**Содержание – выполнил Фёдоров Илья**

**\*7 заданий с сайта**

**\*Абрамян**

**\*Абрамян 2**

**\*Тест**

**\*Игра виселица**

**\*Шифрование**

**Атбаш**

**Цезарь**

**Табличный**

**Вижинера**

**\*Текстовый редактор**

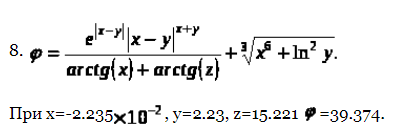
**\*Игра лабиринт поиск побег**

**\*База Данных**

**\*Сложное задание**

С сайта

1.



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

{

double x, y, z;

double n, n1, n2, n3;

x = -2.235 \* Math.Pow(10, 2);

y = 2.23;

z = 15.221;

textBox1.Text = Convert.ToString(x);

textBox2.Text = Convert.ToString(y);

textBox3.Text = Convert.ToString(z);

n1 = Math.Pow(Math.E, Math.Abs(x - y)) \* Math.Pow(Math.Abs(x - y), (x + y));

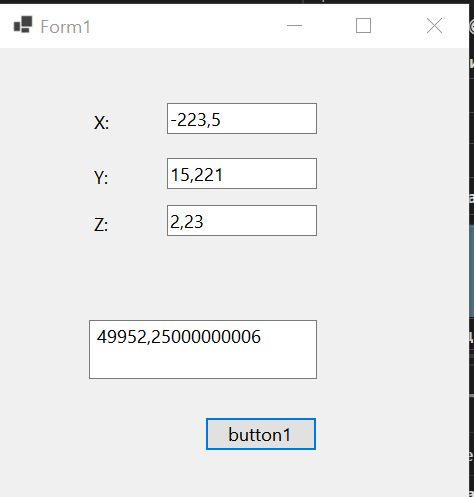
n2 = Math.Atan(x) + Math.Atan(z);

n3 = Math.Pow(Math.Pow(x, 6) + Math.Pow(Math.Log(y), 2), (1.0 / 3));

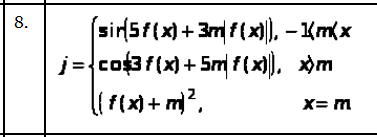
n = n1 / n2 + n3;

textBox4.Text = Convert.ToString(n);

}



1



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

{

double l = Convert.ToDouble(textBox1.Text);

double m = Convert.ToDouble(textBox2.Text);

double x = Convert.ToDouble(textBox3.Text);

int n = 0;

if (radioButton2.Checked) n = 1;

else if (radioButton3.Checked) n = 2;

double j = 0;

switch (n)

{

case 0:

if (-l < m && -l < x)

{

j = Math.Sin(5 \* Math.Sinh(x) + 3 \* m \* Math.Abs(Math.Sinh(x)));

}

else if (x > m)

{

j = Math.Cos(3 \* Math.Sinh(x) + 5 \* m \* Math.Abs(Math.Sinh(x)));

}

else

{

Math.Pow((Math.Sinh(x) + m), 2);

}

textBox4.Text = Convert.ToString(j);

break;

case 1:

if (-l < m && -l < x)

{

j = Math.Sin(5 \* x \* x + 3 \* m \* Math.Abs(x \* x));

}

else if (x > m)

{

j = Math.Cos(3 \* x \* x + 5 \* m \* Math.Abs(x \* x));

}

else

{

Math.Pow((x \* x + m), 2);

}

textBox4.Text = Convert.ToString(j);

break;

case 2:

if (-l < m && -l < x)

{

j = Math.Sin(5 \* Math.Exp(x) + 3 \* m \* Math.Abs(Math.Exp(x)));

}

else if (x > m)

{

j = Math.Cos(3 \* Math.Exp(x) + 5 \* m \* Math.Abs(Math.Exp(x)));

}

else

{

Math.Pow((Math.Exp(x) + m), 2);

}

textBox4.Text = Convert.ToString(j);

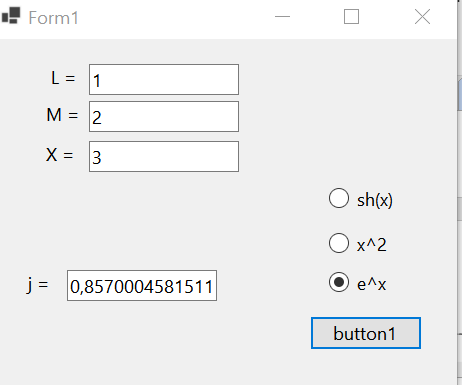
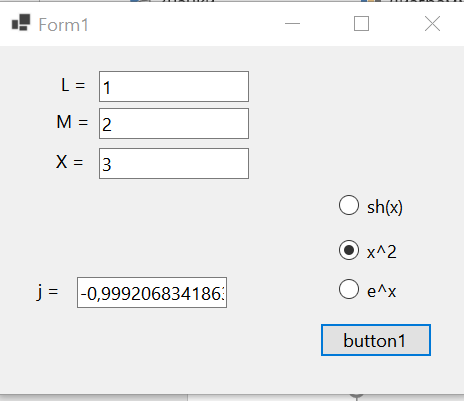
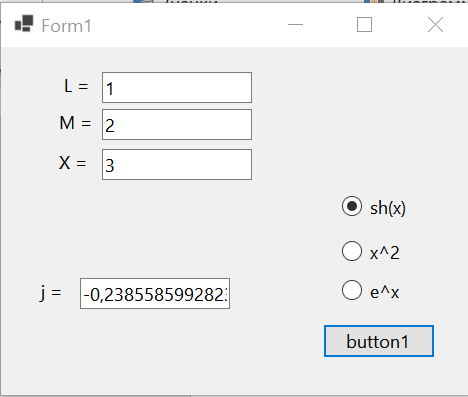
break;

default:

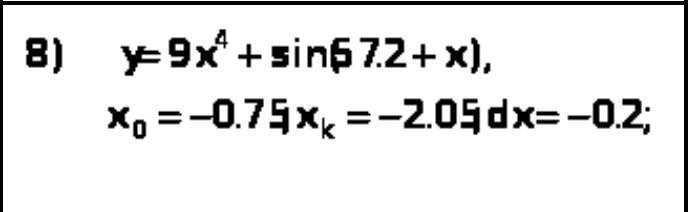
textBox4.Text += "Решение не найдено" + Environment.NewLine;

break;

}

}

2



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

{

double x0 = 0.75;

double xk = 2.05;

double dx = 0.2;

textBox1.Text = Convert.ToString(x0);

textBox2.Text = Convert.ToString(xk);

textBox3.Text = Convert.ToString(dx);

textBox4.Text = "ilusha1" + Environment.NewLine;

double x = x0;

while (x <= (xk + dx / 2))

{

double y = 9 \* x \* x \* x \* x + Math.Sin(57.2 + x);

textBox4.Text += "x=" + Convert.ToString(x) +

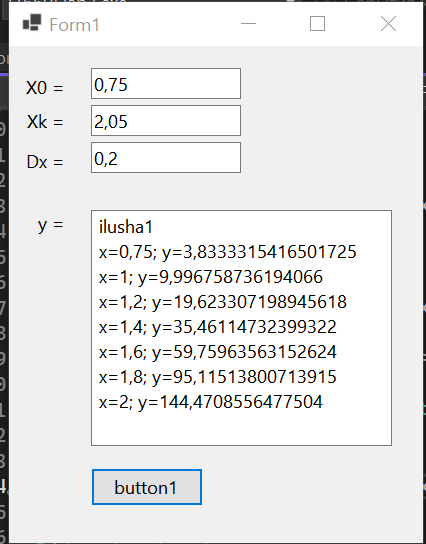
"; y=" + Convert.ToString(y) + Environment.NewLine;

x = x + dx;

x = Math.Round(x, 1);

}

}



3. **Дана строка символов. Вывести на экран только строчные русские буквы, входящие в эту строку.**

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

{

// Получаем номер выделенной строки

int index = listBox1.SelectedIndex;

// Считываем строку в перменную str

string str = (string)listBox1.Items[index];

// Узнаем количество символов в строке

int len = str.Length;

// Считаем, что количество пробелов равно 0

int count = 0;

// Устанавливаем счетчик символов в 0

int i = 0;

//Организуем цикл перебора всех символов в строке

while (i <= len - 1)

{

// Если нашли пробел, то увеличиваем

// счетчик пробелов на 1

if (str[i] >= 'а' && str[i] <= 'я')

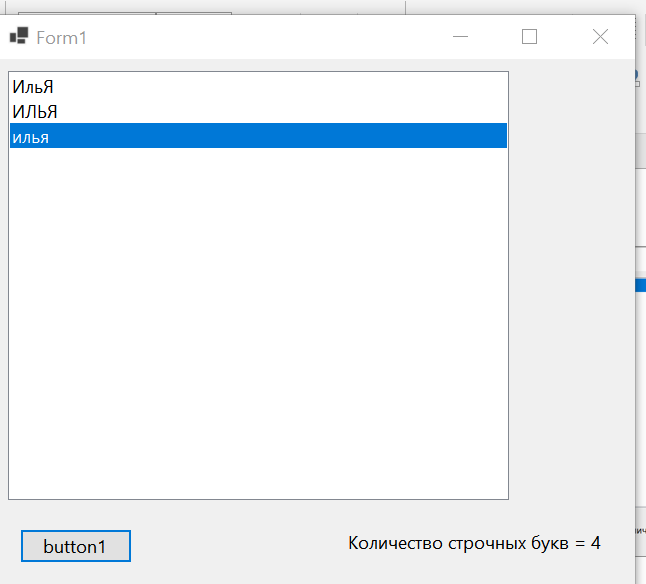
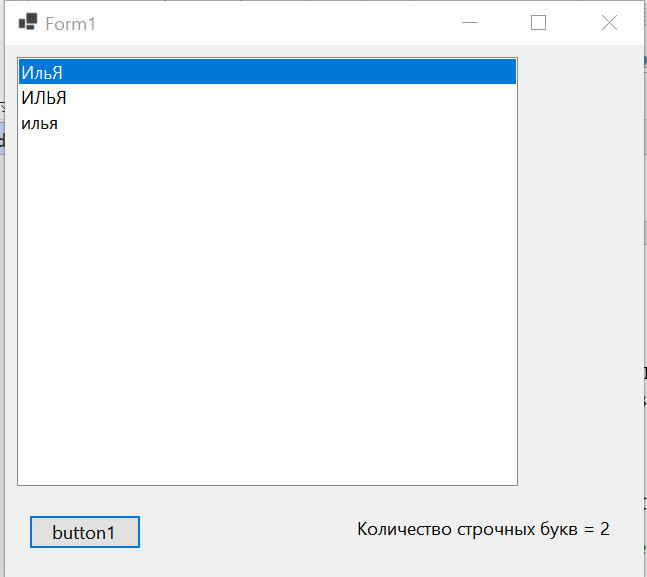
count++;

i++;

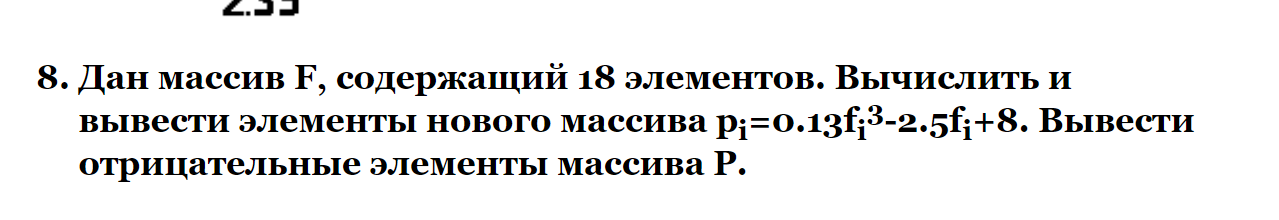
}

label1.Text = "Количество cтрочных букв = " +

count.ToString();

