namespace Riski\_ocenka

{

public partial class Form1 : Form

{

private string a;

private double are = 0; //Отн.риск для лиц подверг.возд. (Are) %

private double are1000 = 0; //Отн.риск для лиц подверг.возд. (Are) на 1000 в год

private string b;

private string c;

private string d;

private DataSet ds;

private double expng = 0; //exp ниж гран

private double expvg = 0; //exp верх гран

private Form5 f5;

// ДИ для RR

private double lnpr = 0; //логарифм RR

//ДИ для (Are) %

private double ngare = 0; //ниж граница

//ДИ для (PAR) %

private double ngpar = 0; //ниж граница

//ДИ для R

private double ngr = 0; //ниж граница

private double ngr1 = 0; //ниж граница

private double ngrr = 0; //ниж граница для RR

private double par = 0; //Отн.риск для населения в целом (PAR) %

private double par1000 = 0; //Отн.риск для населения в целом (PAR) на 1000 в год

private double pri1000 = 0; //Прирост на 1000 в год

private double r0 = 0;

private string random\_string;

private int random6 = 0;

private int random7 = 0;

private double rr = 0; // Относительный риск, (RR)

private double rt = 0;

private double sqrtr = 0; //корень

private int sum1, sum2;

//textBox6.Text = Convert.ToString(ur1000i);

//textBox7.Text = Convert.ToString(ur1000e);

private DataTable table\_for\_report;

private int ur1000e;

private int ur1000e1;

private int ur1000i;

private int ur1000i1;

private double vgare = 0; //верх граница

private double vgpar = 0; //верх граница

private double vgr = 0; //верх граница

private double vgr1 = 0; //верх граница

private double vgrr = 0; //верх граница для RR

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

var now = DateTime.Now;

// Create the list to use as the custom source.

/\* var source = new AutoCompleteStringCollection();}

source.AddRange(new string[]

{

"Агломераційний цех",

"Доменний цех",

"Мартенівський цех",

"Цех підготовки складів",

"Ливарний цех",

"Обжимний цех",

"Цех гарячої прокатки тонкого листа",

"Цех холодної прокатки № 1",

"Цех холодної прокатки № 2",

"Цех холодної прокатки № 3"

});

// Create and initialize the text box.

var textBox9 = new TextBox

{

AutoCompleteCustomSource = source,

AutoCompleteMode = AutoCompleteMode.SuggestAppend,

AutoCompleteSource = AutoCompleteSource.CustomSource,

Location = new Point(93, 55),

//Width = ClientRectangle.Width - 17,

Width = 260,

Text = "Доменний цех",

Visible = true

};

\*/

// Add the text box to the form.

Controls.Add(textBox9);

// this.reportViewer1.RefreshReport();

}

private void button2\_Click(object sender, EventArgs e)

{

//foreach (Form f1 in this.MdiChildren)

{

// f1.Close();

Close();

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)))

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)))

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

}

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)))

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

}

private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

private void textBox10\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)))

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

}

private void textBox11\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(char.IsDigit(e.KeyChar)))

{

if (!(char.IsDigit(e.KeyChar)) && (e.KeyChar != 45) && (e.KeyChar != (char) Keys.Back))

{

e.Handled = true;

}

}

}

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

{

ur1000i = 0;

ur1000e = 0;

ur1000i1 = 0;

ur1000e1 = 0;

a = "0";

b = "0";

c = "0";

d = "0";

a = textBox2.Text;

b = textBox3.Text;

c = textBox4.Text;

d = textBox5.Text;

sum1 = Convert.ToInt32(textBox2.Text) + Convert.ToInt32(textBox4.Text);

sum2 = Convert.ToInt32(textBox3.Text) + Convert.ToInt32(textBox5.Text);

ur1000i = Convert.ToInt32(a);

ur1000e = Convert.ToInt32(b);

ur1000i1 = Convert.ToInt32(c);

ur1000e1 = Convert.ToInt32(d);

if (ur1000e == 0) //This is to make sure that we don't divide by 0

{

MessageBox.Show("Неможливість ділення на нуль!");

random\_string = "Divide by zero error";

}

else

{

var random6 = (double) ur1000i/ur1000e\*1000;

random6 = Math.Round(random6, 4);

random\_string = Convert.ToString(random6);

// random\_string = Convert.ToString((double)random6);

}

//this.label3.Text = random\_string;

textBox6.Text = random\_string;

if (ur1000e1 == 0) //This is to make sure that we don't divide by 0

{

MessageBox.Show("Неможливість ділення на нуль!");

random\_string = "Divide by zero error";

}

else

{

var random7 = (double) ur1000i1/ur1000e1\*1000;

random\_string = Convert.ToString(random7);

random7 = Math.Round(random7, 4);

random\_string = Convert.ToString(random7);

