

## 1 Приложение для вычисления простой функции

ЗАДАНИЕ. Реализовать приложение для вычисления функции

$$\frac{\sin^2 x + y^2}{(y - 4)\sqrt{x^2 - 2x + 1}}$$

Создано окно приложения, содержащее три элемента `TextBox`, три элемента `Label` и один элемент `Button`. Для отображения сообщений об ошибках в окно добавлен элемент `ErrorProvider`. Вид окна представлен на рисунке 1.

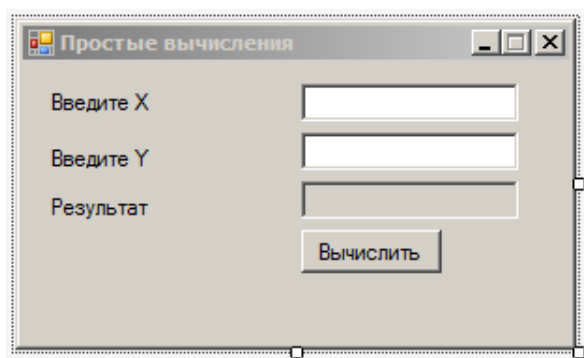


Рисунок 1 – Окно приложения «Простые вычисления» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 1.

Таблица 1 – Значения атрибутов элементов в приложении «Простые вычисления»

Наименование атрибута	Значение
Для формы	
Text	Простые вычисления
MaximizeBox	False
Для первой надписи	
(Name)	lblInputX
Text	Введите целое число
Для второй надписи	
(Name)	lblInputY
Для третьей надписи	
(Name)	lblOutput
Text	Результат
Для первого текстового поля	
(Name)	txtInputX
Для второго текстового поля	
(Name)	txtInputY
Для третьего текстового поля	
(Name)	txtOutput
(ReadOnly)	True
Для кнопки	
(Name)	btnCalculate
Text	Вычислить
Для обработчика ошибок	
(Name)	errorProvider

Была написана функция вычисления выражения:

```

1 long double count(int X, int Y) {
2     //Значение функции
3     return (sin(X)*sin(X) + Y*Y) / ((Y - 4)*abs(X - 1));
4 }

```

На нажатие кнопки «Вычислить» установлено выполнение следующего кода:

```

1     int InputNumberX;
2     int InputNumberY;
3     bool resultX = Int32::TryParse(this->txtInputX->Text, InputNumberX);
4     bool resultY = Int32::TryParse(this->txtInputY->Text, InputNumberY);
5
6     errorProvider->SetError(this->txtInputX, String::Empty);

```

```

7      errorProvider->SetError(this->txtInputY, String::Empty);
8      this->txtOutput->Text = "";
9
10     // Если x не число
11     if (!resultX) {
12         errorProvider->SetError(this->txtInputX, "Введено_не_целое_
            число");
13     }
14     // Если y не число
15     if (!resultY) {
16         errorProvider->SetError(this->txtInputY, "Введено_не_целое_
            число");
17     }
18
19     //если нет ошибок
20     if (resultX && resultY) {
21         // проверка деления на 0
22         if (InputNumberY == 4) {
23             errorProvider->SetError(this->txtInputY, "Деление_на_
                0._Y_не_должен_равняться_4");
24         }
25         if (InputNumberX == 1) {
26             errorProvider->SetError(this->txtInputX, "Деление_на_
                0._X_не_должен_равняться_1");
27         }
28
29         //если нет ошибок
30         if (InputNumberY != 4 && InputNumberX != 1) {
31             double OutputNumber = count(InputNumberX,
                InputNumberY);
32             this->txtOutput->Text = System::Convert::ToString(
                OutputNumber);
33         }
34     }

```

После запуска приложения на экране появляется окно (см. рисунок 2).

