouble(g5);

/////////////////////////////////////////////////

String h1 = dataGridView3[0, 4].Value.ToString();

double ah1 = Convert.ToDouble(h1);

String h2 = dataGridView3[1, 4].Value.ToString();

double ah2 = Convert.ToDouble(h2);

String h3 = dataGridView3[2, 4].Value.ToString();

double ah3 = Convert.ToDouble(h3);

String h4 = dataGridView3[3, 4].Value.ToString();

double ah4 = Convert.ToDouble(h4);

String h5 = dataGridView3[4, 4].Value.ToString();

double ah5 = Convert.ToDouble(h5);

/////////////////////////////////////////////////

String j1 = dataGridView3[0, 5].Value.ToString();

double aj1 = Convert.ToDouble(j1);

String j2 = dataGridView3[1, 5].Value.ToString();

double aj2 = Convert.ToDouble(j2);

String j3 = dataGridView3[2, 5].Value.ToString();

double aj3 = Convert.ToDouble(j3);

String j4 = dataGridView3[3, 5].Value.ToString();

double aj4 = Convert.ToDouble(j4);

String j5 = dataGridView3[4, 5].Value.ToString();

double aj5 = Convert.ToDouble(j5);

/////////////////////////////////////////////////

String k1 = dataGridView3[0, 6].Value.ToString();

double ak1 = Convert.ToDouble(k1);

String k2 = dataGridView3[1, 6].Value.ToString();

double ak2 = Convert.ToDouble(k2);

String k3 = dataGridView3[2, 6].Value.ToString();

double ak3 = Convert.ToDouble(k3);

String k4 = dataGridView3[3, 6].Value.ToString();

double ak4 = Convert.ToDouble(k4);

String k5 = dataGridView3[4, 6].Value.ToString();

double ak5 = Convert.ToDouble(k5);

/////////////////////////////////////////////////

double sum1 = as1 + as2 + as3 + as4 + as5;

double sum2 = ad1 + ad2 + ad3 + ad4 + ad5;

double sum3 = af1 + af2 + af3 + af4 + af5;

double sum4 = ag1 + ag2 + ag3 + ag4 + ag5;

double sum5 = ah1 + ah2 + ah3 + ah4 + ah5;

double sum6 = aj1 + aj2 + aj3 + aj4 + aj5;

double sum7 = ak1 + ak2 + ak3 + ak4 + ak5;

double sred\_sum = (sum1 + sum2 + sum3 + sum4 + sum5 + sum6 + sum7) / 7;

sred\_sum = Math.Round(sred\_sum, 4);

label14.Text = sred\_sum.ToString();

double sumkva1 = Math.Pow(sum1 - sred\_sum, 2);

double sumkva2 = Math.Pow(sum2 - sred\_sum, 2);

double sumkva3 = Math.Pow(sum3 - sred\_sum, 2);

double sumkva4 = Math.Pow(sum4 - sred\_sum, 2);

double sumkva5 = Math.Pow(sum5 - sred\_sum, 2);

double sumkva6 = Math.Pow(sum6 - sred\_sum, 2);

double sumkva7 = Math.Pow(sum7 - sred\_sum, 2);

double suma = (sumkva1 + sumkva2 + sumkva3 + sumkva4 + sumkva5 + sumkva6 + sumkva7);

suma = Math.Round(suma, 4);

label31.Text = suma.ToString();

double konk = (12 \* suma) / (5 \* 5 \* (Math.Pow(7, 3) - 7));

konk = Math.Round(konk, 4);

label33.Text = konk.ToString();

dataGridView4.Rows.Add(sum1, sumkva1);

dataGridView4.Rows.Add(sum2, sumkva2);

dataGridView4.Rows.Add(sum3, sumkva3);

dataGridView4.Rows.Add(sum4, sumkva4);

dataGridView4.Rows.Add(sum5, sumkva5);

dataGridView4.Rows.Add(sum6, sumkva6);

dataGridView4.Rows.Add(sum7, sumkva7);

String obsuma0 = dataGridView5[5, 0].Value.ToString();

String obsuma1 = dataGridView5[5, 1].Value.ToString();

String obsuma2 = dataGridView5[5, 2].Value.ToString();

String obsuma3 = dataGridView5[5, 3].Value.ToString();

String obsuma4 = dataGridView5[5, 4].Value.ToString();

String obsuma5 = dataGridView5[5, 5].Value.ToString();