}

4



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

{

double[] P = new double[18];

double[] F = new double[18];

Random rand = new Random();

textBox1.Text = "";

for (int i = 0; i < 18; i++)

{

F[i] = rand.Next(0, 18);

textBox1.Text += "Mas[" + Convert.ToString(i) + "] = "

+ Convert.ToString(F[i]) + Environment.NewLine;

}

for (int i = 0; i < 18; i++) //i<18

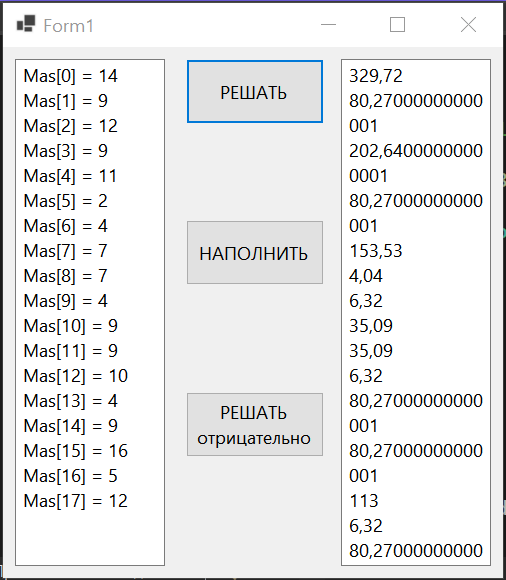
{

P[i] = 0.13 \* Math.Pow(F[i], 3) - 2.5 \* F[i] + 8.0;

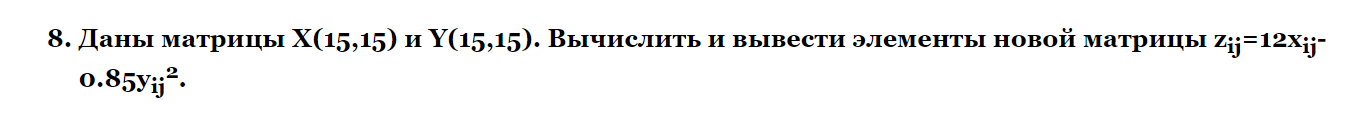
textBox2.Text += P[i] + Environment.NewLine;

}

}



5



Абрамян решебник



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

{

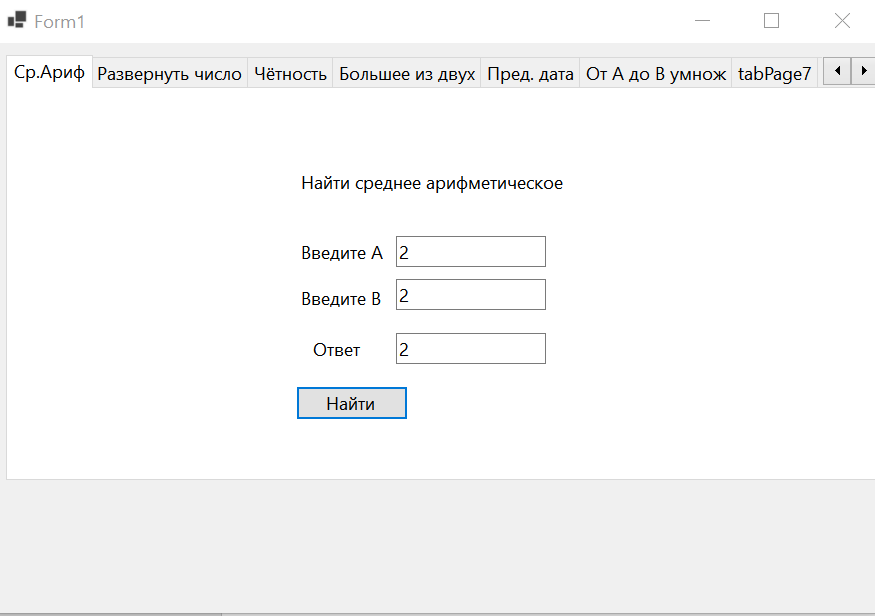
int a1 = Convert.ToInt32(textBox1.Text);

int b1 = Convert.ToInt32(textBox2.Text);

int res1 = (a1 + b1) / 2;

textBox3.Text = Convert.ToString(res1);

}





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

{

int input2 = Convert.ToInt32(textBox6.Text);

int res2 = 0;

while (input2 > 0)

{

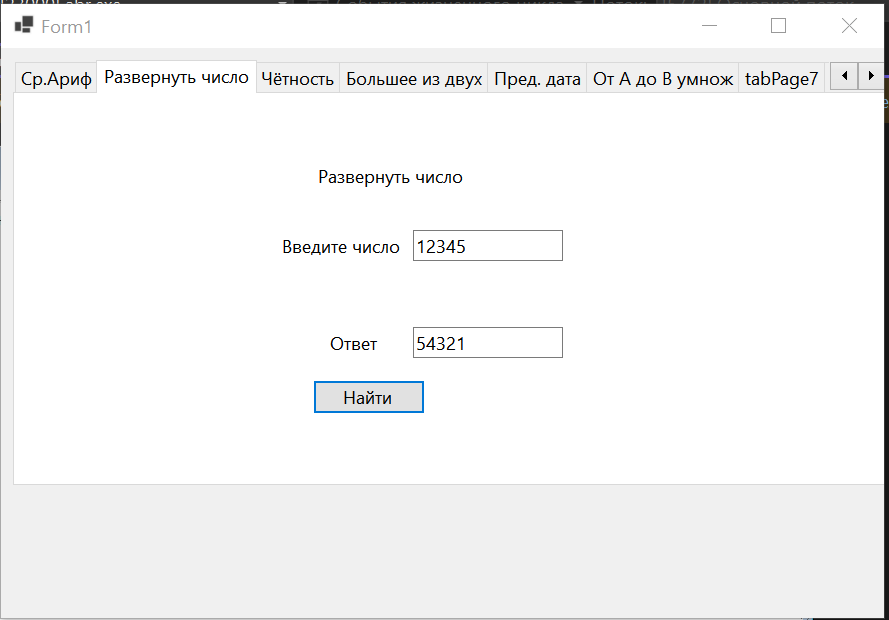
res2 \*= 10;

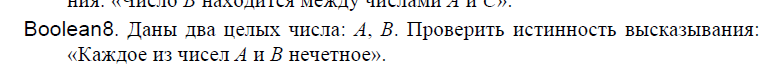
res2 += input2 % 10;

input2 /= 10;

}

textBox4.Text = Convert.ToString(res2);

}



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

{

int a3 = Convert.ToInt32(textBox7.Text);

int b3 = Convert.ToInt32(textBox8.Text);

if (a3 % 2 != 0 && b3 % 2 != 0)

{

textBox5.Text = "Верно";

}

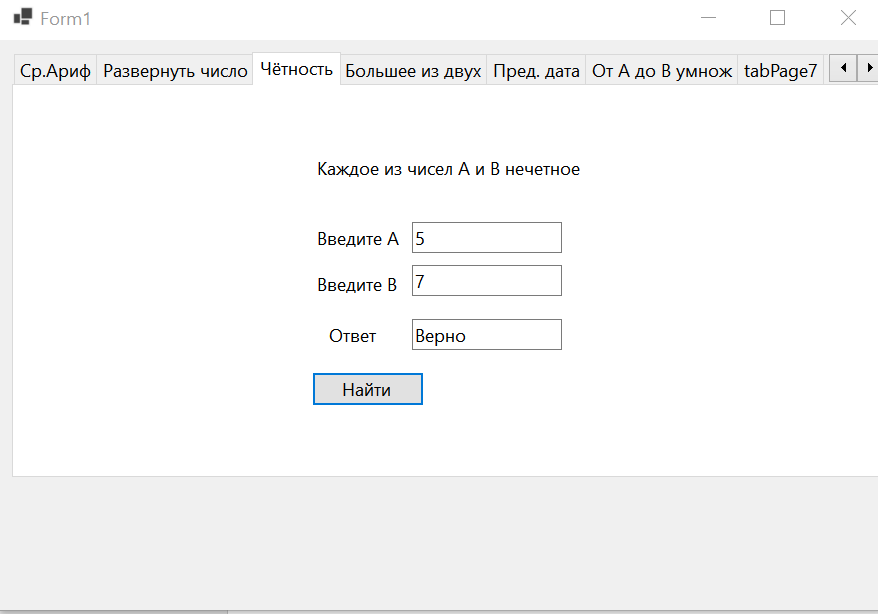
else

{

textBox5.Text = "Неверно";

}

}





private void button4\_Click(object sender, EventArgs e)

{

int a4 = Convert.ToInt32(textBox10.Text);

int b4 = Convert.ToInt32(textBox11.Text);

if (a4 > b4)

{

textBox9.Text = Convert.ToString(a4 + " " + b4);

}

else if (a4 < b4)

{

textBox9.Text = Convert.ToString(b4 + " " + a4);

}

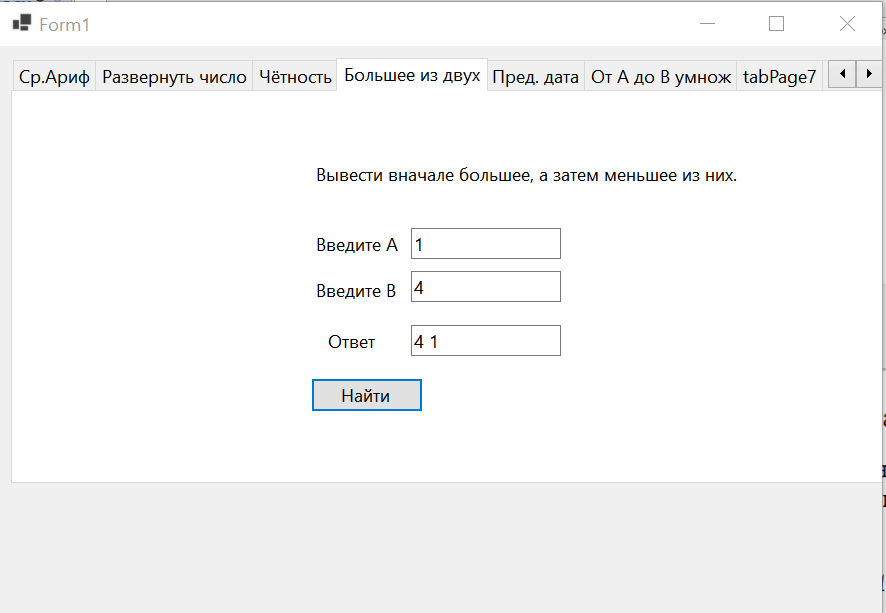
else

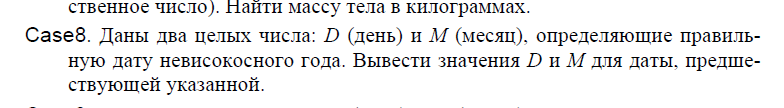
{

textBox9.Text = Convert.ToString("Равны");

}

}





private void button5\_Click(object sender, EventArgs e)

{

int d5 = Convert.ToInt32(textBox13.Text);

int m5 = Convert.ToInt32(textBox14.Text);

int day5 = 0;

int month5 = 0;

switch (m5)

{

case 1:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 1;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 12;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 2:

if (d5 > 1 && d5 <= 28)

{

day5 = d5 - 1;

month5 = 2;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 01;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 3:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 3;

