**Тема 1: «Создание простых проектов, включающих методы обработки событий»**

**Вариант № 5**

**Цель работы:** Получить практические навыки создания простых проектов с использованием кнопочных переключателей, включающих методы обработки событий.

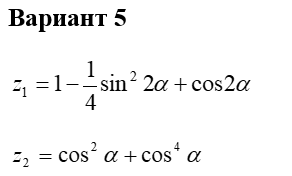
***Задание 1***

* Разработать программу, которая запрашивает исходные данные при помощи полей ввода, вычисляет значения переменных z1 и z2 и выводит ответ в поле результата. Расположить на форме компонент PictureBox и поместить в него рисунок с изображением формул, указанных в задании.

В вариантах **1-14** величины углов могут вводиться в градусах или в радианах по желанию пользователя (используйте переключатели – зависимые кнопки выбора).

***Задание 2***

* Кнопка «Вычислить» должна быть заблокирована, пока пользователь не введёт исходные данные.
* При вводе исходных данных в поле ввода должны отображаться только допустимые символы.
* При наведении мыши на поле результата оно должно подсвечиваться каким-либо цветом.



**Выполнение задания**

1. **Математическая модель**
   1. **Дано:**   
      углы x (задаются в градусах или радианах)
   2. **Найти:**   
      z1 и z2
   3. **Связь:**

Если углы задаются в радианах: z1 =z1(), z2 =z2()

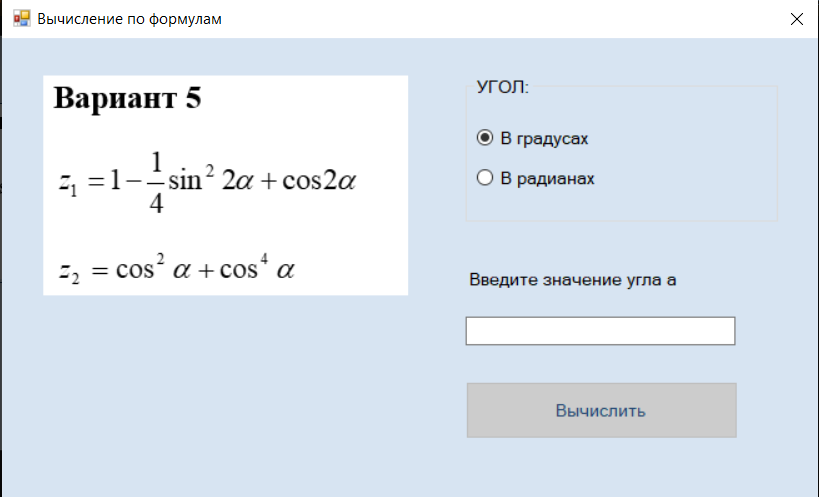
Если углы задаются в градусах: z1= z1(), z2= z2()

* 1. **ОДЗ:**дополнительных ограничений нет

1. **Проектирование пользовательского интерфейса**

Внешний вид проекта представлен на рисунке 1, значения измененных свойств компонентов представлены в таблице 1.

radioButton1



radioButton2

Form1

pictureBox1

label\_z1

button1

textBox1

label1

groupBox1

**Рис. 2.1**

1. **Свойства компонентов формы**

Значения свойств, установленные на этапе конструирования интерфейса:

*Таблица 1.*

|  |  |  |
| --- | --- | --- |
| **Название компонента** | **Свойства** | **Значения** |
| form1 | Text | «Вычисление по формулам» |
| BackColor | GradientInactiveCaption |
| Font | Гарнитура: Microsoft Sans Serif |
| Размер: 10,2 |
| Начертание: обычный |
| label1 | Text | «Введите значение угла a» |
| label\_z1 | Text | «» |
| Font | Размер: 10 |
| FontColor | MediumVioletRed |
| button1 | Text | «Вычислить» |
| Enabled | False |
| pictureBox1 | Image | team1.Properties.Resources.form |
| SizeMode | AutoSize |
| groupBox1 | Text | «УГОЛ:» |
| radioButton1 | Text | «В градусах» |
| radioButton2 | Text | «В радианах» |

1. **Сценарий использования программы**
   1. Выбор единицы измерения угла.
   2. Ввод величины угла α в поле ввода
   3. Нажатие на кнопку “Вычисление”
   4. Получение значений z1, z2
   5. Возможность вернуться к пунктам 4.1 и/или 4.2
   6. Завершение работы программы
2. **Код программы**

**Код модуля Form1.cs:**

using ClassLibrary1;

using System;

using System.Drawing;

using System.Linq;

using System.Text.RegularExpressions;

using System.Windows.Forms;

namespace team1

{

public partial class Form1 : Form

{

Class1 cl;

public Form1()

{

InitializeComponent();

button1.Enabled = false;

cl = new Class1();

}

private void label\_z1\_MouseEnter(object sender, EventArgs e)

{

label\_z1.BackColor = Color.Aqua;

}

private void label\_z1\_MouseLeave(object sender, EventArgs e)

{

label\_z1.BackColor = SystemColors.GradientInactiveCaption;

}

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

{

double val = double.Parse(textBox1.Text);

double z1 = cl.res\_z1(val);

double z2 = cl.res\_z2(val);

label\_z1.Text="Ответ: z1 = "+z1+ " z2 = " + z2;

}

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

{

if (textBox1.Text.Length > 0)

{

button1.Enabled = true;

}

else

{

button1.Enabled = false;

}

if (textBox1.Text.StartsWith(","))

{

button1.Enabled = false;

}

}

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

{

cl.grade = true;

cl.rade = false;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

bool hasComm = (sender as TextBox).Text.Contains(',');

bool hasMin = (sender as TextBox).Text.Contains('-');

if ((e.KeyChar == ',' && hasComm) || (e.KeyChar == '-' && hasMin))