String obsuma6 = dataGridView5[5, 6].Value.ToString();

double obsum0 = Convert.ToDouble(obsuma0);

double obsum1 = Convert.ToDouble(obsuma1);

double obsum2 = Convert.ToDouble(obsuma2);

double obsum3 = Convert.ToDouble(obsuma3);

double obsum4 = Convert.ToDouble(obsuma4);

double obsum5 = Convert.ToDouble(obsuma5);

double obsum6 = Convert.ToDouble(obsuma6);

double obsum10 = obsum0 + obsum1 + obsum2 + obsum3 + obsum4 + obsum5 + obsum6;

double kospi = 1 - ((6 / (Math.Pow(7, 3) - 7)) \* obsum10);

kospi = Math.Round(kospi, 4);

label40.Text = kospi.ToString();

double a = 5 \* (7 - 1) \* konk;

label42.Text = a.ToString();

double q21 = (kz1 \* b1) + (kz2 \* b2) + (kz3 \* b3) + (kz4 \* b4) + (kz5 \* b5) + (kz6 \* b6) + (kz7 \* b7);

double q22 = (kz1 \* b11) + (kz2 \* b21) + (kz3 \* b31) + (kz4 \* b41) + (kz5 \* b51) + (kz6 \* b61) + (kz7 \* b71);

double q23 = (kz1 \* b12) + (kz2 \* b22) + (kz3 \* b32) + (kz4 \* b42) + (kz5 \* b52) + (kz6 \* b62) + (kz7 \* b72);

double q24 = (kz1 \* b13) + (kz2 \* b23) + (kz3 \* b33) + (kz4 \* b43) + (kz5 \* b53) + (kz6 \* b63) + (kz7 \* b73);

double q25 = (kz1 \* b14) + (kz2 \* b24) + (kz3 \* b34) + (kz4 \* b44) + (kz5 \* b54) + (kz6 \* b64) + (kz7 \* b74);

double q26 = (kz1 \* b15) + (kz2 \* b25) + (kz3 \* b35) + (kz4 \* b45) + (kz5 \* b55) + (kz6 \* b65) + (kz7 \* b75);

double q27 = (kz1 \* b16) + (kz2 \* b26) + (kz3 \* b36) + (kz4 \* b46) + (kz5 \* b56) + (kz6 \* b66) + (kz7 \* b76);

double k21 = q21 / sumkk;

double k22 = q22 / sumkk;

double k23 = q23 / sumkk;

double k24 = q24 / sumkk;

double k25 = q25 / sumkk;

double k26 = q26 / sumkk;

double k27 = q27 / sumkk;

double q31 = (k21 \* b1) + (k22 \* b2) + (k23 \* b3) + (k24 \* b4) + (k25 \* b5) + (k26 \* b6) + (k27 \* b7);

double q32 = (k21 \* b11) + (k22 \* b21) + (k23 \* b31) + (k24 \* b41) + (k25 \* b51) + (k26 \* b61) + (k27 \* b71);

double q33 = (k21 \* b12) + (k22 \* b22) + (k23 \* b32) + (k24 \* b42) + (k25 \* b52) + (k26 \* b62) + (k27 \* b72);

double q34 = (k21 \* b13) + (k22 \* b23) + (k23 \* b33) + (k24 \* b43) + (k25 \* b53) + (k26 \* b63) + (k27 \* b73);

double q35 = (k21 \* b14) + (k22 \* b24) + (k23 \* b34) + (k24 \* b44) + (k25 \* b54) + (k26 \* b64) + (k27 \* b74);

double q36 = (k21 \* b15) + (k22 \* b25) + (k23 \* b35) + (k24 \* b45) + (k25 \* b55) + (k26 \* b65) + (k27 \* b75);

double q37 = (k21 \* b16) + (k22 \* b26) + (k23 \* b36) + (k24 \* b46) + (k25 \* b56) + (k26 \* b66) + (k27 \* b76);

double k31 = q31 / sumkk;

double k32 = q32 / sumkk;

double k33 = q33 / sumkk;

double k34 = q34 / sumkk;

double k35 = q35 / sumkk;

double k36 = q36 / sumkk;

double k37 = q37 / sumkk;

double q41 = (k31 \* b1) + (k32 \* b2) + (k33 \* b3) + (k34 \* b4) + (k35 \* b5) + (k36 \* b6) + (k37 \* b7);

double q42 = (k31 \* b11) + (k32 \* b21) + (k33 \* b31) + (k34 \* b41) + (k35 \* b51) + (k36 \* b61) + (k37 \* b71);

double q43 = (k31 \* b12) + (k32 \* b22) + (k33 \* b32) + (k34 \* b42) + (k35 \* b52) + (k36 \* b62) + (k37 \* b72);

double q44 = (k31 \* b13) + (k32 \* b23) + (k33 \* b33) + (k34 \* b43) + (k35 \* b53) + (k36 \* b63) + (k37 \* b73);

double q45 = (k31 \* b14) + (k32 \* b24) + (k33 \* b34) + (k34 \* b44) + (k35 \* b54) + (k36 \* b64) + (k37 \* b74);

double q46 = (k31 \* b15) + (k32 \* b25) + (k33 \* b35) + (k34 \* b45) + (k35 \* b55) + (k36 \* b65) + (k37 \* b75);

double q47 = (k31 \* b16) + (k32 \* b26) + (k33 \* b36) + (k34 \* b46) + (k35 \* b56) + (k36 \* b66) + (k37 \* b76);