}

else if (d5 == 1)

{

day5 = 28;

month5 = 02;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 4:

if (d5 > 1 && d5 <= 30)

{

day5 = d5 - 1;

month5 = 4;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 03;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 5:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 5;

}

else if (d5 == 1)

{

day5 = 30;

month5 = 04;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 6:

if (d5 > 1 && d5 <= 30)

{

day5 = d5 - 1;

month5 = 6;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 05;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 7:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 7;

}

else if (d5 == 1)

{

day5 = 30;

month5 = 06;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 8:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 8;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 07;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 9:

if (d5 > 1 && d5 <= 30)

{

day5 = d5 - 1;

month5 = 9;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 12;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 10:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 10;

}

else if (d5 == 1)

{

day5 = 30;

month5 = 09;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 11:

if (d5 > 1 && d5 <= 30)

{

day5 = d5 - 1;

month5 = 11;

}

else if (d5 == 1)

{

day5 = 31;

month5 = 10;

}

else

{

textBox12.Text = Convert.ToString("NaN");

}

break;

case 12:

if (d5 > 1 && d5 <= 31)

{

day5 = d5 - 1;

month5 = 12;

}

else if (d5 == 1)

{

day5 = 30;

month5 = 11;

}

else

{

textBox12.Text = Convert.ToString("NaN");

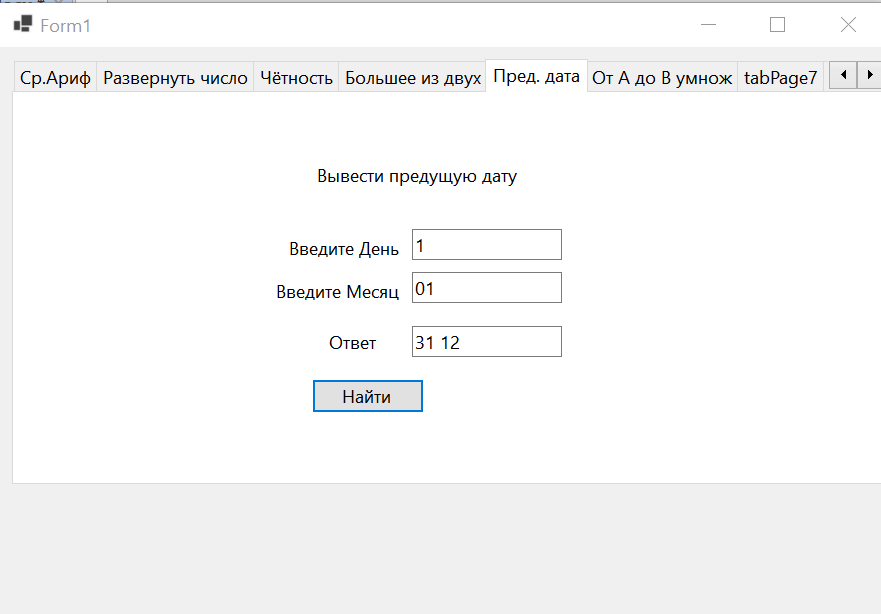
}

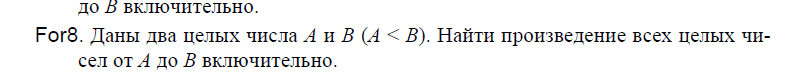
break;

}

textBox12.Text = Convert.ToString(day5 + " " + month5);

}





private void button6\_Click(object sender, EventArgs e)

{

try

{

int a6 = Convert.ToInt32(textBox16.Text);

int b6 = Convert.ToInt32(textBox17.Text);

BigInteger res6 = 1;

for (int i = a6; i <= b6; i++)

{

res6 \*= i;

}

textBox15.Text = Convert.ToString(res6);

}

catch (DivideByZeroException)

{ MessageBox.Show("Ошибка деления на ноль !!!"); }

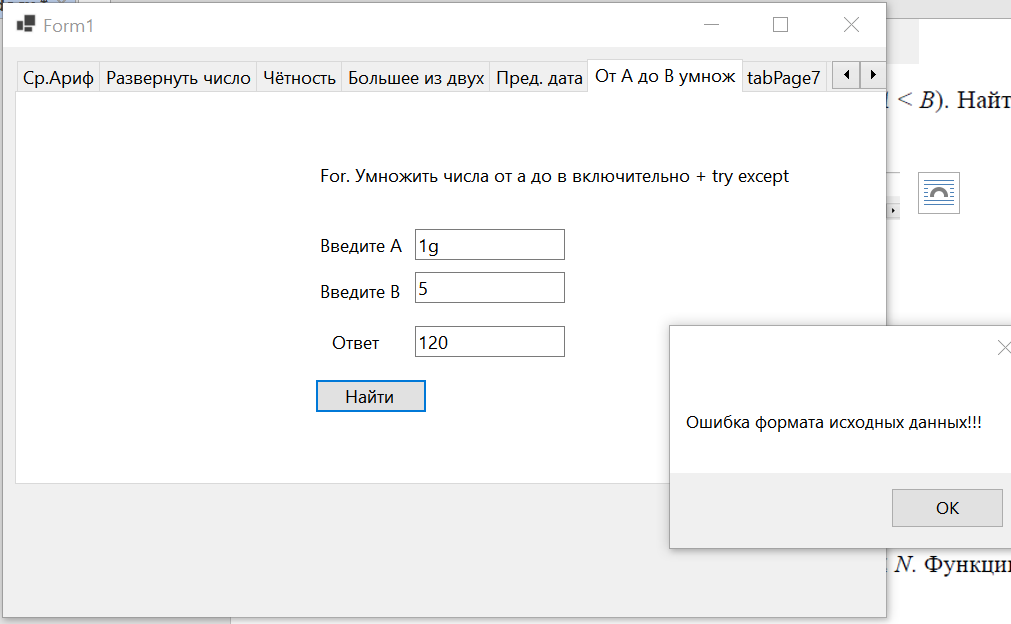
catch (FormatException)

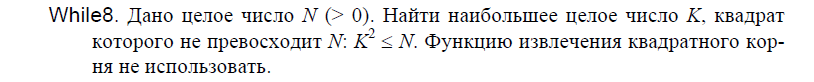
{ MessageBox.Show("Ошибка формата исходных данных!!!"); }

catch (Exception)

{ MessageBox.Show("Ошибка !!!"); }

}





private void button7\_Click(object sender, EventArgs e)

{

int n7= Convert.ToInt32(textBox19.Text);

int k7 = 0;

while ((k7 \* k7) <= n7)

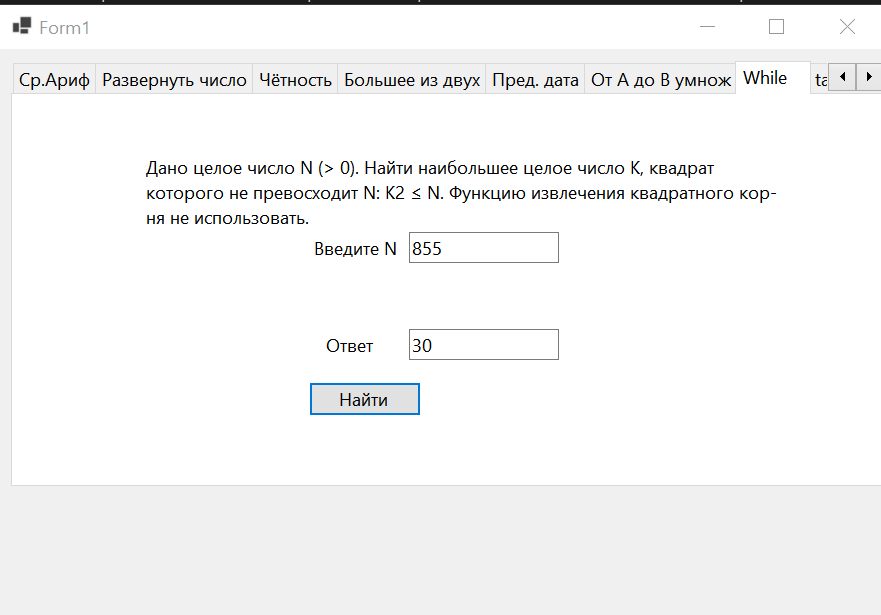
{

++k7;

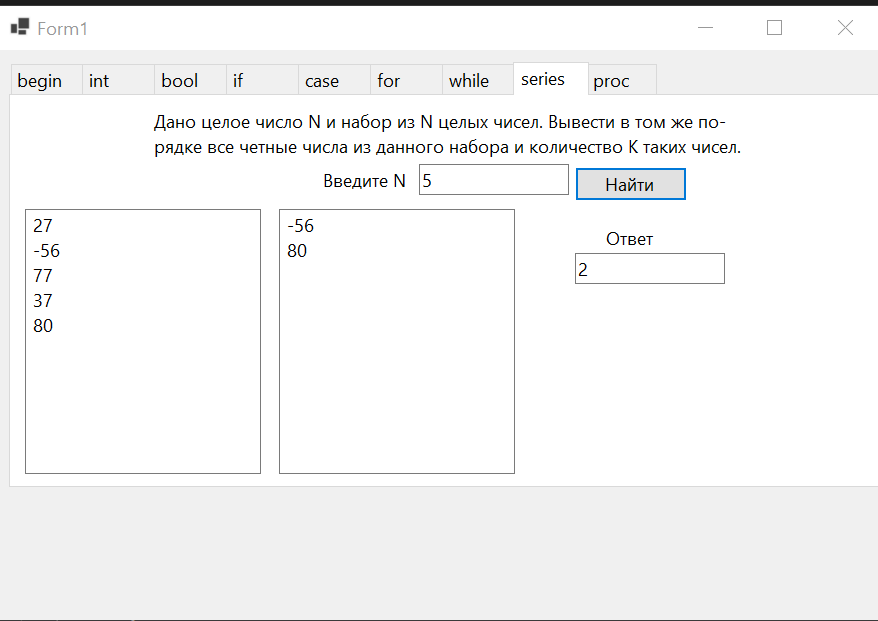
textBox18.Text = Convert.ToString(k7);

}

}







private void button8\_Click(object sender, EventArgs e)

{

Random rand = new Random();

int n = int.Parse(textBox21.Text);

int k = 0;

int[] mas = new int[n];

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

{

mas[i] = rand.Next(-100, 100);

textBox20.Text += mas[i] + Environment.NewLine;

}

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

{

if (mas[i] % 2 == 0)

{

textBox25.Text += mas[i] + Environment.NewLine;

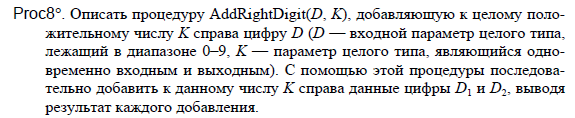
k++;

}

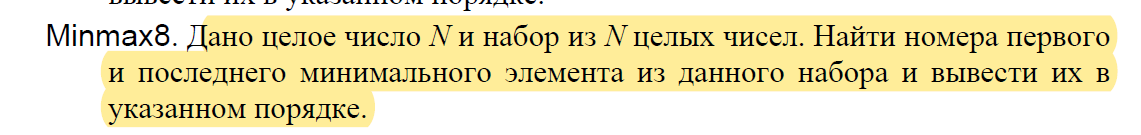
}

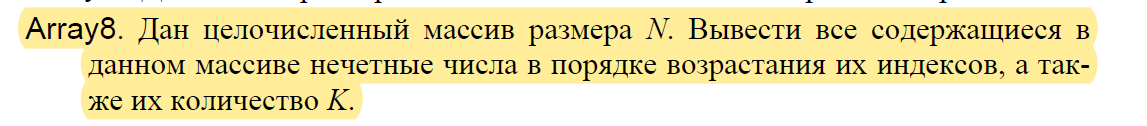
textBox24.Text += k.ToString();