{

e.Handled = true;

}

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != ',') && (e.KeyChar != '-'))

{

e.Handled = true;

}

if ((e.KeyChar == ',' && hasComm) || (e.KeyChar == '-' && hasMin))

{

e.Handled = true;

}

if (e.KeyChar == '-' && hasComm)

{

e.Handled = true;

}

if (e.KeyChar == ',' && hasMin && textBox1.Text.Length > 2)

{

e.Handled = true;

}

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

cl.grade =false;

cl.rade = true;

}

}

}

**Код модуля ClassLibrary1.dll:**

using System;

namespace ClassLibrary1

{

public class Class1

{

public Class1() { }

public int a;

public bool grade;

public bool rade;

public double pars(double val)

{

if (grade)

{

return val\*(Math.PI/180.0);

}

else if(rade)

{

return val;

}

grade = false;

rade = false;

return 0;

}

public double res\_z1(double z1)

{

double val = pars(z1);

double z = 1-((Math.Pow(Math.Sin(2\*val),2))/4)+Math.Cos(2\*val);

return z;

}

public double res\_z2(double z2)

{

double val = pars(z2);

double co1 = Math.Pow(Math.Cos(val),2);

double z = co1+Math.Pow(co1,2);

return z;

}

}

}

1. **Схемы основных подпрограмм**

(e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf(',') > -1)

!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != ',')

e.Handled = true

да

нет

да

нет

Начало

Конец

e.Handled = false

e.Handled = false

**Рис 6.1.** **Обработка ввода textBox\_KeyPress**

radioButton1.Checked?

double val = double.Parse(textBox1.Text);

да

нет

X= Math.PI\*val / 180

Вывод(z1,z2)

Начало

Конец

z1=z1(x)

z2=z2(x)

radioButton2.Checked?

да

X = val

нет

**Рис 6.2. Метод button1\_Click**

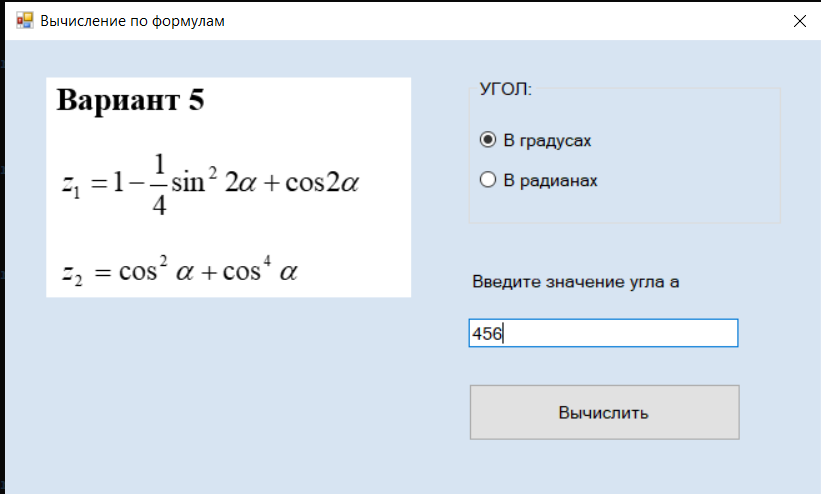
1. **Протокол испытаний:**

*Таблица 2.1.* Протокол испытаний

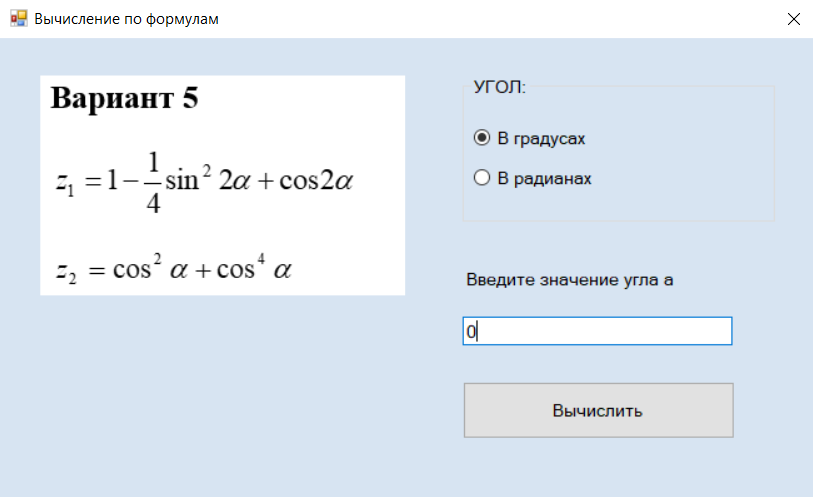
|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Сообщения программы и вводимые значения** | **Ожидаемые результаты** | **Фактические результаты** |
| Способность ввода данных | x = 456 | Числа вводятся в текстовое поле | Такие же, как ожидаемые результаты  (См. Рис. 7.1) |
| Способность контроля вводимых данных | x = M0F | x = 0 | Такие же, как ожидаемые результаты  (См. Рис. 7.2) |
| Способность программы обеспечить контроль вводимых данных. | x = -8- | x = -8 | Соответствуют ожиданиям.  (См. Рис. 7.3) |
| Способность контроля непереводимых полей ввода | Оставить поле ввода пустым или с знаком ‘,’ но без чисел | Кнопка для решения заблокирована | Такие же, как ожидаемые результаты  (Рис.7.4) |
| Проверить способность программы вычислять значения z1 и z2 с исходными данными, заданными в радианах | Ввод переменной, нажатие на переключатель «в радианах» и нажатие на кнопку “Вычислить” Входные данные 1:  x = 30 Входные данные 2:  x = 60 | Результат 1 z1 = 0,024 z2 = 0,024  Результат 2: z1 = 1,73 z2 = 1,73 | Результаты соответствуют контрольным подсчётам.  (см. Рис. 7.5, 7.6) |

