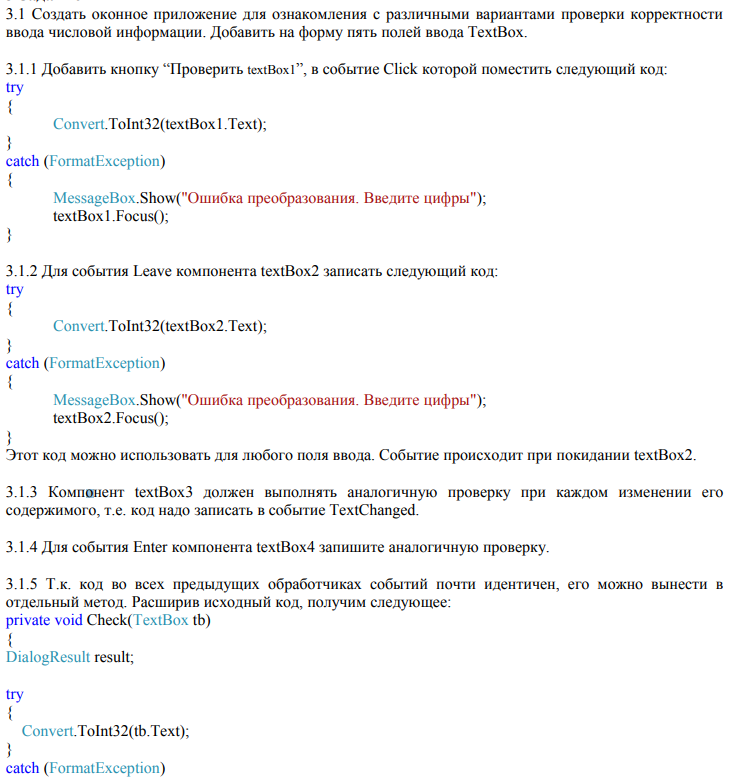
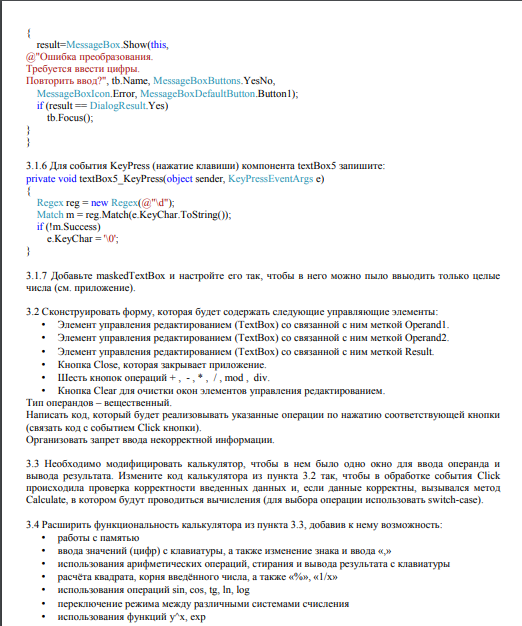
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГАКОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение «ПЕТРОВСКИЙ КОЛЛЕДЖ» (СПб ГБПОУ «Петровский колледж») | | | | |
| Отделение информационно-промышленных технологий и судостроения | | | | |
| Практическая работа №13-14 | | | | |
|  | | | | |
| **Обзор управляющих элементов и получение навыков работы с ними** | | | | |
| Выполнил: | | | |
| Семенов Сергей Андреевич | | | |
| студент(ка): | | 3902 | группы |
|  | |  |  |
| Специальность: | | 09.02.03 Программирование в компьютерных системах | |
| Преподаватель: | |  | |
|  | Спиридонова Т. С.  Бережков А. В. | | | |

|  |
| --- |
| Санкт-Петербург, 2022 |
|  |
|  |

# Задание





# Листинг

Главная форма

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsPract13\_14

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void flowLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

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

{

}

private void Check(TextBox tb)

{

DialogResult result;

try

{

Convert.ToInt32(tb.Text);

}

catch (FormatException)

{

result = MessageBox.Show(this,

@"Ошибка преобразования. Требуется ввести цифры. Повторить ввод?", tb.Name, MessageBoxButtons.YesNo, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);

if (result==DialogResult.Yes)