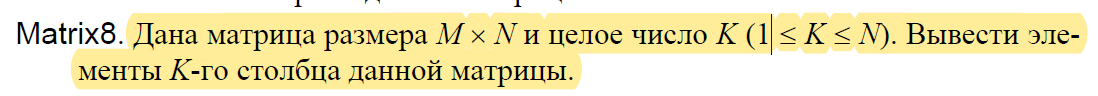
}

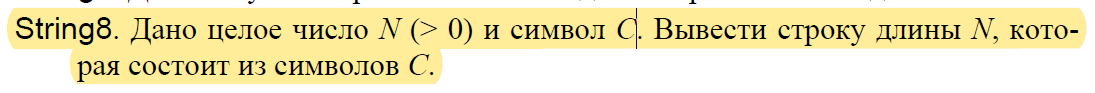


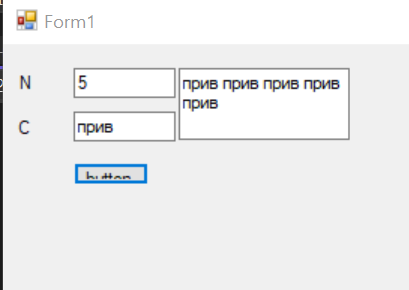
Абрамян 2











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

{

int N = Convert.ToInt32(textBox1.Text);

string C = Convert.ToString(textBox2.Text);

string res = "";

for (int i = 1; i <= N; i++)

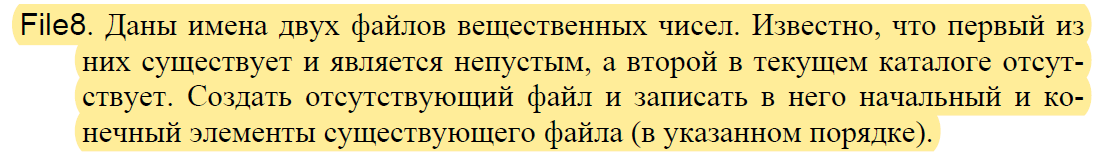
{

res = res + C;

}

textBox3.Text = res;

}



Тест

public Form1()

{

InitializeComponent();

this.tabControl1.Cursor = System.Windows.Forms.Cursors.Default;

listBox1.AllowDrop = true;

//подписываемся на события

listBox1.MouseMove += new MouseEventHandler(listBox1\_MouseMove);

listBox1.DragEnter += new DragEventHandler(listBox1\_DragEnter);

listBox1.DragDrop += new DragEventHandler(listBox1\_DragDrop);

textBox4.Text = "";

}

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

{

;

}

public double correct = 0;

public double incorrect = 0;

int indexToMove;

private void listBox1\_MouseMove(object sender, MouseEventArgs e)

{

//если нажата левая кнопка мыши, начинаем Drag&Drop

if (e.Button == MouseButtons.Left)

{

//индекс элемента, который мы перемещаем

indexToMove = listBox1.IndexFromPoint(e.X, e.Y);

listBox1.DoDragDrop(indexToMove, DragDropEffects.Move);

}

}

private void listBox1\_DragEnter(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.Move;

}

private void listBox1\_DragDrop(object sender, DragEventArgs e)

{

//индекс, куда перемещаем

//listBox1.PointToClient(new Point(e.X, e.Y)) - необходимо

//использовать поскольку в e храниться

//положение мыши в экранных коородинатах, а эта

//функция позволяет преобразовать в клиентские

int newIndex = listBox1.IndexFromPoint(listBox1.PointToClient(new Point(e.X, e.Y)));

//если вставка происходит в начало списка

if (newIndex == -1)

{

//получаем перетаскиваемый элемент

object itemToMove = listBox1.Items[indexToMove];

//удаляем элемент

listBox1.Items.RemoveAt(indexToMove);

//добавляем в конец списка

listBox1.Items.Add(itemToMove);

}

//вставляем где-то в середину списка

else if (indexToMove != newIndex)

{

//получаем перетаскиваемый элемент

object itemToMove = listBox1.Items[indexToMove];

//удаляем элемент

listBox1.Items.RemoveAt(indexToMove);

//вставляем в конкретную позицию

listBox1.Items.Insert(newIndex, itemToMove);

}

}

private void CloseButton()

{

this.Close();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

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

{

CloseButton();

}

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

{

CloseButton();

}

private void button14\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK)

{

correct += 1;

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

openFileDialog1.FileName = "text";

}

}

private void tabPage1\_Click(object sender, EventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

CloseButton();

}

private void button7\_Click(object sender, EventArgs e)

{

CloseButton();

}

private void button9\_Click(object sender, EventArgs e)

{

CloseButton();

}

private void button11\_Click(object sender, EventArgs e)

{

CloseButton();

}

private void button13\_Click(object sender, EventArgs e)

{

CloseButton();

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

if (radioButton1.Checked == true)

{

correct += 1;

}

else

{

incorrect += 1;

textBox4.Text += " 1";

}

tabControl1.SelectTab(1);

}

private void button4\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "2022")

{

correct += 1;

}

else

{

incorrect += 1;

textBox4.Text += " 2";

}

tabControl1.SelectTab(2);

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button6\_Click(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex == 0)

{

correct += 0.25;

}

if (comboBox2.SelectedIndex == 1)

{

correct += 0.25;

}

if (comboBox3.SelectedIndex == 2)

{

correct += 0.25;

}

if (comboBox4.SelectedIndex == 3)

{

correct += 0.25;

}

else

{

incorrect += 1;

textBox4.Text += " 3";

}

tabControl1.SelectTab(3);

}

private void button15\_Click(object sender, EventArgs e)

{

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

correct = 0;

incorrect = 0;

tabControl1.SelectTab(0);

}

private void button12\_Click(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

{

correct += 0.5;

}

if (checkBox2.Checked == true)

{

correct += 0.5;

}

else

{

incorrect += 1;

textBox4.Text += " 6";

}

tabControl1.SelectTab(6);

textBox2.Text = Convert.ToString(correct + " из 6");

textBox3.Text = Convert.ToString(incorrect + " из 6");

if (correct >= 6.0)

{

label21.Text = "5";

label21.ForeColor = Color.Green;

}

else if (correct >= 5.0)

{

label21.Text = "4";

label21.ForeColor = Color.GreenYellow;

}

else if (correct >= 4.0)

{

label21.Text = "3";

label21.ForeColor = Color.Yellow;

}

else if (correct >= 3.0)

{

label21.Text = "2";

label21.ForeColor = Color.Red;

}

else if (correct >= 2.0)

{

label21.Text = "1";

label21.ForeColor = Color.DarkRed;

}

else if (correct >= 1.0)

{

label21.Text = "1";

label21.ForeColor = Color.IndianRed;

}

else if (correct == 0)

{

label21.Text = "0";

label21.ForeColor = Color.OrangeRed;

}

}

private void tabPage6\_Click(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button8\_Click(object sender, EventArgs e)

{

if (listBox1.Items[0] == "1" &&

listBox1.Items[1] == "2" &&

listBox1.Items[2] == "3" &&

listBox1.Items[3] == "4" &&

listBox1.Items[4] == "5")

{

correct += 1;

}

else

{

incorrect += 1;

textBox4.Text += " 4";

}

tabControl1.SelectTab(4);

}

private void button10\_Click(object sender, EventArgs e)

{

if (openFileDialog1.FileName != "text")

{

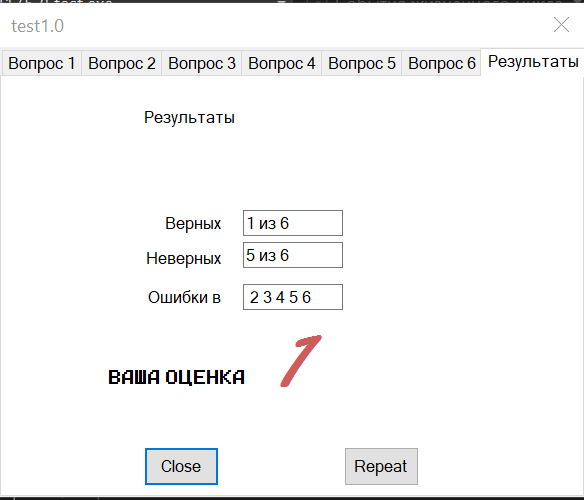
textBox4.Text += " 5";

incorrect += 1;

}

tabControl1.SelectTab(5);

}



Игра висельчак

using hangmanGame.Properties;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace hangmanGame

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

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

{

}

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

{

//Задаем колво окошек = колво букв

pictureBox0.Visible = true;

pictureBox1.Visible = false;

pictureBox2.Visible = false;

pictureBox3.Visible = false;

pictureBox4.Visible = false;

pictureBox5.Visible = false;

}

public string word;

private void fileScript()

{

string[] mas = new string[5] {"ПРИВЕТ", "САЛАМ", "ЗДАРОВА", "КУ", "ДРАТУТИ"};

Random rand = new Random();

int ran = rand.Next(0, mas.Length);

word = mas[ran];

wordVisible(mas[ran]);

}

//Меню верхнее

//Игра

//Новая игра...

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

{

}

//Продолжить...

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

{

}

//Звуки

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

{

}

//Статистика...

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

{

}

//Настройки...

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

{

}

//Выход

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

{

}

//Справка

//Центр справки и поддержки

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

{

}

//Поддержать проект...

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

{

}

//Сообщить об ошибке...

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

{

}

//О программе...

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

{

FormAbout newForm = new FormAbout();

newForm.Show();

}

//Проверить обновление...

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

{

}

//Слово

//Счётчик всех введённых букв

public int all = 0;

//Счётчик ошибок

public int incorrect = 0;

//Счётчик на конец слова и победу

public int i = 0;

public bool res = false;

//Проверка введённых букв

private void keyboard(string letter)

{

res = false;

if (0 < word.Length && Convert.ToString(word[0]).Contains(letter) && labelW1.Text == "\_")

{

labelW1.Text = Convert.ToString(letter);

i++;

res = true;

}

if (1 < word.Length && Convert.ToString(word[1]).Contains(letter) && labelW2.Text == "\_")

{

labelW2.Text = Convert.ToString(letter);

i++;

res = true;

}

if (2 < word.Length && Convert.ToString(word[2]).Contains(letter) && labelW3.Text == "\_")

{

labelW3.Text = Convert.ToString(letter);

i++;

res = true;

}

if (3 < word.Length && Convert.ToString(word[3]).Contains(letter) && labelW4.Text == "\_")

{

labelW4.Text = Convert.ToString(letter);

i++;

res = true;

}

if (4 < word.Length && Convert.ToString(word[4]).Contains(letter) && labelW5.Text == "\_")

{

labelW5.Text = Convert.ToString(letter);

i++;

res = true;

}

if (5 < word.Length && Convert.ToString(word[5]).Contains(letter) && labelW6.Text == "\_")

{

labelW6.Text = Convert.ToString(letter);

i++;

res = true;

}

if (6 < word.Length && Convert.ToString(word[6]).Contains(letter) && labelW7.Text == "\_")

{

labelW7.Text = Convert.ToString(letter);

i++;

res = true;

}

if (7 < word.Length && Convert.ToString(word[7]).Contains(letter) && labelW8.Text == "\_")

{

labelW8.Text = Convert.ToString(letter);

i++;

res = true;

}

if (8 < word.Length && Convert.ToString(word[8]).Contains(letter) && labelW9.Text == "\_")