|  |  |  |  |
| --- | --- | --- | --- |
| Проверить способность программы вычислять значения z1 и z2 с исходными данными, заданными в градусах | Ввод переменной, нажатие на переключатель «в градусах» и нажатие на кнопку “Вычислить”  Входные данные 1:  x = 1718,87  Входные данные 2:  x = 3437,75 | Результат 1: z1 = 0,024 z2 = 0,024  Результат 2: z1 = 1,73 z2 = 1,73 | Результаты соответствуют контрольным подсчётам.  (см. Рис. 7.7, 7.8) |
| Работоспособность подсветки результата при наведении курсора | Наведение курсора мыши на поле результата и позже снятие курсора с поля | Поле изменит фоновый цвет пока курсор на нём и изменит его обратно при отведении курсора | Такие же, как ожидаемые результаты  (Рис. 7.9) |

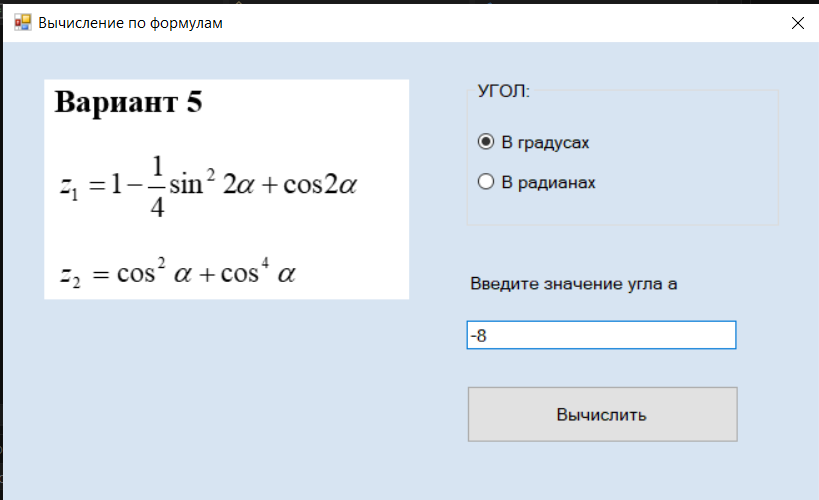
1. **Скриншоты на этапе тестирования проекта:**