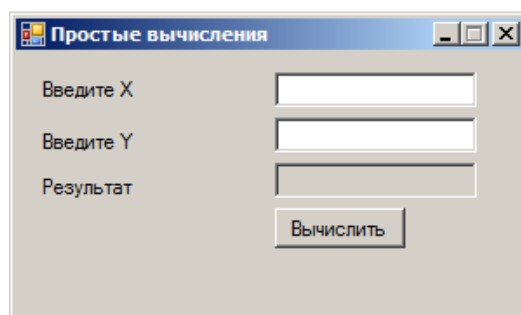


Рисунок 2 – Окно приложения «Простые вычисления»: начальный запуск

При вводе целых чисел после нажатия кнопки в поле вывода приводится результат вычисления функции для заданных чисел (см. рисунок 3).

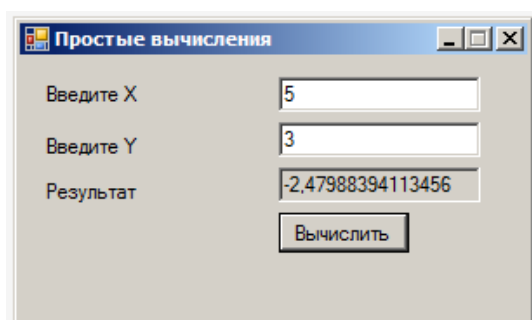


Рисунок 3 – Окно приложения «Простые вычисления»: корректное вычисление

При вводе значений, не попадающих в ОДЗ, возникает сообщение об ошибке (см. рисунок 4).

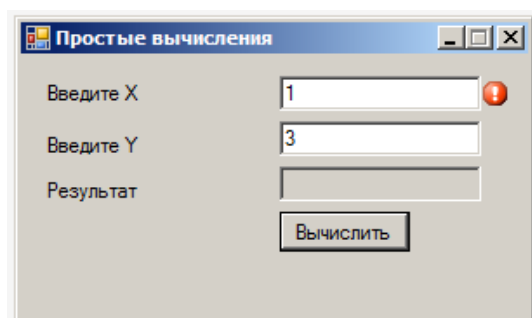


Рисунок 4 – Окно приложения «Простые вычисления»: сообщение о вводе числа, не попадающего в ОДЗ

При вводе некорректных значений возникает сообщение об ошибке (см. рисунок 5).

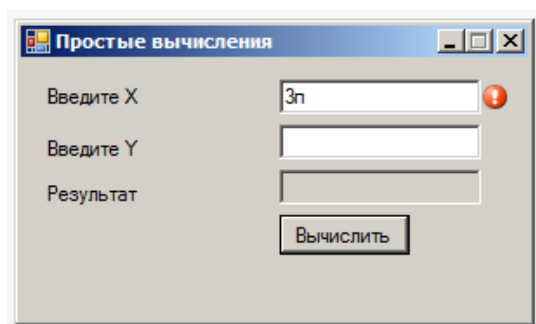


Рисунок 5 – Окно приложения «Простые вычисления»: сообщение о некорректном вводе

Полный код программы приведен в приложении 2.

## 2 Листинг файлов проекта

### 2.1 Листинг файла simpleCount.cpp

```
1 #include "count.h"
2 #include "MyForm.h"
3
4 using namespace factorial;
5
6 [STAThreadAttribute]
7 int main(array<System::String ^> ^args) {
8
9     Application::EnableVisualStyles();
10    Application::SetCompatibleTextRenderingDefault(false);
11
12    Application::Run(gcnew MyForm1());
13    return 0;
14 }
```

### 2.2 Листинг файла count.cpp

```
1 #pragma once
2 #include <cmath>
3 long double count(int X, int Y) {
4     //Значение функции
5     return (sin(X)*sin(X) + Y*Y) / ((Y - 4)*abs(X - 1));
6 }
```