}

//this.label3.Text = random\_string;

textBox7.Text = random\_string;

textBox10.Text = Convert.ToString(sum1);

textBox11.Text = Convert.ToString(sum2);

//prirost

var pri1000 = Convert.ToDouble(textBox6.Text) - Convert.ToDouble(textBox7.Text); //Прирост на 1000 в год

var rr = ((double) ur1000i/ur1000e)/((double) ur1000i1/ur1000e1); // Относительный риск, (RR)

var are1000 = (((double) ur1000i/ur1000e) - ((double) ur1000i1/ur1000e1))\*1000;

//Отн.риск для лиц подверг.возд. (Are) на 1000 в год

var are = ((rr - 1)/rr)\*100; //Отн.риск для лиц подверг.возд. (Are) %

var par1000 = ((Convert.ToDouble(textBox10.Text)/Convert.ToDouble(textBox11.Text)) -

(Convert.ToDouble(textBox4.Text)/Convert.ToDouble(textBox5.Text)))\*1000;

//Отн.риск для населения в целом (PAR) на 1000 в год

var par = ((Convert.ToDouble(textBox10.Text)/Convert.ToDouble(textBox11.Text)) -

(Convert.ToDouble(textBox4.Text)/Convert.ToDouble(textBox5.Text)))/

((Convert.ToDouble(textBox10.Text)/Convert.ToDouble(textBox11.Text)))\*100;

//Отн.риск для населения в целом (PAR) %

// ДИ для RR

var lnpr = Math.Log(rr); //логарифм RR

var sqrtr = Math.Sqrt(1/Convert.ToDouble(textBox2.Text) + 1/Convert.ToDouble(textBox4.Text)); //корень

var ngrr = lnpr - (1.96\*sqrtr); //ниж граница для RR

var vgrr = lnpr + (1.96\*sqrtr); //верх граница для RR

var expng = Math.Exp(ngrr); //exp ниж гран

var expvg = Math.Exp(vgrr); //exp верх гран

//ДИ для (Are) %

var ngare = ((expng - 1)/expng)\*100; //ниж граница

var vgare = ((expvg - 1)/expvg)\*100; //верх граница

//ДИ для (PAR) %

var r0 = (double) ur1000i1/ur1000e1;

var rt = (double) sum1/sum2;

var ngpar = par -

1.96\*

Math.Sqrt(ur1000i1\*Math.Pow(rt, 2)/Math.Pow(ur1000e1, 2) + Math.Pow(r0, 2)/Math.Pow(rt, 2) -

sum1/Math.Pow(sum2, 2)); //ниж граница

var vgpar = par +

1.96\*

Math.Sqrt(ur1000i1\*Math.Pow(rt, 2)/Math.Pow(ur1000e1, 2) + Math.Pow(r0, 2)/Math.Pow(rt, 2) -

sum1/Math.Pow(sum2, 2)); //верх граница

//ДИ для R исследуемая группа

var ngr = (double) ur1000i/ur1000e\*1000 - 1.96\*(Math.Sqrt((double) ur1000i/Math.Pow(ur1000e,2))); //ниж граница

var vgr = (double) ur1000i/ur1000e\*1000 + 1.96\*(Math.Sqrt((double) ur1000i/Math.Pow(ur1000e,2))); //верх граница

//ДИ для R контрольная группа

var ngr1 = (double)ur1000i / ur1000e \* 1000 - 1.96 \* (Math.Sqrt((double)ur1000i1 / Math.Pow(ur1000e1, 2))); //ниж граница

var vgr1 = (double)ur1000i / ur1000e \* 1000 + 1.96 \* (Math.Sqrt((double)ur1000i1 / Math.Pow(ur1000e1, 2))); //верх граница

if (ds == null)

{

ds = new DataSet("DataSet1");

// get from disigner

//table\_for\_report = ds.Tables["dt1"];

// or create in code

table\_for\_report = ds.Tables.Add("DataTable1");

table\_for\_report.Columns.AddRange(new[]

{

new DataColumn("ID", typeof (long)),

new DataColumn("name", typeof (string)),

new DataColumn("znach\_text", typeof (string)),

new DataColumn("znach", typeof (double)),

new DataColumn("descr", typeof (string)),

new DataColumn("z1", typeof (double)),

new DataColumn("z2", typeof (double))

});

}

table\_for\_report.Rows.Add(1, "Назва дослідження", textBox1.Text, 0.0, " ",null,null);

table\_for\_report.Rows.Add(2, "Досліджувана група", textBox9.Text, 0.0, " ", null, null);

table\_for\_report.Rows.Add(3, "Контрольна група", textBox8.Text, 0.0, " ", null, null);

table\_for\_report.Rows.Add(4, "Кількість захворювань у досліджуваній групі", "",