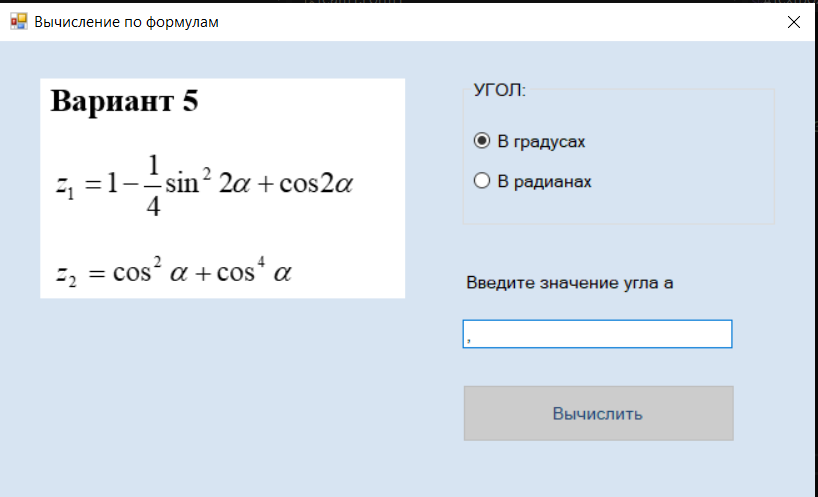
**Рис. 7.1**



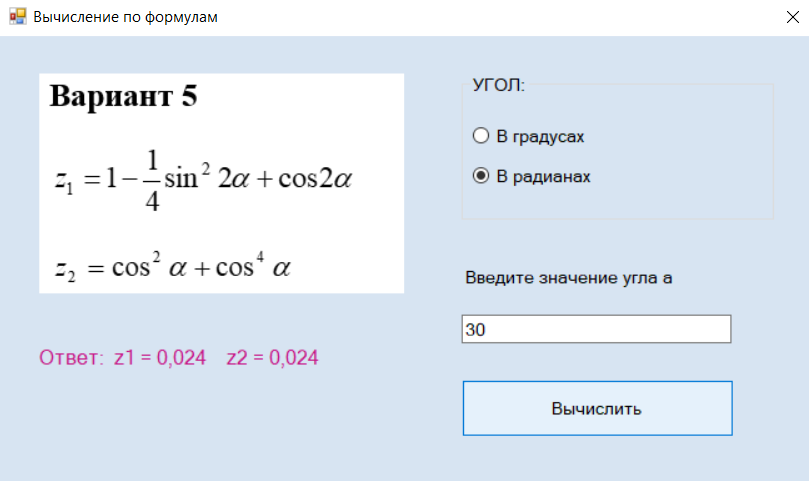
**Рис. 7.2**



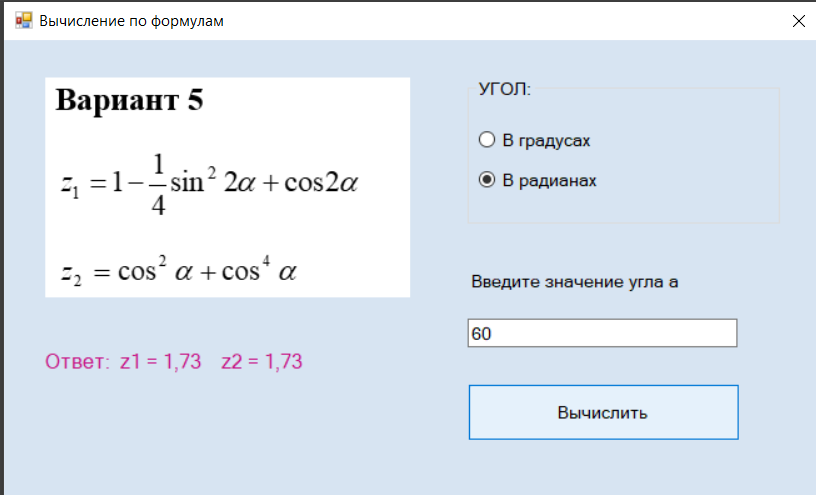
**Рис. 7.3**



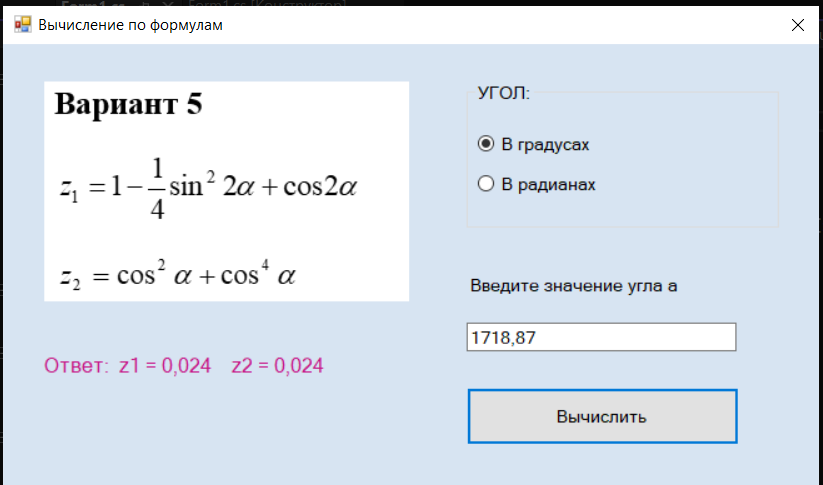
**Рис. 7.4**



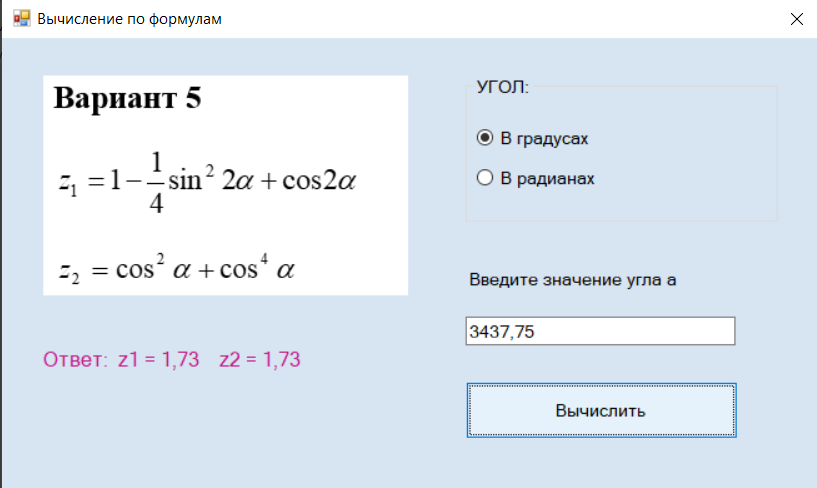
**Рис. 7.5**



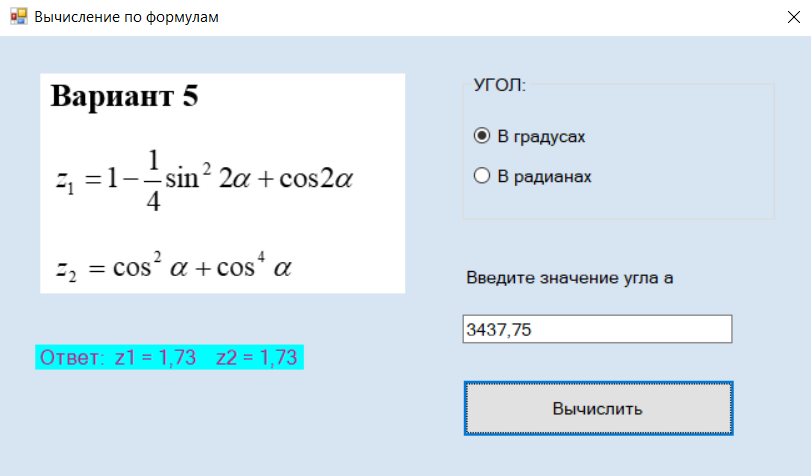
**Рис. 7.6**



**Рис. 7.7**

****

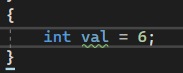
**Рис. 7.8**



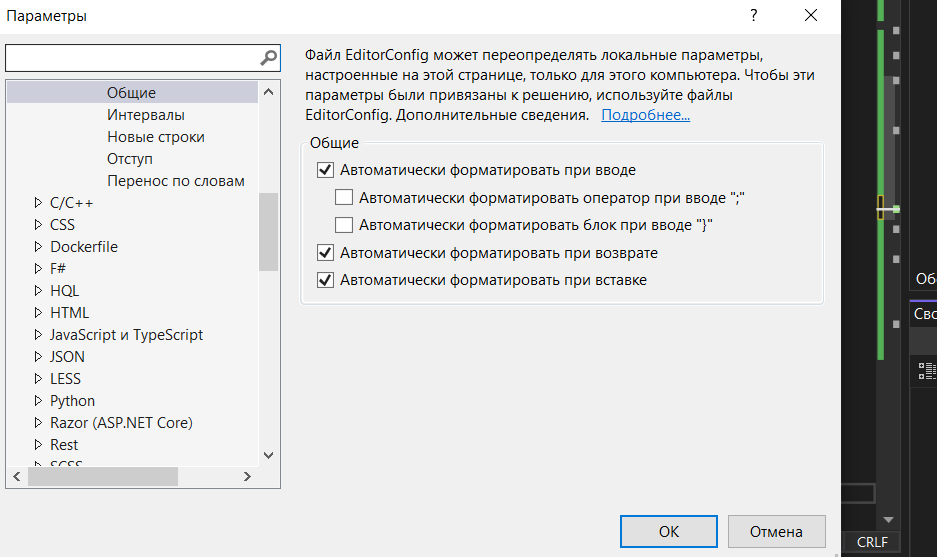
**Рис. 7.9**

**Задание 3**

Решение задания 3 представлено на рисунках 3.1.1-3.4.4

****

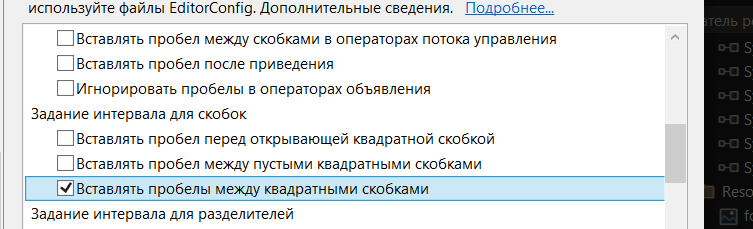