{

labelW9.Text = Convert.ToString(letter);

i++;

res = true;

}

if (9 < word.Length && Convert.ToString(word[9]).Contains(letter) && labelW10.Text == "\_")

{

labelW10.Text = Convert.ToString(letter);

i++;

res = true;

}

if (10 < word.Length && Convert.ToString(word[10]).Contains(letter) && labelW11.Text == "\_")

{

labelW11.Text = Convert.ToString(letter);

i++;

res = true;

}

//Счётчик нажатых кнопок

all++;

incorrect = all - i;

if (incorrect > 0) pictureBox1.Visible = true; //pictureBox0.Visible = false;

if (incorrect > 1) pictureBox2.Visible = true; //pictureBox1.Visible = false;

if (incorrect > 2) pictureBox3.Visible = true; //pictureBox2.Visible = false;

if (incorrect > 3) pictureBox4.Visible = true; //pictureBox3.Visible = false;

if (incorrect > 4) pictureBox5.Visible = true; //pictureBox4.Visible = false;

//Если выиграл

if (i == word.Length)

{

MessageBox.Show("Win");

}

//Если проиграл

if (incorrect > 4)

{

MessageBox.Show("Lose");

}

}

//Окошек столько, сколько букв

private void wordVisible(string word)

{

int length = word.Length;

if (length < 1) labelW1.Visible = false;

if (length < 2) labelW2.Visible = false;

if (length < 3) labelW3.Visible = false;

if (length < 4) labelW4.Visible = false;

if (length < 5) labelW5.Visible = false;

if (length < 6) labelW6.Visible = false;

if (length < 7) labelW7.Visible = false;

if (length < 8) labelW8.Visible = false;

if (length < 9) labelW9.Visible = false;

if (length < 10) labelW10.Visible = false;

if (length < 11) labelW11.Visible = false;

if (length > 1) labelW1.Visible = true;

if (length > 2) labelW2.Visible = true;

if (length > 3) labelW3.Visible = true;

if (length > 4) labelW4.Visible = true;

if (length > 5) labelW5.Visible = true;

if (length > 6) labelW6.Visible = true;

if (length > 7) labelW7.Visible = true;

if (length > 8) labelW8.Visible = true;

if (length > 9) labelW9.Visible = true;

if (length > 10) labelW10.Visible = true;

if (length > 11) labelW11.Visible = true;

}

//Нажатия кнопок клавиатуры

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

{

keyboard("А");

buttonA.BackColor= Color.LightGray;

if (res == true) buttonA.BackColor = Color.YellowGreen; buttonA.Enabled = false;

}

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

{

keyboard("Б");

buttonB.BackColor = Color.LightGray;

if (res == true) buttonB.BackColor = Color.YellowGreen; buttonB.Enabled = false;

}

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

{

keyboard("В");

buttonV.BackColor = Color.LightGray;

if (res == true) buttonV.BackColor = Color.YellowGreen; buttonV.Enabled = false;

}

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

{

keyboard("Г");

buttonG.BackColor = Color.LightGray;

if (res == true) buttonG.BackColor = Color.YellowGreen; buttonG.Enabled = false;

}

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

{

keyboard("Д");

buttonD.BackColor = Color.LightGray;

if (res == true) buttonD.BackColor = Color.YellowGreen; buttonD.Enabled = false;

}

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

{

keyboard("Е");

buttonE.BackColor = Color.LightGray;

if (res == true) buttonE.BackColor = Color.YellowGreen; buttonE.Enabled = false;

}

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

{

keyboard("Ё");

buttonYo.BackColor = Color.LightGray;

if (res == true) buttonYo.BackColor = Color.YellowGreen; buttonYo.Enabled = false;

}

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

{

keyboard("Ж");

buttonJ.BackColor = Color.LightGray;

if (res == true) buttonJ.BackColor = Color.YellowGreen; buttonJ.Enabled = false;

}

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

{

keyboard("З");

buttonZ.BackColor = Color.LightGray;

if (res == true) buttonZ.BackColor = Color.YellowGreen; buttonZ.Enabled = false;

}

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

{

keyboard("И");

buttonI.BackColor = Color.LightGray;

if (res == true) buttonI.BackColor = Color.YellowGreen; buttonI.Enabled = false;

}

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

{

keyboard("Й");

buttonIK.BackColor = Color.LightGray;

if (res == true) buttonIK.BackColor = Color.YellowGreen; buttonIK.Enabled = false;

}

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

{

keyboard("К");

buttonK.BackColor = Color.LightGray;

if (res == true) buttonK.BackColor = Color.YellowGreen; buttonK.Enabled = false;

}

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

{

keyboard("Л");

buttonL.BackColor = Color.LightGray;

if (res == true) buttonL.BackColor = Color.YellowGreen; buttonL.Enabled = false;

}

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

{

keyboard("М");

buttonM.BackColor = Color.LightGray;

if (res == true) buttonM.BackColor = Color.YellowGreen; buttonM.Enabled = false;

}

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

{

keyboard("Н");

buttonN.BackColor = Color.LightGray;

if (res == true) buttonN.BackColor = Color.YellowGreen; buttonN.Enabled = false;

}

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

{

keyboard("О");

buttonO.BackColor = Color.LightGray;

if (res == true) buttonO.BackColor = Color.YellowGreen; buttonO.Enabled = false;

}

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

{

keyboard("П");

buttonP.BackColor = Color.LightGray;

if (res == true) buttonP.BackColor = Color.YellowGreen; buttonP.Enabled = false;

}

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

{

keyboard("Р");

buttonR.BackColor = Color.LightGray;

if (res == true) buttonR.BackColor = Color.YellowGreen; buttonR.Enabled = false;

}

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

{

keyboard("С");

buttonS.BackColor = Color.LightGray;

if (res == true) buttonS.BackColor = Color.YellowGreen; buttonS.Enabled = false;

}

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

{

keyboard("Т");

buttonT.BackColor = Color.LightGray;

if (res == true) buttonT.BackColor = Color.YellowGreen; buttonT.Enabled = false;

}

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

{

keyboard("У");

buttonY.BackColor = Color.LightGray;

if (res == true) buttonY.BackColor = Color.YellowGreen; buttonY.Enabled = false;

}

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

{

keyboard("Ф");

buttonF.BackColor = Color.LightGray;

if (res == true) buttonF.BackColor = Color.YellowGreen; buttonF.Enabled = false;

}

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

{

keyboard("Х");

buttonX.BackColor = Color.LightGray;

if (res == true) buttonX.BackColor = Color.YellowGreen; buttonX.Enabled = false;

}

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

{

keyboard("Ц");

buttonTS.BackColor = Color.LightGray;

if (res == true) buttonTS.BackColor = Color.YellowGreen; buttonTS.Enabled = false;

}

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

{

keyboard("Ч");

buttonCH.BackColor = Color.LightGray;

if (res == true) buttonCH.BackColor = Color.YellowGreen; buttonCH.Enabled = false;

}

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

{

keyboard("Ш");

buttonSHE.BackColor = Color.LightGray;

if (res == true) buttonSHE.BackColor = Color.YellowGreen; buttonSHE.Enabled = false;

}

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

{

keyboard("Щ");

buttonSHEE.BackColor = Color.LightGray;

if (res == true) buttonSHEE.BackColor = Color.YellowGreen; buttonSHEE.Enabled = false;

}

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

{

keyboard("Ъ");

buttonTznak.BackColor = Color.LightGray;

if (res == true) buttonTznak.BackColor = Color.YellowGreen; buttonTznak.Enabled = false;

}

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

{

keyboard("Ы");

buttonYI.BackColor = Color.LightGray;

if (res == true) buttonYI.BackColor = Color.YellowGreen; buttonYI.Enabled = false;

}

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

{

keyboard("Ь");

buttonMZnak.BackColor = Color.LightGray;

if (res == true) buttonMZnak.BackColor = Color.YellowGreen; buttonMZnak.Enabled = false;

}

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

{

keyboard("Э");

buttonEE.BackColor = Color.LightGray;

if (res == true) buttonEE.BackColor = Color.YellowGreen; buttonEE.Enabled = false;

}

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

{

keyboard("Ю");

buttonU.BackColor = Color.LightGray;

if (res == true) buttonU.BackColor = Color.YellowGreen; buttonU.Enabled = false;

}

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

{

keyboard("Я");

buttonYa.BackColor = Color.LightGray;

if (res == true) buttonYa.BackColor = Color.YellowGreen; buttonYa.Enabled = false;

}

//Продолжить

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

{

fileScript();

labelW1.Text = "\_";

labelW2.Text = "\_";

labelW3.Text = "\_";

labelW4.Text = "\_";

labelW5.Text = "\_";

labelW6.Text = "\_";

labelW7.Text = "\_";

labelW8.Text = "\_";

labelW9.Text = "\_";

labelW10.Text = "\_";

labelW11.Text = "\_";

all = 0;

incorrect = 0;

i = 0;

res = false;