### 2.3 Листинг файла MyForm.h

```
1 #pragma once
2 #include "count.h"
3
4 namespace factorial {
5
6     using namespace System;
7     using namespace System::ComponentModel;
8     using namespace System::Collections;
9     using namespace System::Windows::Forms;
10    using namespace System::Data;
11    using namespace System::Drawing;
12
13    /// <summary>
14    /// Сводка для MyForm1
15    /// </summary>
```

```

16 public ref class MyForm1 : public System::Windows::Forms::Form
17 {
18 public:
19     MyForm1(void)
20     {
21         InitializeComponent();
22         //
23         //TODO: добавьте код конструктора
24         //
25     }
26
27 protected:
28     /// <summary>
29     /// Освободить все используемые ресурсы.
30     /// </summary>
31     ~MyForm1()
32     {
33         if (components)
34         {
35             delete components;
36         }
37     }
38 private: System::Windows::Forms::TextBox^ txtInputX;
39 protected:
40
41 private: System::Windows::Forms::TextBox^ txtOutput;
42 private: System::Windows::Forms::Label^ lblInputX;
43
44 protected:
45
46 protected:
47
48
49 private: System::Windows::Forms::Label^ lblOutput;
50 private: System::Windows::Forms::Button^ btnCalculate;
51 private: System::Windows::Forms::ErrorProvider^ errorProvider;
52 private: System::Windows::Forms::Label^ lblInputY;
53 private: System::Windows::Forms::TextBox^ txtInputY;
54
55
56 private: System::ComponentModel::IContainer^ components;

```

```

57
58
59
60     private:
61         ///  
62         ///  
63         ///  
64  
65  
66     #pragma region Windows Form Designer generated code  
67         ///  
68         ///  
69         ///  
70         ///  
71     void InitializeComponent(void)  
72     {  
73         this->components = (gcnew System::ComponentModel::Container  
74             ());  
75         this->txtInputX = (gcnew System::Windows::Forms::TextBox());  
76         this->txtOutput = (gcnew System::Windows::Forms::TextBox());  
77         this->lblInputX = (gcnew System::Windows::Forms::Label());  
78         this->lblOutput = (gcnew System::Windows::Forms::Label());  
79         this->btnCalculate = (gcnew System::Windows::Forms::Button());  
80         this->errorProvider = (gcnew System::Windows::Forms::  
81             ErrorProvider(this->components));  
82         this->lblInputY = (gcnew System::Windows::Forms::Label());  
83         this->txtInputY = (gcnew System::Windows::Forms::TextBox());  
84         (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->errorProvider))->BeginInit();  
85         this->SuspendLayout();  
86         //  
87         // txtInputX  
88         //  
89         this->txtInputX->Location = System::Drawing::Point(149, 12);  
90         this->txtInputX->Name = L"txtInputX";  
91         this->txtInputX->Size = System::Drawing::Size(116, 20);  
92         this->txtInputX->TabIndex = 0;  
93         //  
94         // txtOutput  
95         //  
96         this->txtOutput->Location = System::Drawing::Point(149, 64);

```



```

95      this->txtOutput->Name = L"txtOutput";
96      this->txtOutput->ReadOnly = true;
97      this->txtOutput->Size = System::Drawing::Size(116, 20);
98      this->txtOutput->TabIndex = 1;
99      //
100     // lblInputX
101     //
102     this->lblInputX->AutoSize = true;
103     this->lblInputX->Location = System::Drawing::Point(12, 15);
104     this->lblInputX->Name = L"lblInputX";
105     this->lblInputX->Size = System::Drawing::Size(59, 13);
106     this->lblInputX->TabIndex = 2;
107     this->lblInputX->Text = L"Введите_X";
108     this->lblInputX->Click += gcnew System::EventHandler(this, &
        MyForm1::label1_Click);
109     //
110     // lblOutput
111     //
112     this->lblOutput->AutoSize = true;
113     this->lblOutput->Location = System::Drawing::Point(12, 71);
114     this->lblOutput->Name = L"lblOutput";
115     this->lblOutput->Size = System::Drawing::Size(59, 13);
116     this->lblOutput->TabIndex = 3;
117     this->lblOutput->Text = L"Результат";
118     //
119     // btnCalculate
120     //
121     this->btnCalculate->Location = System::Drawing::Point(149, 90);
122     this->btnCalculate->Name = L"btnCalculate";
123     this->btnCalculate->Size = System::Drawing::Size(75, 23);
124     this->btnCalculate->TabIndex = 4;
125     this->btnCalculate->Text = L"Вычислить";
126     this->btnCalculate->UseVisualStyleBackColor = true;
127     this->btnCalculate->Click += gcnew System::EventHandler(this,
        &MyForm1::btnCalculate_Click);
128     //
129     // errorProvider
130     //
131     this->errorProvider->ContainerControl = this;
132     //
133     // lblInputY

