

1 Приложение для вычисления факториала

ЗАДАНИЕ. Реализовать приложение для вычисления факториала.

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

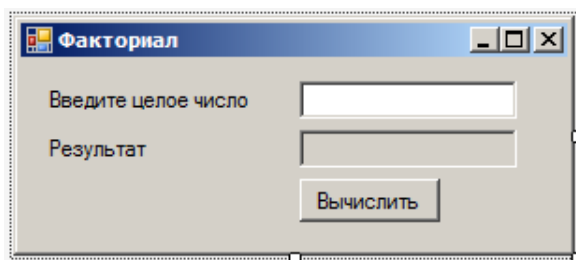


Рисунок 1 – Окно приложения «Факториал» открытое в конструкторе

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

Таблица 1 – Значения атрибутов элементов в приложении «Факториал»

Наименование атрибута	Значение
Для формы	
<code>Text</code>	Факториал
Для первой надписи	
<code>(Name)</code>	<code>lblInput</code>
<code>Text</code>	Введите целое число
Для второй надписи	
<code>(Name)</code>	<code>lblOutput</code>
<code>Text</code>	Результат
Для первого текстового поля	
<code>(Name)</code>	<code>txtInput</code>
Для второго текстового поля	
<code>(Name)</code>	<code>txtOutput</code>
<code>ReadOnly</code>	<code>True</code>
Для кнопки	
<code>(Name)</code>	<code>btnCalculate</code>
<code>Text</code>	Вычислить
Для обработчика ошибок	
<code>(Name)</code>	<code>errorProvider</code>

Была написана функция вычисления факториала:

```
1 long double fact(int N) {  
2     // если отрицательное число  
3     if (N < 0)
```

```

4         return 0;
5     if (N == 0)
6         return 1;
7     else
8         return N * fact(N - 1);
9 }

```

На нажатие кнопки «Вычислить» установлено выполнение следующего кода:

```

1     int InputNumber;
2     bool result = Int32::TryParse(this->txtInput->Text, InputNumber);
3     if (result) {
4         errorProvider->SetError(this->txtInput, String::Empty);
5         if (InputNumber < 0) {
6             this->txtOutput->Text = "";
7             errorProvider->SetError(this->txtInput, "Введено_
            отрицательное_число");
8         }
9         else if (InputNumber > 17) {
10            this->txtOutput->Text = "";
11            errorProvider->SetError(this->txtInput, "Выход_за_
            границы_типа");
12        }
13        else {
14            double OutputNumber = fact(InputNumber);
15            this->txtOutput->Text = System::Convert::ToString(
                OutputNumber);
16        }
17    }
18    else {
19        this->txtOutput->Text = "";
20        errorProvider->SetError(this->txtInput, "Введено_не_целое_
            число");
21    }

```

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

При вводе целого числа после нажатия кнопки в поле вывода приводится результат вычисления факториала для заданного числа (см. рисунок 3).

При вводе некорректного значения возникает сообщение об ошибке (см. рисунок 4).