}

//Назвать слово целиком

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

{

}

//Пропустить слово

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

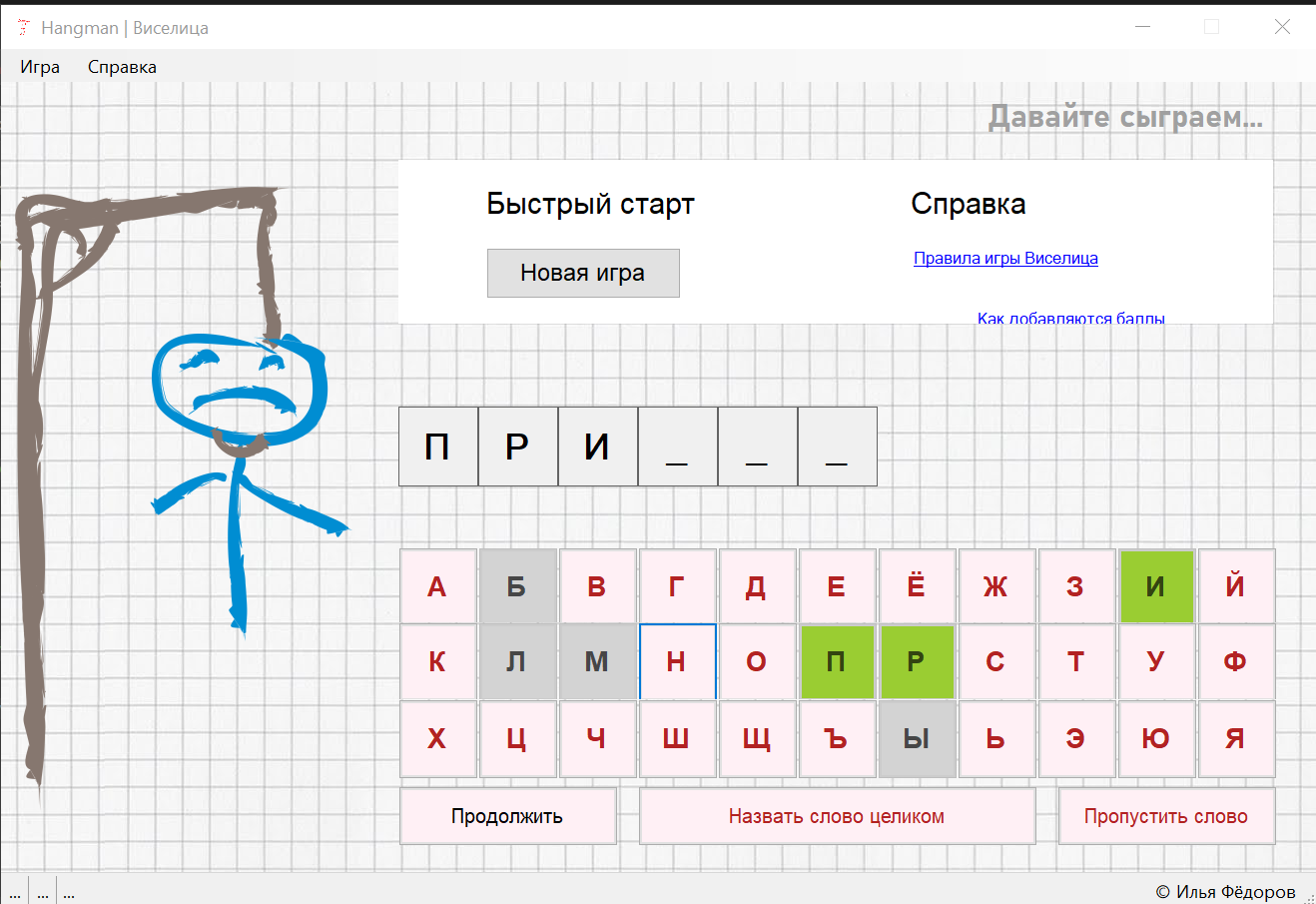
{

}

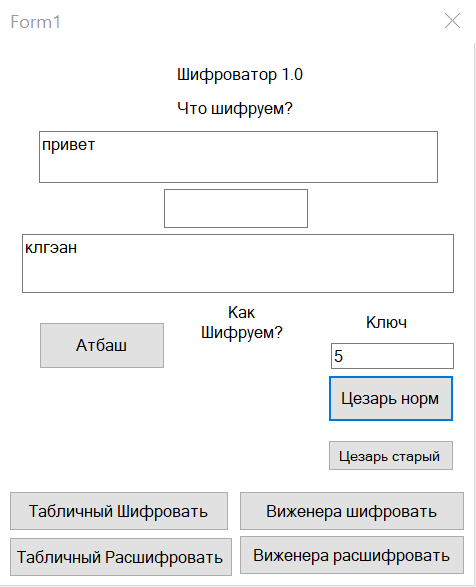
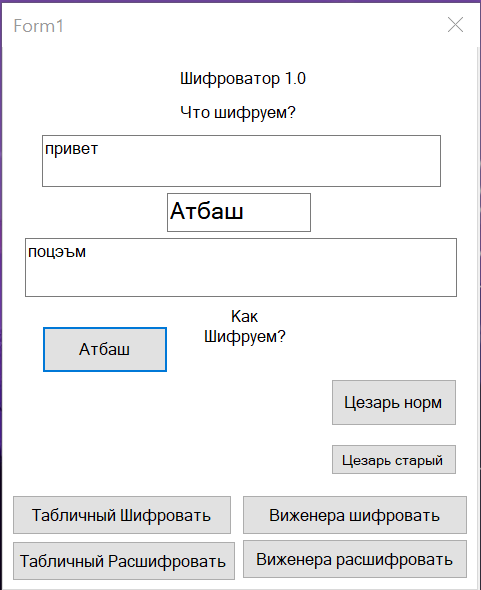
//

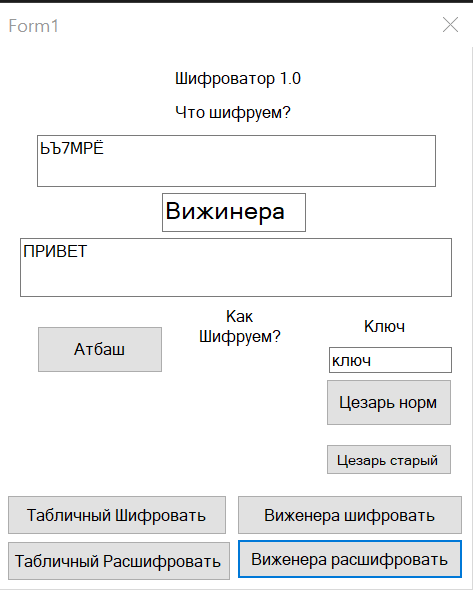
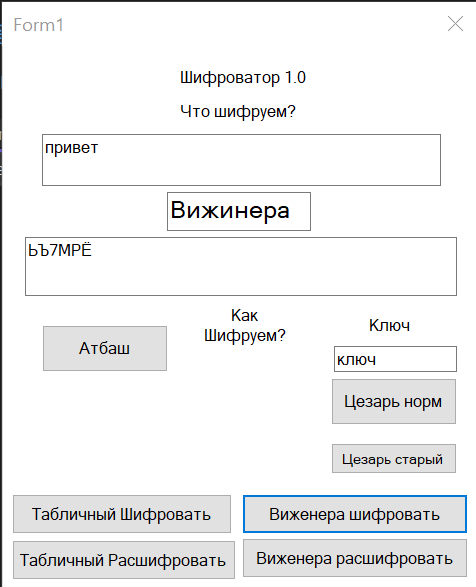
}

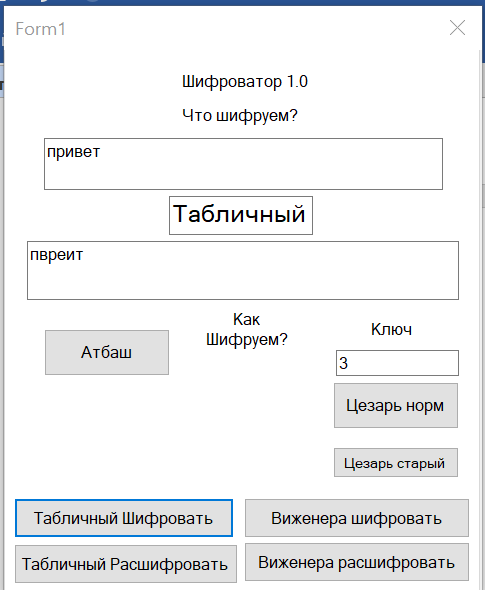
}



Шифрование







using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.Remoting.Messaging;

using System.Security;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace encryption

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

public string alphabet\_rus\_small = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

public string alphabet\_rus\_big = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

public string alphabet\_eng\_small = "zyxwvutsrqponmlkjihgfedcba";

public string alphabet\_eng\_big = "ZYXWVUTSRQPONMLKJIHGFEDCBA";

char[] alfavit = { 'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я' };

char[] characters = new char[] { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И',

'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т',

'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ы', 'Ъ',

'Э', 'Ю', 'Я', ' ', '1', '2', '3', '4', '5', '6',

'7', '8', '9', '0' };

private void atbash()

{

char[] characters = textBox1.Text.ToCharArray();

string temp\_word = "";

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

{

if (alphabet\_rus\_small.Contains(characters[i])) temp\_word += alphabet\_rus\_small[32 - alphabet\_rus\_small.IndexOf(characters[i])];

if (alphabet\_rus\_big.Contains(characters[i])) temp\_word += alphabet\_rus\_big[32 - alphabet\_rus\_big.IndexOf(characters[i])];

if (alphabet\_eng\_small.Contains(characters[i])) temp\_word += alphabet\_eng\_small[25 - alphabet\_eng\_small.IndexOf(characters[i])];

if (alphabet\_eng\_big.Contains(characters[i])) temp\_word += alphabet\_eng\_big[25 - alphabet\_eng\_big.IndexOf(characters[i])];

}

textBox5.Text = temp\_word;

}

private void caesar()

{

char[] massage = textBox1.Text.ToCharArray();

string s = "";

int d;

try

{

d = Convert.ToInt16(textBox2.Text);

}

catch(FormatException)

{

textBox2.Text = "Ошибка формата. Введите цифру";

}

int nomer; // Номер в алфавите

int j; // Переменная для циклов

// Перебираем каждый символ сообщения

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

{

// Ищем индекс буквы

for (j = 0; j < alfavit.Length; j++)

{

if (massage[i] == alfavit[j])

{

break;

}

}

if (j != 33) // Если j равно 33, значит символ не из алфавита

{

nomer = j; // Индекс буквы

d = nomer + 3; // Делаем смещение

// Проверяем, чтобы не вышли за пределы алфавита

if (d > 32)

{

d = d - 33;

}

massage[i] = alfavit[d]; // Меняем букву

}

}

s = new string(massage); // Собираем символы обратно в строку.

textBox5.Text = s;

}

string caesarDecode(string word, int key)

{

word = word.ToLower();

StringBuilder sb = new StringBuilder();

foreach (char character in word)

{

char ch = character;

ch = (char)((int)ch - key);

if (ch >= 'я')

ch = (char)(ch + 32);

sb.Append(ch.ToString().ToLower());

}

textBox5.Text = sb.ToString();

return sb.ToString();

}

private string vijinerEncode(string input, string keyword)

{

int N = characters.Length;

input = input.ToUpper();

keyword = keyword.ToUpper();

string result = "";

int keyword\_index = 0;

foreach (char symbol in input)

{

int c = (Array.IndexOf(characters, symbol) +

Array.IndexOf(characters, keyword[keyword\_index])) % N;

result += characters[c];

keyword\_index++;

if ((keyword\_index + 1) == keyword.Length)

keyword\_index = 0;

}

textBox5.Text = result;

return result;

}

private string vijinerDecode(string input, string keyword)

{

int N = characters.Length;

input = input.ToUpper();

keyword = keyword.ToUpper();

string result = "";

int keyword\_index = 0;

foreach (char symbol in input)

{

int p = (Array.IndexOf(characters, symbol) + N -

Array.IndexOf(characters, keyword[keyword\_index])) % N;

result += characters[p];

keyword\_index++;

if ((keyword\_index + 1) == keyword.Length)

keyword\_index = 0;

}

textBox5.Text = result;

return result;

}

public string skitalaEncrypt(string text, int d)

{

var k = text.Length % d;

if (k > 0)

{

//дополняем строку пробелами

text += new string(' ', d - k);

}

var column = text.Length / d;

var result = "";

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

{

for (int j = 0; j < d; j++)

{

result += text[i + column \* j].ToString();

}

}

textBox5.Text = result;

return result;

}

public string skitalaDecrypt(string text, int d)

{

var column = text.Length / d;

var symbols = new char[text.Length];

int index = 0;

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

{

for (int j = 0; j < d; j++)

{

symbols[i + column \* j] = text[index];

index++;

}

}

return string.Join("", symbols);

}

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

{

label4.Visible = false;

textBox2.Visible = false;

textBox3.Text = "Атбаш";

atbash();

}

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

{

textBox3.Text = "Цезаря (N)";

caesar();

}

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

{

textBox3.Text = "Вижинера Зашифровка";

textBox5.Text = vijinerEncode(textBox1.Text, textBox2.Text);

}

private void button4\_Click(object sender, EventArgs e)

{

textBox3.Text = "Табличный Зашифровка";

textBox5.Text = skitalaEncrypt(textBox1.Text, Convert.ToInt16(textBox2.Text));

}

private void tabPage1\_Click(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void button5\_Click(object sender, EventArgs e)

{

caesarDecode(textBox1.Text, Convert.ToInt32(textBox2.Text));

}

private void button6\_Click(object sender, EventArgs e)

{

textBox3.Text = "Вижинера Расшифровка";

vijinerDecode(textBox1.Text, textBox2.Text);

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

textBox3.Text = "Табличный Расшифровка";

textBox5.Text = skitalaDecrypt(textBox1.Text, Convert.ToInt16(textBox2.Text));