Convert.ToDouble(textBox2.Text), " ", null, null);

table\_for\_report.Rows.Add(5, "Людино-роки у досліджуваній групі", "", Convert.ToDouble(textBox3.Text), " ", null, null);

table\_for\_report.Rows.Add(6, "Кількість захворювань у контрольній групі", "",

Convert.ToDouble(textBox4.Text), " ", null, null);

table\_for\_report.Rows.Add(7, "Людино-роки у контрольній групі", "", Convert.ToDouble(textBox5.Text), " ", null, null);

table\_for\_report.Rows.Add(8, "Всього захворювань у 2-х групах", "", Convert.ToDouble(textBox10.Text), " ", null, null);

table\_for\_report.Rows.Add(9, "Всього людино-років у 2-х групах", "", Convert.ToDouble(textBox11.Text), " ", null, null);

table\_for\_report.Rows.Add(10, "Рівень захворюваності на 1000 працюючих у досліджуваній групі", "",

Convert.ToDouble(textBox6.Text), " ", ngr, vgr);

table\_for\_report.Rows.Add(11, "Рівень захворюваності на 1000 працюючих у контрольній групі", "",

Convert.ToDouble(textBox7.Text), " ", ngr1, vgr1);

//prirost

table\_for\_report.Rows.Add(12, "Приріст на 1000 працюючих у рік", "", pri1000, " ", null, null);

table\_for\_report.Rows.Add(13, "Відносний ризик, (RR)", "", Convert.ToDecimal(rr), "RR", expng, expvg);

table\_for\_report.Rows.Add(14,

"Відносний ризик для осіб, які підлягали шкідливому впливу (ARe) на 1000 працюючих в рік", "", are1000,

" ", null, null);

table\_for\_report.Rows.Add(15, "Відносний ризик для осіб, які підлягали шкідливому впливу (ARe, %) ", "", are,

" ", ngare, vgare);

table\_for\_report.Rows.Add(16, "Відносний ризик для населення в цілому (PAR) на 1000 працюючих в рік", "",

par1000, " ", null, null);

table\_for\_report.Rows.Add(17, "Відносний ризик для населення в цілому (PAR, %)", "", par, " ", ngpar, vgpar);

// ДИ для RR

// table\_for\_report.Rows.Add(18, "логарифм RR", "", lnpr, " ",null,null);

// table\_for\_report.Rows.Add(19, "корень", "", sqrtr, " ",null,null);

// table\_for\_report.Rows.Add(20, "ниж граница для RR", "", ngrr, " ",null,null);

// table\_for\_report.Rows.Add(21, "верх граница для RR", "", vgrr, " ",null,null);

//table\_for\_report.Rows.Add(22, "ДИ для RR, нижня межа", "", expng, " ", null, null);

// table\_for\_report.Rows.Add(23, "ДИ для RR, верхня межа", "", expvg, " ", null, null);

//ДИ для (Are) %

//table\_for\_report.Rows.Add(24, "ДИ для ARe %, нижня межа", "", ngare, " ", null, null);

// table\_for\_report.Rows.Add(25, "ДИ для ARe %, верхня межа", "", vgare, " ", null, null);

//ДИ для (PAR) %

// table\_for\_report.Rows.Add (26, "ДИ для (PAR) % R0", "", r0, " ",null,null);

// table\_for\_report.Rows.Add (27, "ДИ для (PAR) % Rt", "", rt, " ",null,null);

// table\_for\_report.Rows.Add(28, "ДИ для PAR %, нижня межа", "", ngpar, " ", null, null); //ниж граница

// table\_for\_report.Rows.Add(29, "ДИ для PAR %, верхня межа", "", vgpar, " ", null, null); //верх граница

//ДИ для R

// table\_for\_report.Rows.Add(30, "ДИ для R, нижня межа", "", ngr, " ",null,null); //ниж граница

// table\_for\_report.Rows.Add(31, "ДИ для R, верхня межа", "", vgr, " ",null,null); //верх граница

}

private void button3\_Click(object sender, EventArgs e)

{

if (f5 == null)

{

f5 = new Form5();

f5.MdiParent = MdiParent;

f5.FormClosed += f5\_FormClosed;

var cryRpt = new ReportDocument();

cryRpt.Load(@"CrystalReport1.rpt");

cryRpt.SetDataSource(table\_for\_report);

f5.crystalReportViewer1.ReportSource = cryRpt;

f5.crystalReportViewer1.Refresh();

f5.Show();

}

else

f5.Activate();

}

private void f5\_FormClosed(object sender, FormClosedEventArgs e)

{

f5 = null;

// throw new NotImplementedException();

}

}