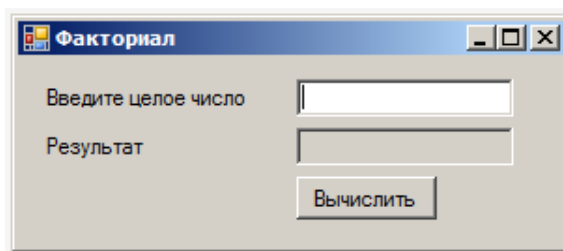


Рисунок 2 – Окно приложения «Факториал»: начальный запуск

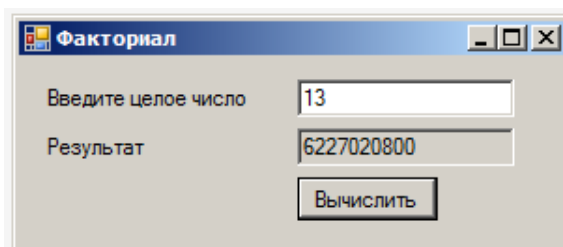


Рисунок 3 – Окно приложения «Факториал»: корректное вычисление

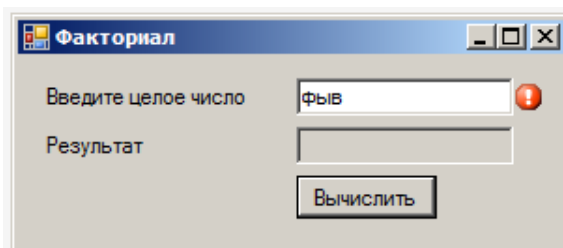


Рисунок 4 – Окно приложения «Факториал»: сообщение о некорректном вводе

При вводе слишком большого положительного числа возникает сообщение об ошибке (см. рисунок 5).

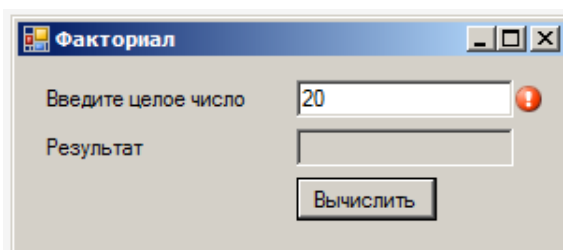


Рисунок 5 – Окно приложения «Факториал»: ввод слишком большого положительного числа

При вводе отрицательного числа возникает сообщение об ошибке (см. рисунок 6).

Полный код программы приведен в приложении А.

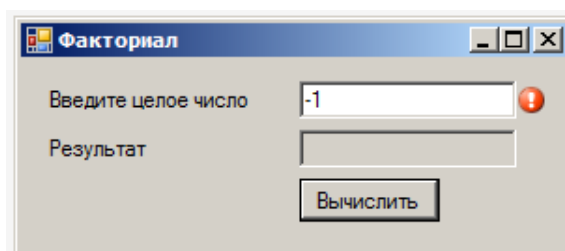


Рисунок 6 – Окно приложения «Факториал»: ввод отрицательного числа

ПРИЛОЖЕНИЕ А

Листинг файлов проекта

1.1 Листинг файла factorial.cpp

```
1  #include "fact.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 }
```

1.2 Листинг файла fact.cpp

```
1  #pragma once
2  long double fact(int N) {
3      // если отрицательное число
4      if (N < 0)
5          return 0;
6      if (N == 0)
7          return 1;
8      else
9          return N * fact(N - 1);
10 }
```

1.3 Листинг файла MyForm.h

```
1  #pragma once
2  #include "fact.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^ txtInput;
39     private: System::Windows::Forms::TextBox^ txtOutput;
40     private: System::Windows::Forms::Label^ lblInput;
41     protected:
42
43     protected:
44
45
46     private: System::Windows::Forms::Label^ lblOutput;
47     private: System::Windows::Forms::Button^ btnCalculate;
48     private: System::Windows::Forms::ErrorProvider^ errorProvider;
49     private: System::ComponentModel::IContainer^ components;
50

```

```

51
52
53     private:
54         ///  
55         ///  
56         ///  
57  
58  
59     #pragma region Windows Form Designer generated code  
60         ///  
61         ///  
62         ///  
63         ///  
64         void InitializeComponent(void)  
65         {  
66             this->components = (gcnew System::ComponentModel::Container  
67                 ());  
68             this->txtInput = (gcnew System::Windows::Forms::TextBox());  
69             this->txtOutput = (gcnew System::Windows::Forms::TextBox());  
70             this->lblInput = (gcnew System::Windows::Forms::Label());  
71             this->lblOutput = (gcnew System::Windows::Forms::Label());  
72             this->btnCalculate = (gcnew System::Windows::Forms::Button());  
73             this->errorProvider = (gcnew System::Windows::Forms::  
74                 ErrorProvider(this->components));  
75             (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->errorProvider))->BeginInit();  
76             this->SuspendLayout();  
77             //  
78             ///  
79             ///  
80             this->txtInput->Location = System::Drawing::Point(149, 12);  
81             this->txtInput->Name = L"txtInput";  
82             this->txtInput->Size = System::Drawing::Size(116, 20);  
83             this->txtInput->TabIndex = 0;  
84             //  
85             ///  
86             ///  
87             this->txtOutput->Location = System::Drawing::Point(149, 38);  
88             this->txtOutput->Name = L"txtOutput";  
89             this->txtOutput->ReadOnly = true;  
90             this->txtOutput->Size = System::Drawing::Size(116, 20);