**Рис. 3.1.1. Автоматическое форматирование блока**

****

**Рис. 3.1.2. Сброс автоматического форматирования**

****

**Рис. 3.1.3. Ввод кода без автоматического форматирования**



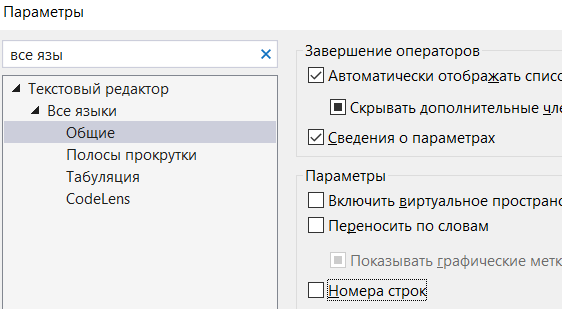
**Рис. 3.1.4. Включение флага во вкладке «интервалы»**



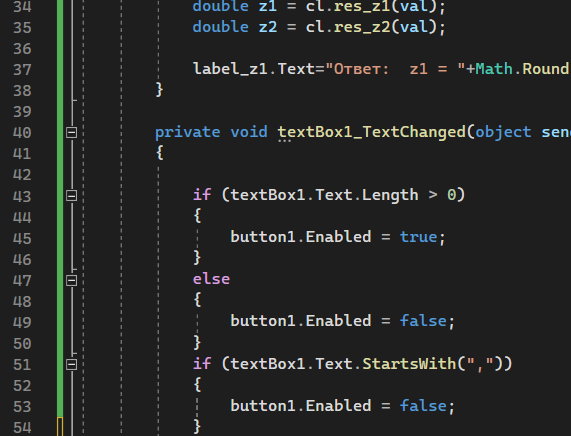
**Рис. 3.1.5. Результат до включения флага**



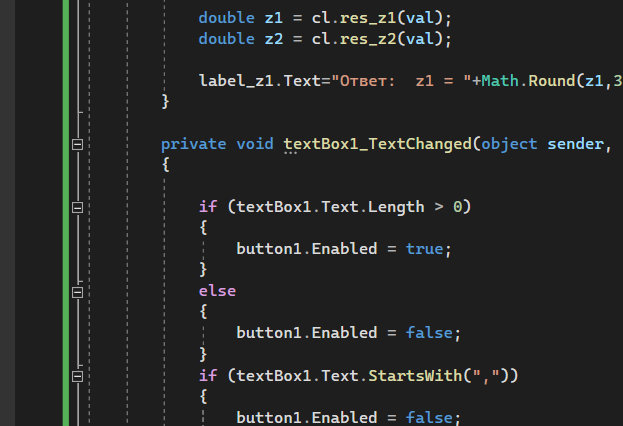
**Рис. 3.1.6. Результат после включения флага**

****

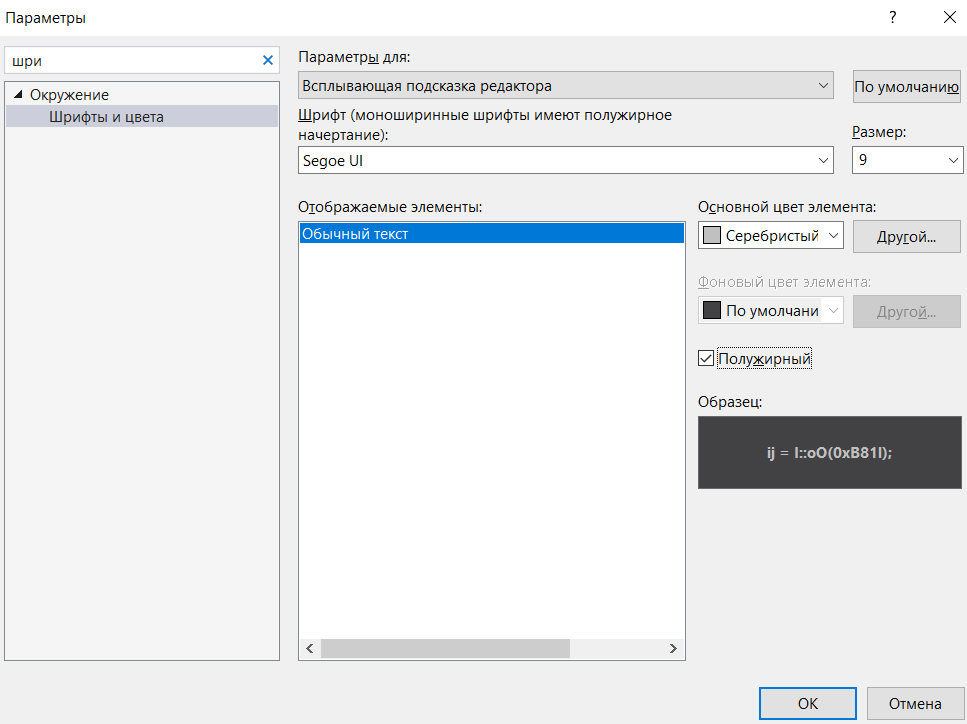
**Рис. 3.1.7. Выключение нумерации строк**