```

```

134 //
135 this->lblInputY->AutoSize = true;
136 this->lblInputY->Location = System::Drawing::Point(12, 45);
137 this->lblInputY->Name = L"lblInputY";
138 this->lblInputY->Size = System::Drawing::Size(59, 13);
139 this->lblInputY->TabIndex = 6;
140 this->lblInputY->Text = L"Введите_Y";
141 //
142 // txtInputY
143 //
144 this->txtInputY->Location = System::Drawing::Point(149, 38);
145 this->txtInputY->Name = L"txtInputY";
146 this->txtInputY->Size = System::Drawing::Size(116, 20);
147 this->txtInputY->TabIndex = 5;
148 //
149 // MyForm1
150 //
151 this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
152 this->AutoScaleMode = System::Windows::Forms::AutoScaleMode
        ::Font;
153 this->ClientSize = System::Drawing::Size(292, 150);
154 this->Controls->Add(this->lblInputY);
155 this->Controls->Add(this->txtInputY);
156 this->Controls->Add(this->btnCalculate);
157 this->Controls->Add(this->lblOutput);
158 this->Controls->Add(this->lblInputX);
159 this->Controls->Add(this->txtOutput);
160 this->Controls->Add(this->txtInputX);
161 this->MaximizeBox = false;
162 this->Name = L"MyForm1";
163 this->Text = L"Простые_вычисления";
164 ( cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(
        this->errorProvider))->EndInit();
165 this->ResumeLayout(false);
166 this->PerformLayout();
167
168     }
169 #pragma endregion
170     private: System::Void label1_Click(System::Object^ sender, System::EventArgs^ e)
171     {

```

```

172 private: System::Void btnCalculate_Click(System::Object^ sender, System::
    EventArgs^ e) {
173     int InputNumberX;
174     int InputNumberY;
175     bool resultX = Int32::TryParse(this->txtInputX->Text, InputNumberX);
176     bool resultY = Int32::TryParse(this->txtInputY->Text, InputNumberY);
177
178     errorProvider->SetError(this->txtInputX, String::Empty);
179     errorProvider->SetError(this->txtInputY, String::Empty);
180     this->txtOutput->Text = "";
181
182     // Если x не число
183     if (!resultX) {
184         errorProvider->SetError(this->txtInputX, "Введено_не_целое_
            число");
185     }
186     // Если y не число
187     if (!resultY) {
188         errorProvider->SetError(this->txtInputY, "Введено_не_целое_
            число");
189     }
190
191     //если нет ошибок
192     if (resultX && resultY) {
193         // проверка деления на 0
194         if (InputNumberY == 4) {
195             errorProvider->SetError(this->txtInputY, "Деление_на_
                0._Y_не_должен_равняться_4");
196         }
197         if (InputNumberX == 1) {
198             errorProvider->SetError(this->txtInputX, "Деление_на_
                0._X_не_должен_равняться_1");
199         }
200
201         //если нет ошибок
202         if (InputNumberY != 4 && InputNumberX != 1) {
203             double OutputNumber = count(InputNumberX,
                InputNumberY);
204             this->txtOutput->Text = System::Convert::ToString(
                OutputNumber);
205         }

```

206 }

207 }

208 };

209 }