}

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

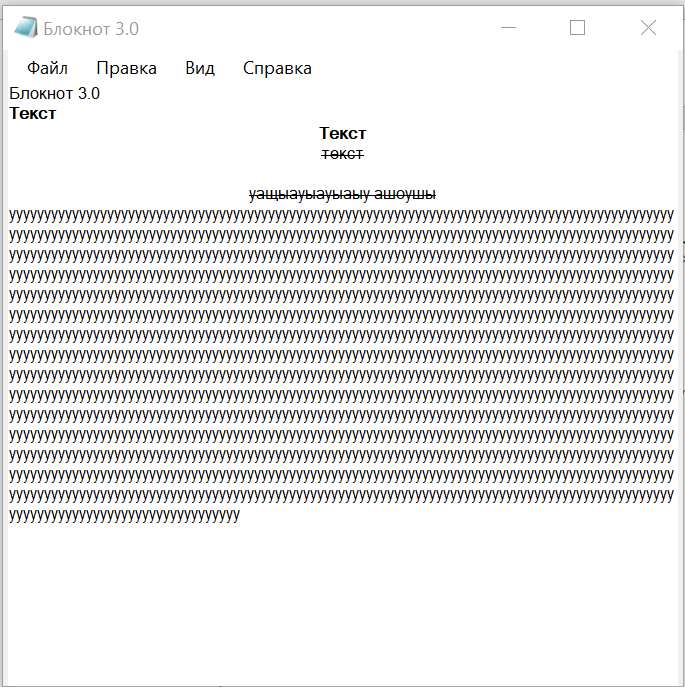
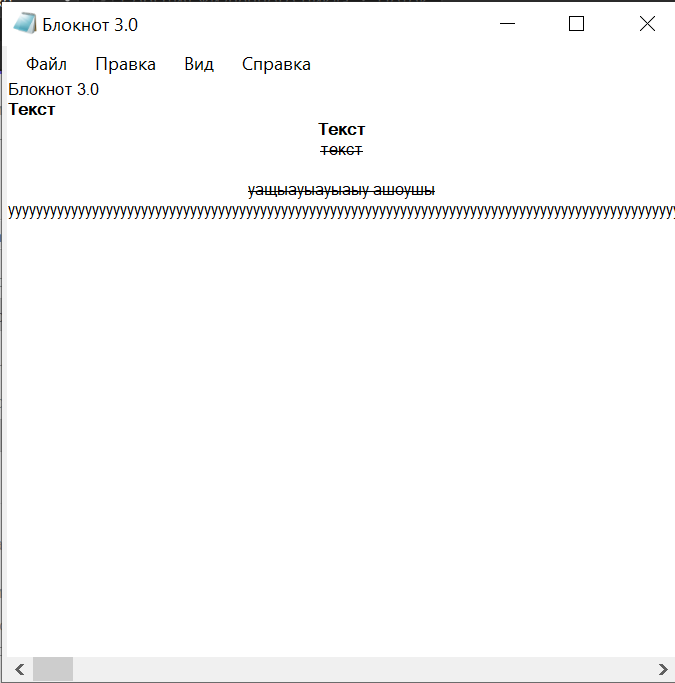
{

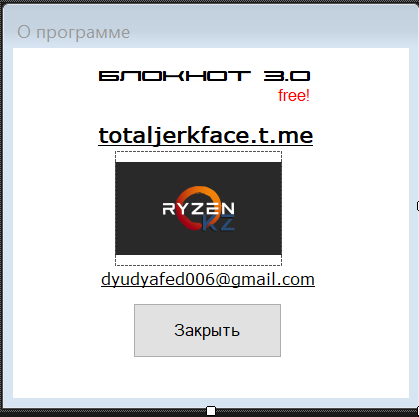
}

}

}

Блокнот





Форма 1

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.IO;

using System.Drawing.Printing;

namespace simpleNotepad3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

/// <summary>

/// Установка отметки строк меню Font->CharacterStyle

/// </summary>

private void CheckMenuFontCharacterStyle()

{

if (richTextBox1.SelectionFont.Bold == true)

{

menuFormatFontCharacterStyleBold.Checked = true;

}

else

{

menuFormatFontCharacterStyleBold.Checked = false;

}

if (richTextBox1.SelectionFont.Italic == true)

{

menuFormatFontCharacterStyleItalic.Checked = true;

}

else

{

menuFormatFontCharacterStyleItalic.Checked = false;

}

if (richTextBox1.SelectionFont.Underline == true)

{

menuFormatFontCharacterStyleUnderline.Checked = true;

}

else

{

menuFormatFontCharacterStyleUnderline.Checked = false;

}

if (richTextBox1.SelectionFont.Strikeout == true)

{

зачеркнутый.Checked = true;

}

else

{

зачеркнутый.Checked = false;

}

}

/// <summary>

/// Установка стиля символов Bold

/// </summary>

private void SetBold()

{

if (richTextBox1.SelectionFont != null)

{

System.Drawing.Font currentFont = richTextBox1.SelectionFont;

System.Drawing.FontStyle newFontStyle;

if (richTextBox1.SelectionFont.Bold == true)

{

newFontStyle = FontStyle.Regular;

}

else

{

newFontStyle = FontStyle.Bold;

}

richTextBox1.SelectionFont = new Font(

currentFont.FontFamily, currentFont.Size, newFontStyle);

CheckMenuFontCharacterStyle();

}

}

/// <summary>

/// Установка стиля символов Italic

/// </summary>

private void SetItalic()

{

if (richTextBox1.SelectionFont != null)

{

System.Drawing.Font currentFont = richTextBox1.SelectionFont;

System.Drawing.FontStyle newFontStyle;

CheckMenuFontCharacterStyle();

if (richTextBox1.SelectionFont.Italic == true)

{

newFontStyle = FontStyle.Regular;

}

else

{

newFontStyle = FontStyle.Italic;

}

richTextBox1.SelectionFont = new Font(

currentFont.FontFamily, currentFont.Size, newFontStyle);

CheckMenuFontCharacterStyle();

}

}

/// <summary>

/// Установка стиля символов Underline

/// </summary>

private void SetUnderline()

{

if (richTextBox1.SelectionFont != null)

{

System.Drawing.Font currentFont = richTextBox1.SelectionFont;

System.Drawing.FontStyle newFontStyle;

CheckMenuFontCharacterStyle();

if (richTextBox1.SelectionFont.Underline == true)

{

newFontStyle = FontStyle.Regular;

}

else

{

newFontStyle = FontStyle.Underline;

}

richTextBox1.SelectionFont = new Font(

currentFont.FontFamily, currentFont.Size, newFontStyle);

CheckMenuFontCharacterStyle();

}

}

/// <summary>

/// Установка стиля символов Strikeout

/// </summary>

private void SetStrikeout()

{

if (richTextBox1.SelectionFont != null)

{

System.Drawing.Font currentFont = richTextBox1.SelectionFont;

System.Drawing.FontStyle newFontStyle;

if (richTextBox1.SelectionFont.Strikeout == true)

{

newFontStyle = FontStyle.Regular;

}

else

{

newFontStyle = FontStyle.Strikeout;

}

richTextBox1.SelectionFont = new Font(

currentFont.FontFamily, currentFont.Size, newFontStyle);

CheckMenuFontCharacterStyle();

}

}

/// <summary>

/// StringReader для печати содержимого редактора текста

/// </summary>

private StringReader m\_myReader;

/// <summary>

/// Номер текущей распечатываемой страницы документа

/// </summary>

private uint m\_PrintPageNumber;

private bool m\_DocumentChanged = false;

private void richTextBox1\_TextChanged(object sender, System.EventArgs e)

{

m\_DocumentChanged = true;

}

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

{

}

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

{

if (m\_DocumentChanged)

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

this.Text = "Файл [" + saveFileDialog1.FileName + "]";

}

richTextBox1.Clear();

}

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

{

if (openFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

openFileDialog1.FileName.Length > 0)

{

try

{

richTextBox1.LoadFile(openFileDialog1.FileName,

RichTextBoxStreamType.RichText);

}

catch (System.ArgumentException ex)

{

richTextBox1.LoadFile(openFileDialog1.FileName,

RichTextBoxStreamType.PlainText);

}

this.Text = "Файл [" + openFileDialog1.FileName + "]";

}

}

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

{

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

this.Text = "Файл [" + saveFileDialog1.FileName + "]";

}

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

m\_DocumentChanged = false;

}

}

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

{

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

this.Text = "Файл [" + saveFileDialog1.FileName + "]";

}

}

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

{

pageSetupDialog1.ShowDialog();

}

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

{

MenuFilePrint();

}

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

{

MenuFilePrintPreview();

}

/// <summary>

/// Предварительный просмотр перед печатью документа

/// </summary>

private void MenuFilePrintPreview()

{

m\_PrintPageNumber = 1;

string strText = this.richTextBox1.Text;

m\_myReader = new StringReader(strText);

Margins margins = new Margins(100, 50, 50, 50);

printDocument1.DefaultPageSettings.Margins = margins;

printPreviewDialog1.ShowDialog();

m\_myReader.Close();

}

/// <summary>

/// Печать документа

/// </summary>

private void MenuFilePrint()

{

m\_PrintPageNumber = 1;

string strText = this.richTextBox1.Text;

m\_myReader = new StringReader(strText);

Margins margins = new Margins(100, 50, 50, 50);

printDocument1.DefaultPageSettings.Margins = margins;

if (printDialog1.ShowDialog() == DialogResult.OK)

{

this.printDocument1.Print();

}

m\_myReader.Close();

}

/// <summary>

/// Обработка события PrintPage

/// </summary>

private void PrintPageEventHandler(object sender,

System.Drawing.Printing.PrintPageEventArgs e)

{

int lineCount = 0; // счетчик строк

float linesPerPage = 0; // количество строк на одной странице

float yLinePosition = 0; // текущая позиция при печати по

// вертикальной оси

string currentLine = null; // текст текущей строки

// Шрифт для печати текста

Font printFont = this.richTextBox1.Font;

// Кисть для печати текста

SolidBrush printBrush = new SolidBrush(Color.Black);

// Размер отступа слева

float leftMargin = e.MarginBounds.Left;

// Размер отступа сверху

float topMargin = e.MarginBounds.Top +

3 \* printFont.GetHeight(e.Graphics);

// Вычисляем количество строк на одной странице с учетом отступа

linesPerPage = (e.MarginBounds.Height -

6 \* printFont.GetHeight(e.Graphics)) /

printFont.GetHeight(e.Graphics);

// Цикл печати всех строк страницы

while (lineCount < linesPerPage &&

((currentLine = m\_myReader.ReadLine()) != null))

{

// Вычисляем позицию очередной распечатываемой строки

yLinePosition = topMargin + (lineCount \*

printFont.GetHeight(e.Graphics));

// Печатаем очередную строку

e.Graphics.DrawString(currentLine, printFont, printBrush,

leftMargin, yLinePosition, new StringFormat());

// Переходим к следующей строке

lineCount++;

}

// Печать колонтитулов страницы

// Номер текущей страницы

string sPageNumber = "Page " + m\_PrintPageNumber.ToString();

// Вычисляем размеры прямоугольной области, занимаемой верхним

// колонтитулом страницы

SizeF stringSize = new SizeF();

stringSize = e.Graphics.MeasureString(sPageNumber, printFont,

e.MarginBounds.Right - e.MarginBounds.Left);

// Печатаем номер страницы

e.Graphics.DrawString(sPageNumber, printFont, printBrush,

e.MarginBounds.Right - stringSize.Width, e.MarginBounds.Top,

new StringFormat());

// Печатаем имя файла документа

e.Graphics.DrawString(this.Text, printFont, printBrush,

e.MarginBounds.Left, e.MarginBounds.Top, new StringFormat());

// Кисть для рисования горизонтальной линии,

// отделяющей верхний колонтитул

Pen colontitulPen = new Pen(Color.Black);

colontitulPen.Width = 2;

// Рисуем верхнюю линию

e.Graphics.DrawLine(colontitulPen,

leftMargin,

e.MarginBounds.Top + printFont.GetHeight(e.Graphics) + 3,

e.MarginBounds.Right, e.MarginBounds.Top +

printFont.GetHeight(e.Graphics) + 3);

// Рисуем линию, отделяющую нижний колонтитул документа

e.Graphics.DrawLine(colontitulPen,

leftMargin, e.MarginBounds.Bottom - 3,

e.MarginBounds.Right, e.MarginBounds.Bottom - 3);

// Печатаем текст нижнего колонтитула

e.Graphics.DrawString(

"SimpleNotepad, (c) Александр Фролов, http://www.frolov.pp.ru",

printFont, printBrush,

e.MarginBounds.Left, e.MarginBounds.Bottom, new StringFormat());

// Если напечатаны не все строки документа,

// переходим к следующей странице

if (currentLine != null)

{

e.HasMorePages = true;

m\_PrintPageNumber++;

}

// Иначе завершаем печать страницы

else

e.HasMorePages = false;

// Освобождаем ненужные более ресурсы

printBrush.Dispose();

colontitulPen.Dispose();

}

/// <summary>

/// Clean up any resources being used.