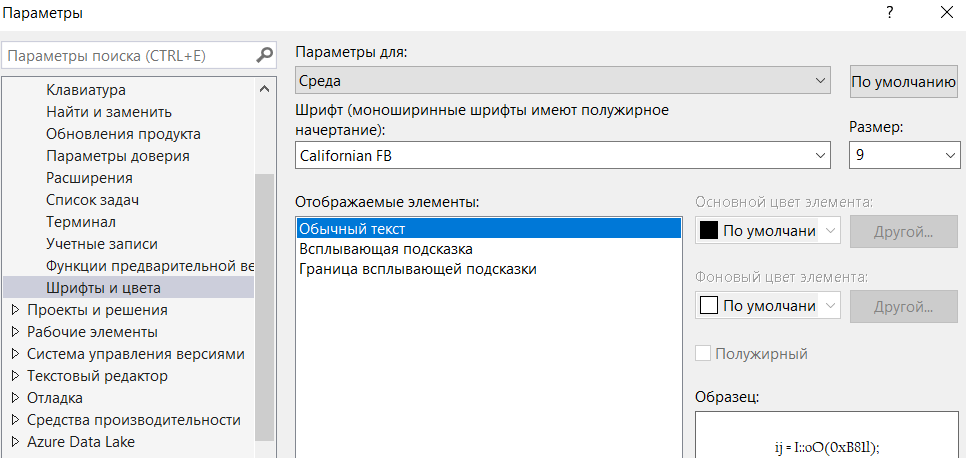
**Рис. 3.1.8. Результат до**



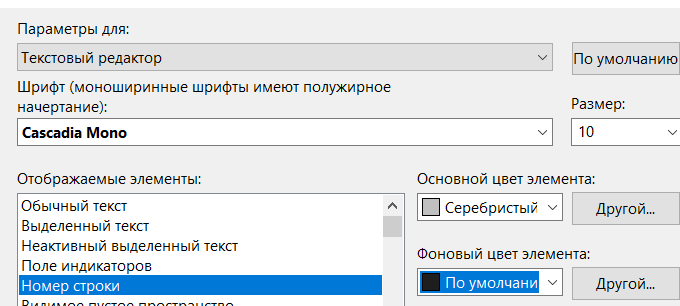
**Рис. 3.1.9. Результат после**



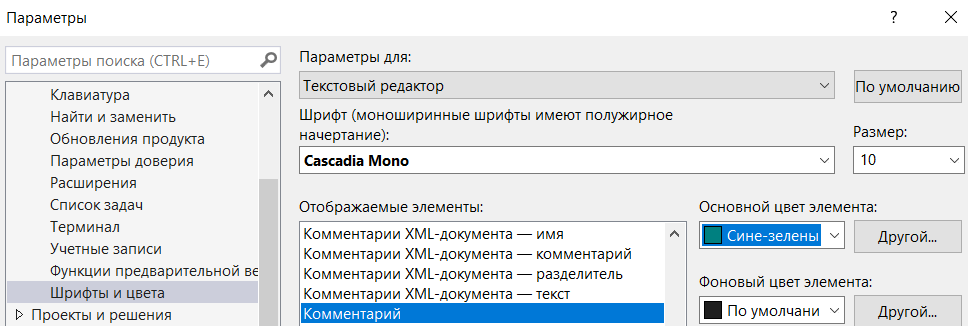
**Рис. 3.2.1. Изменение параметров всплывающей подсказки**



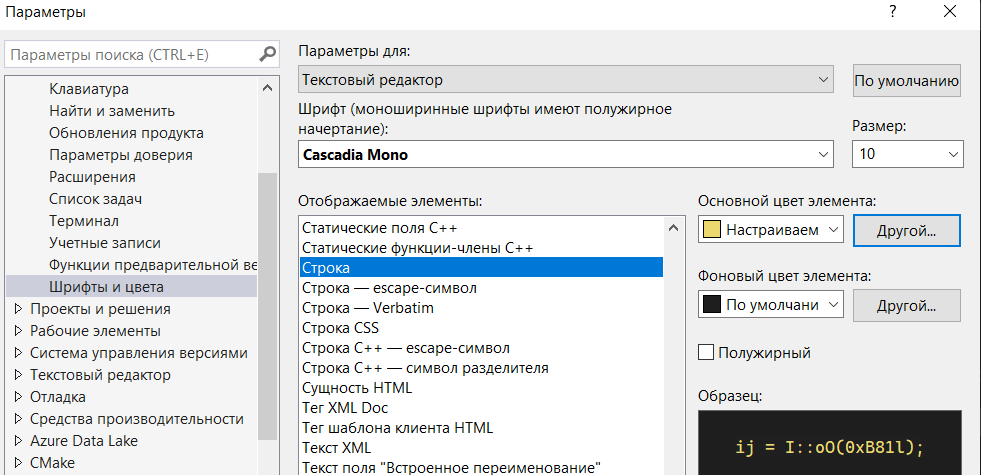
**Рис. 3.2.2. Изменение параметров текста среды**