{

tb.Focus();

}

}

}

private void btnCheck1TextBox\_Click(object sender, EventArgs e)

{

Check(textBox1);

}

private void textBox2\_Leave(object sender, EventArgs e)

{

Check(textBox2);

}

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

{

Check(textBox3);

}

private void textBox4\_Enter(object sender, EventArgs e)

{

Check(textBox4);

}

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

{

Regex reg = new Regex(@"\d");

Match match = reg.Match(e.KeyChar.ToString());

if (!match.Success)

{

e.KeyChar = '\0';

}

}

private void maskedTextBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

Regex reg = new Regex(@"\d");

Match match = reg.Match(e.KeyChar.ToString());

if (!match.Success)

{

e.KeyChar = '\0';

}

}

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

{

Form2 form = new Form2();

Hide();

form.ShowDialog();

}

}

}

Вторая форма

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsPract13\_14

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

{

Hide();

Form1 form= new Form1();

form.ShowDialog();

}

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

{

Close();

}

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

{

}

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

{

string calcQ = textBoxCalc.Text;

void commonOperations()

{

int i = 0;

double op1 = 0, op2 = 0, answer = 0;

int dots = 0;

string action = " ";

if (calcQ.Length < 3)

{

MessageBox.Show("Неверный ввод");

}

else

{

try

{

while (Char.IsNumber(calcQ[i]) || calcQ[i] == '.')

{

i++;

}

try

{

op1 = Double.Parse(calcQ.Substring(0, i));

}

catch (FormatException)

{

MessageBox.Show("Wrong 1st operand");

}

action = calcQ[i].ToString();

i++;

//MessageBox.Show($"{i-1} action = {action}; {i} number {calcQ[i]}");

try

{

op2 = Double.Parse(calcQ.Substring(i, calcQ.Length - i));

}

catch (FormatException)

{

MessageBox.Show("Wrong 2st operand");

}

}

catch (IndexOutOfRangeException)

{

MessageBox.Show("Неверный ввод");

}

switch (action[0])

{

case '+':

answer = op1 + op2;

break;

case '-':

answer = op1 - op2;

break;

case '/':

if (op2 == 0)

{

MessageBox.Show("Операция невозвможна из-за попытки деления на ноль.");

textBoxCalc.Text = "";

}

else

{

answer = op1 / op2;

}

break;

case '\*':

answer = op1 \* op2;

break;

case '^':

answer = Math.Pow(op2, op1);

break;

case '%':

if (op2 == 0)

{

MessageBox.Show("Операция невозвможна из-за попытки деления на ноль.");

textBoxCalc.Text = "";

}

else

{

answer = op1 % op2;

}

break;

default:

answer = 0;

break;

}

textBoxCalc.Text = answer.ToString();

}

}

void specialOperations()

{

string pattern = $"^( sin | cos | tg) | ( (?<=\\() [-0-9]+ (?=\\))) | ( (?<=\\() [-0-9]+[.]\\d+ (?=\\))) |"+

"^(ln | log | exp) | ((?<=\\()[0 - 9] + (?=\\))) | ((?<=\\()[-0 - 9] +[.]\\d + (?=\\)))";

MatchCollection matches = Regex.Matches(textBoxCalc.Text, pattern, RegexOptions.IgnorePatternWhitespace);

string str="";

if (matches.Count==2)

{

string action = matches[0].Value;

double op = 0;

double answer = 0;

try

{

op = Double.Parse(matches[1].Value);

}

catch (FormatException)

{

MessageBox.Show("Ошибка при конвертации строки операнда в число");

}

switch (action)

{

case "exp":

answer = Math.Exp(Math.Exp(op));

break;

case "log":

answer = Math.Log10(Math.Abs(op));

break;

case "ln":

answer = Math.Log(Math.Abs(op));

break;

case "tg":

answer = Math.Tan(op);

break;

case "sin":

answer = Math.Sin(op);

break;

case "cos":

answer = Math.Cos(op);

break;

default:

break;

}

textBoxCalc.Text = answer.ToString();

}

}

if (Char.IsDigit(calcQ[0]))

{

commonOperations();

}

else

{

specialOperations();

}

}

}

}