/// </summary>

protected override void Dispose(bool disposing)

{

if (m\_DocumentChanged)

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

this.Text = "Файл [" + saveFileDialog1.FileName + "]";

}

if (disposing)

{

if (components != null)

{

components.Dispose();

}

}

base.Dispose(disposing);

}

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

{

if (m\_DocumentChanged)

if (saveFileDialog1.ShowDialog() ==

System.Windows.Forms.DialogResult.OK &&

saveFileDialog1.FileName.Length > 0)

{

richTextBox1.SaveFile(saveFileDialog1.FileName);

this.Text = "Файл [" + saveFileDialog1.FileName + "]";

}

this.Close();

}

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

{

richTextBox1.Undo();

}

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

{

richTextBox1.Redo();

}

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

{

richTextBox1.Cut();

}

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

{

richTextBox1.Copy();

}

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

{

richTextBox1.Paste();

}

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

{

richTextBox1.Clear();

}

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

{

richTextBox1.SelectAll();

}

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

{

if (fontDialog1.ShowDialog() == DialogResult.OK)

{

richTextBox1.SelectionFont = fontDialog1.Font;

}

}

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

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

richTextBox1.SelectionColor = colorDialog1.Color;

}

}

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

{

SetUnderline();

}

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

{

}

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

{

SetBold();

}

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

{

SetItalic();

}

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

{

SetStrikeout();

}

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

{

richTextBox1.SelectionAlignment = HorizontalAlignment.Left;

}

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

{

richTextBox1.SelectionAlignment = HorizontalAlignment.Right;

}

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

{

richTextBox1.SelectionAlignment = HorizontalAlignment.Center;

}

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

{

Form dlgAbout = new Form2();

dlgAbout.ShowDialog();

}

private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

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

{

richTextBox1.ScrollBars = RichTextBoxScrollBars.Vertical | RichTextBoxScrollBars.Horizontal;

}

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

{

}

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

{

richTextBox1.WordWrap = true;

}

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

{

richTextBox1.WordWrap = false;

}

}

}

Форма 2

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Linq.Expressions;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace simpleNotepad3

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

{

this.Close();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.linkLabel1.LinkVisited = true;

// Navigate to a URL.

System.Diagnostics.Process.Start("totaljerkface.t.me");

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

linkLabel2.Links[linkLabel2.Links.IndexOf(e.Link)].Visited = true;

System.Diagnostics.Process.Start(" mailto:dyudyafed006@gmail.com");

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void Form2\_Load(object sender, EventArgs e)

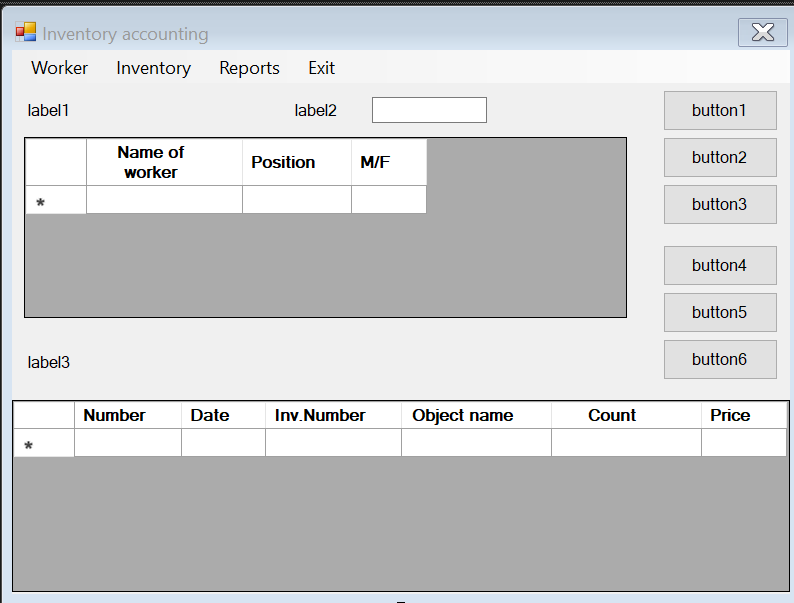
{

}

}

}

База данных



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;

namespace bd

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "accountDataSet.Worker". При необходимости она может быть перемещена или удалена.

this.workerTableAdapter.Fill(this.accountDataSet.Worker);

// TODO: данная строка кода позволяет загрузить данные в таблицу "accountDataSet.Account". При необходимости она может быть перемещена или удалена.

this.accountTableAdapter.Fill(this.accountDataSet.Account);

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

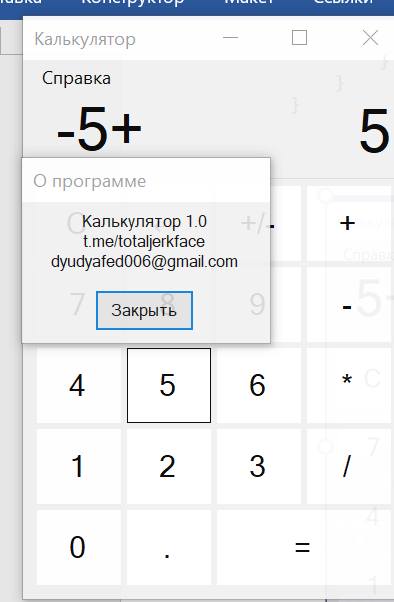
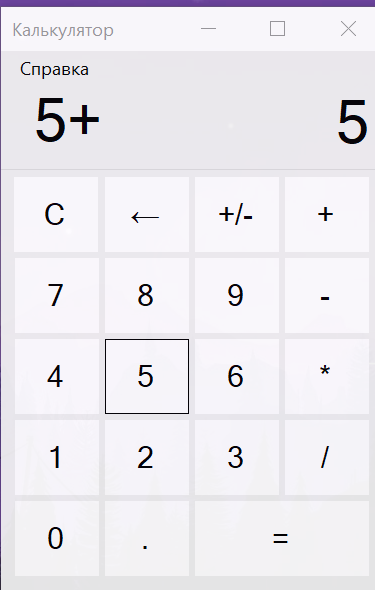
{

}

}

}

Калькулятор сам написал



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 static System.Windows.Forms.VisualStyles.VisualStyleElement.ProgressBar;

namespace calculator1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

textBox1.BackColor = BackColor;

textBox1.BorderStyle = BorderStyle.None;

label1.BackColor = BackColor;

label1.BorderStyle = BorderStyle.None;

menuStrip1.BackColor = BackColor;

}

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

{

textBox1.Text = textBox1.Text + 1;

}

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

{

textBox1.Text = textBox1.Text + 2;

}

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

{

textBox1.Text = textBox1.Text + 3;

}

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

{

textBox1.Text = textBox1.Text + 4;

}

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

{

textBox1.Text = textBox1.Text + 5;

}

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

{

textBox1.Text = textBox1.Text + 6;

}

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

{

textBox1.Text = textBox1.Text + 7;

}

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

{

textBox1.Text = textBox1.Text + 8;

}

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

{

textBox1.Text = textBox1.Text + 9;

}

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

{

textBox1.Text = textBox1.Text + 0;

}

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

{

textBox1.Text = textBox1.Text + ",";

}

float a, b;

int count;

bool znak = true;

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

{

a = float.Parse(textBox1.Text);

textBox1.Clear();

count = 2;

label1.Text = a.ToString() + "-";

znak = true;

}

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

{

a = float.Parse(textBox1.Text);

textBox1.Clear();

count = 3;

label1.Text = a.ToString() + "\*";

znak = true;

}

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

{

a = float.Parse(textBox1.Text);

textBox1.Clear();

count = 4;

label1.Text = a.ToString() + "/";

znak = true;

}

private void calculate()

{

switch (count)

{

case 1:

b = a + float.Parse(textBox1.Text);

textBox1.Text = b.ToString();

break;

case 2:

b = a - float.Parse(textBox1.Text);

textBox1.Text = b.ToString();

break;

case 3:

b = a \* float.Parse(textBox1.Text);

textBox1.Text = b.ToString();

break;

case 4:

b = a / float.Parse(textBox1.Text);

textBox1.Text = b.ToString();

break;

default:

break;

}

}

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

{

calculate();

label1.Text = "";

}

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

{

textBox1.Text = "";

label1.Text = "";

}

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

{

int lenght = textBox1.Text.Length - 1;

string text = textBox1.Text;

textBox1.Clear();

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

{

textBox1.Text = textBox1.Text + text[i];

}

}

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

{

if (znak == true)

{

textBox1.Text = "-" + textBox1.Text;

znak = false;

}

else if (znak == false)

{

textBox1.Text = textBox1.Text.Replace("-", "");

znak = true;

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void оПрограммеToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Show();

}

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

{

a = float.Parse(textBox1.Text);

textBox1.Clear();

count = 1;

label1.Text = a.ToString() + "+";

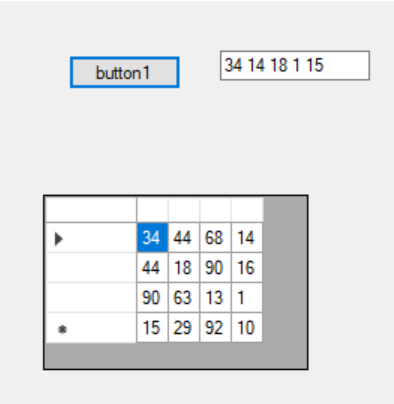
znak = true;

}

}

}

Сложный вариант 2 задание вывести числа младшие соседних



using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace kontrolnaya

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void listBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

int[,] mas = { { 34, 44, 68, 14, 100},

{ 44, 18, 90, 16, 100},

{ 90, 63, 13, 1, 100},

{ 15, 29, 92, 10, 100,},

{ 100, 100, 100, 100, 100} };

dataGridView1.RowCount = 4;

dataGridView1.ColumnCount = 4;

string res = "";

int i, j;

for (i = 0; i < 4; ++i)

for (j = 0; j < 4; ++j)

dataGridView1.Rows[i].Cells[j].Value = mas[i, j];

for (i = 0; i < 4; ++i)

{

for (j = 0; j < 4; ++j)

{

if (mas[i, j] < mas[i + 1, j] && mas[i, j] < mas[i, j + 1])

{

res += mas[i, j] + " ";

}

}

}

string result = res.Remove(res.LastIndexOf(" "));

int index = 0;

for (int m = result.Length - 4; m >= 0; m--)

{

if (result[m] == ' ')

{

index = m;

break;

}

}

result = result.Remove(index);

textBox2.Text = result;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}