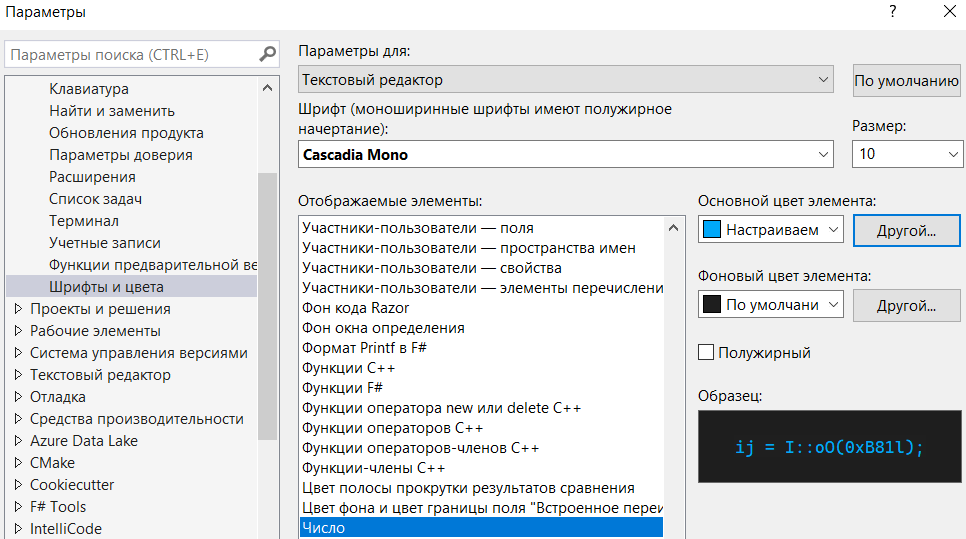
**Рис. 3.2.3. Изменение параметров номеров строки**



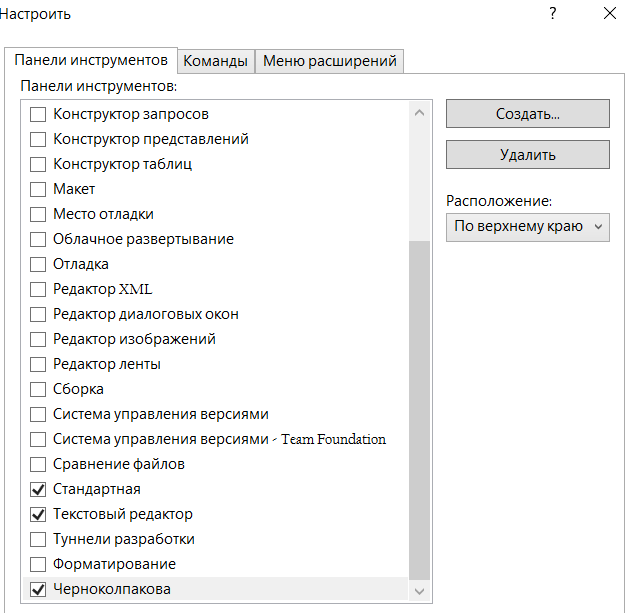
**Рис. 3.2.4. Изменение параметров комментариев строки**



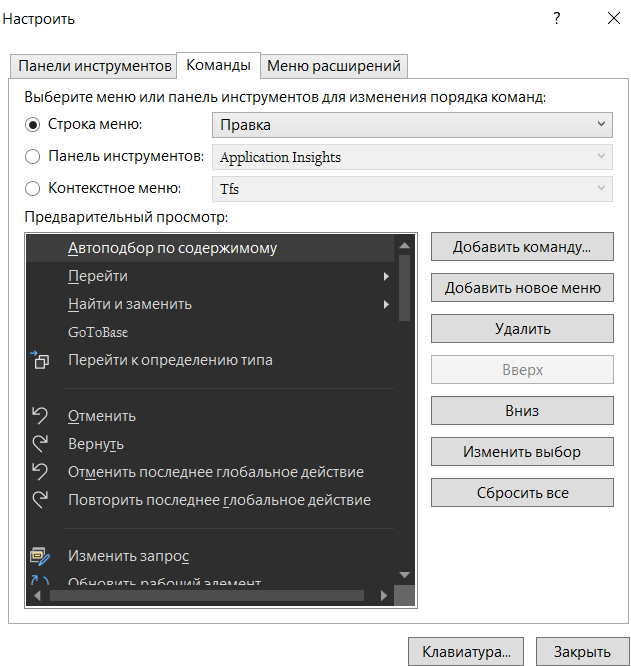
**Рис. 3.2.6. Изменение параметров типа «строка»**



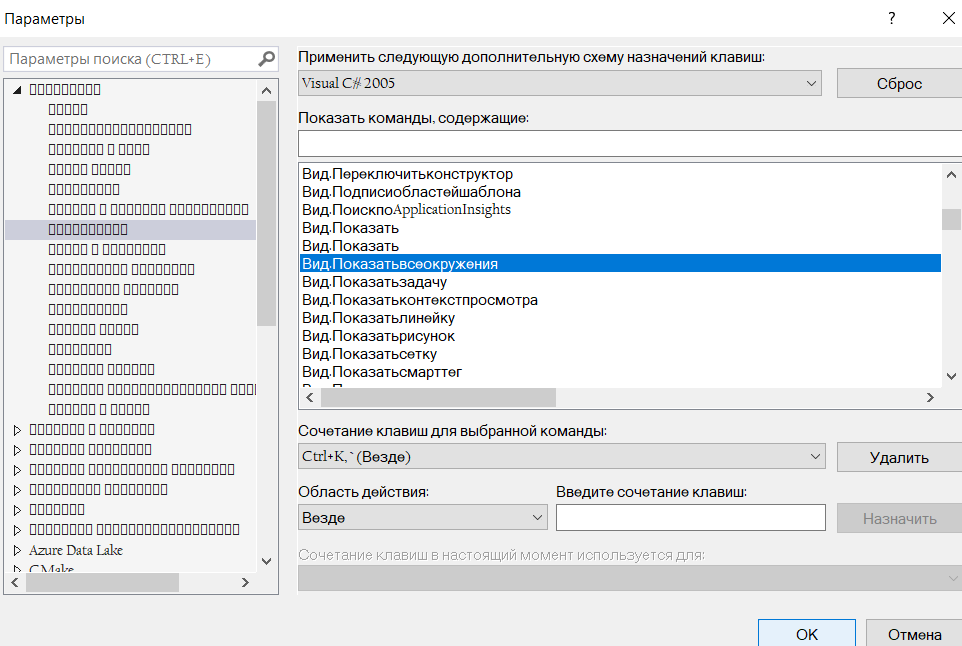
**Рис. 3.2.7. Изменение параметров типа «Число»**