```

```

89         this->txtOutput->TabIndex = 1;
90         //
91         // lblInput
92         //
93         this->lblInput->AutoSize = true;
94         this->lblInput->Location = System::Drawing::Point(12, 15);
95         this->lblInput->Name = L"lblInput";
96         this->lblInput->Size = System::Drawing::Size(114, 13);
97         this->lblInput->TabIndex = 2;
98         this->lblInput->Text = L"Введите_целое_число";
99         this->lblInput->Click += gnew System::EventHandler(this, &
            MyForm1::label1_Click);
100        //
101        // lblOutput
102        //
103        this->lblOutput->AutoSize = true;
104        this->lblOutput->Location = System::Drawing::Point(12, 41);
105        this->lblOutput->Name = L"lblOutput";
106        this->lblOutput->Size = System::Drawing::Size(59, 13);
107        this->lblOutput->TabIndex = 3;
108        this->lblOutput->Text = L"Результат";
109        //
110        // btnCalculate
111        //
112        this->btnCalculate->Location = System::Drawing::Point(149, 64);
113        this->btnCalculate->Name = L"btnCalculate";
114        this->btnCalculate->Size = System::Drawing::Size(75, 23);
115        this->btnCalculate->TabIndex = 4;
116        this->btnCalculate->Text = L"Вычислить";
117        this->btnCalculate->UseVisualStyleBackColor = true;
118        this->btnCalculate->Click += gnew System::EventHandler(this,
            &MyForm1::btnCalculate_Click);
119        //
120        // errorProvider
121        //
122        this->errorProvider->ContainerControl = this;
123        //
124        // MyForm1
125        //
126        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
127        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode

```



```

128         ::Font;
129         this->ClientSize = System::Drawing::Size(292, 101);
130         this->Controls->Add(this->btnCalculate);
131         this->Controls->Add(this->lblOutput);
132         this->Controls->Add(this->lblInput);
133         this->Controls->Add(this->txtOutput);
134         this->Controls->Add(this->txtInput);
135         this->Name = L"MyForm1";
136         this->Text = L"Факториал";
137         (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(
138             this->errorProvider))->EndInit();
139         this->ResumeLayout(false);
140         this->PerformLayout();
141     }
142     #pragma endregion
143     private: System::Void label1_Click(System::Object^ sender, System::EventArgs^ e)
144     {
145     }
146     private: System::Void btnCalculate_Click(System::Object^ sender, System::
147         EventArgs^ e) {
148         int InputNumber;
149         bool result = Int32::TryParse(this->txtInput->Text, InputNumber);
150         if (result) {
151             errorProvider->SetError(this->txtInput, String::Empty);
152             if (InputNumber < 0) {
153                 this->txtOutput->Text = "";
154                 errorProvider->SetError(this->txtInput, "Введено_
155                     отрицательное_число");
156             }
157             else if (InputNumber > 17) {
158                 this->txtOutput->Text = "";
159                 errorProvider->SetError(this->txtInput, "Выход_за_
160                     границы_типа");
161             }
162             else {
163                 double OutputNumber = fact(InputNumber);
164                 this->txtOutput->Text = System::Convert::ToString(
165                     OutputNumber);
166             }
167         }
168     }

```

```
162         else {
163             this->txtOutput->Text = "";
164             errorProvider->SetError(this->txtInput, "Введено_не_целое_
                число");
165         }
166     }
167 };
168 }
```