double k41 = q41 / sumkk;

double k42 = q42 / sumkk;

double k43 = q43 / sumkk;

double k44 = q44 / sumkk;

double k45 = q45 / sumkk;

double k46 = q46 / sumkk;

double k47 = q47 / sumkk;

double q51 = (k41 \* b1) + (k42 \* b2) + (k43 \* b3) + (k44 \* b4) + (k45 \* b5) + (k46 \* b6) + (k47 \* b7);

double q52 = (k41 \* b11) + (k42 \* b21) + (k43 \* b31) + (k44 \* b41) + (k45 \* b51) + (k46 \* b61) + (k47 \* b71);

double q53 = (k41 \* b12) + (k42 \* b22) + (k43 \* b32) + (k44 \* b42) + (k45 \* b52) + (k46 \* b62) + (k47 \* b72);

double q54 = (k41 \* b13) + (k42 \* b23) + (k43 \* b33) + (k44 \* b43) + (k45 \* b53) + (k46 \* b63) + (k47 \* b73);

double q55 = (k41 \* b14) + (k42 \* b24) + (k43 \* b34) + (k44 \* b44) + (k45 \* b54) + (k46 \* b64) + (k47 \* b74);

double q56 = (k41 \* b15) + (k42 \* b25) + (k43 \* b35) + (k44 \* b45) + (k45 \* b55) + (k46 \* b65) + (k47 \* b75);

double q57 = (k41 \* b16) + (k42 \* b26) + (k43 \* b36) + (k44 \* b46) + (k45 \* b56) + (k46 \* b66) + (k47 \* b76);

double k51 = q51 / sumkk;

double k52 = q52 / sumkk;

double k53 = q53 / sumkk;

double k54 = q54 / sumkk;

double k55 = q55 / sumkk;

double k56 = q56 / sumkk;

double k57 = q57 / sumkk;

dataGridView1.Rows.Add(q21, k21, q31, k31, q41, k41, q51, k51);

dataGridView1.Rows.Add(q22, k22, q32, k32, q42, k42, q52, k52);

dataGridView1.Rows.Add(q23, k23, q33, k33, q43, k43, q53, k53);

dataGridView1.Rows.Add(q24, k24, q34, k34, q44, k44, q54, k54);

dataGridView1.Rows.Add(q25, k25, q35, k35, q45, k45, q55, k55);

dataGridView1.Rows.Add(q26, k26, q36, k36, q46, k46, q56, k56);

dataGridView1.Rows.Add(q27, k27, q37, k37, q47, k47, q57, k57);

}

catch

{

MessageBox.Show("Данной ячейки не существует!");

}

//Вывод графика

cartesianChart2.Series = new SeriesCollection

{

new LineSeries

{

Title = "1",

Values = new ChartValues<double> { 0.825299387, 0.14176922, 0.837085511, 0.14379383, 0.849062304, 0.145851193, 0.861209451, 0.14793782},

PointGeometry = DefaultGeometries.Circle,

PointGeometrySize = 3

},

new LineSeries

{

Title = "2",

Values = new ChartValues<double> { 1.010865798, 0.173645658, 1.025427121, 0.17614699, 1.040101606, 0.178667761, 1.05498194, 0.181223892},

PointGeometry = DefaultGeometries.Square,

PointGeometrySize = 3

},

new LineSeries

{

Title = "3",

Values = new ChartValues<double> { 0.843956442, 0.144974113, 0.855876239, 0.147021685, 0.868111964, 0.149123528, 0.880531053, 0.151256868},

PointGeometry = DefaultGeometries.Cross,

PointGeometrySize = 3

},

new LineSeries

{

Title = "4",

Values = new ChartValues<double> { 0.902268098, 0.154990839, 0.915165674, 0.157206373, 0.928256556, 0.159455114, 0.941536452, 0.161736323},

PointGeometry = DefaultGeometries.Diamond,

PointGeometrySize = 3

},

new LineSeries

{

Title = "5",

Values = new ChartValues<double> { 0.737299387, 0.126652655, 0.747840957, 0.128463477, 0.758541716, 0.130301645, 0.76939388, 0.13216582},

PointGeometry = DefaultGeometries.Cross,

PointGeometrySize = 3

},

new LineSeries

{

Title = "6",

Values = new ChartValues<double> { 0.687532209, 0.118103692, 0.697277465, 0.119777724, 0.707248917, 0.121490612, 0.717366907, 0.123228671},

PointGeometry = DefaultGeometries.Square,

PointGeometrySize = 3

},

new LineSeries

{

Title = "7",

Values = new ChartValues<double> { 0.895561503, 0.153838786, 0.908507184, 0.156062584, 0.921510612, 0.158296301, 0.934694416, 0.160561004},

PointGeometry = DefaultGeometries.Triangle,

PointGeometrySize = 3

},

};

cartesianChart2.LegendLocation = LegendLocation.Right;

}

}

}