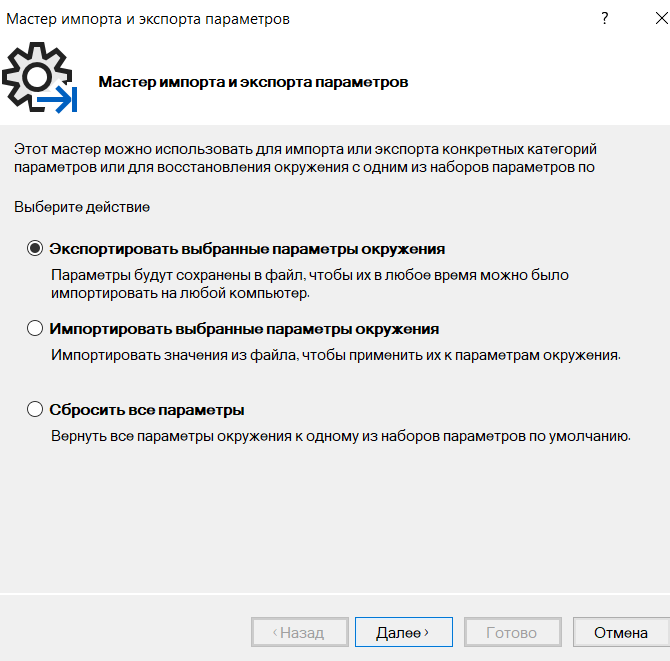
**Рис. 3.3.1. Создание своей панели инструментов**



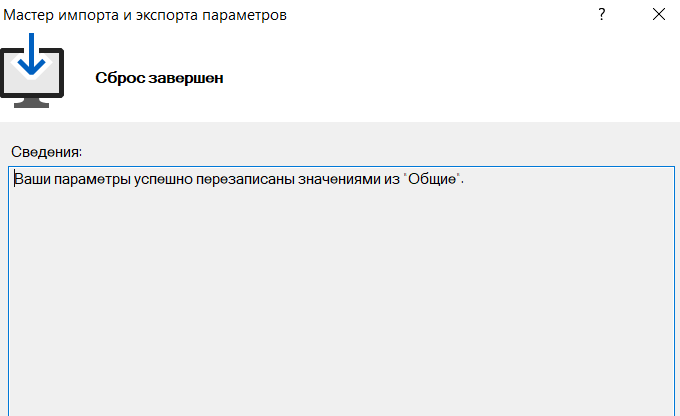
**Рис. 3.3.2. Добавление команд на свою панель инструментов**



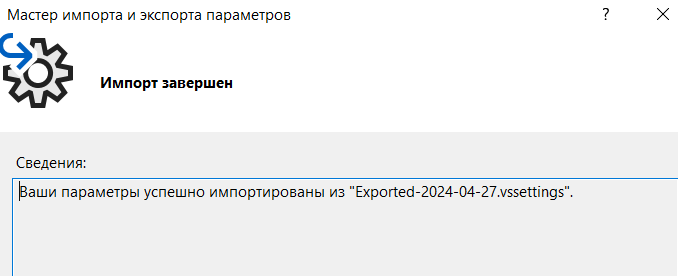
**Рис. 3.3.3. Добавление горячих клавиш на свою панель инструментов**



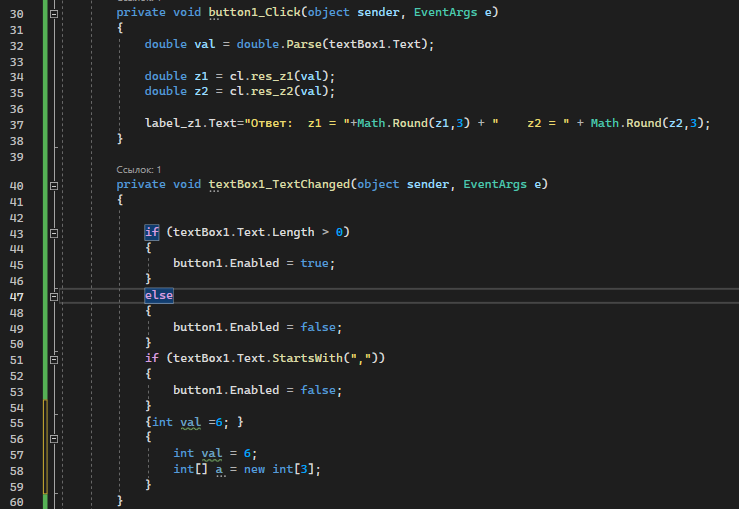
**Рис. 3.4.1. Экспорт настроек параметров среды**



**Рис. 3.4.2. Сброс текущих настроек параметров среды**



**Рис. 3.4.3 Импорт своих настроек параметров среды**



**Рис. 3.4.4. Результат установки своих настроек параметров среды**