

VISUAL STUDIO

C#

VISUAL STUDIO (VS)

- **Integrated Development Environment (IDE)** – интегрированная среда разработчика
- IDE – это набор инструментов разработчика ПО, собранный в составе единого приложения и облегчающий труд программиста при написании приложений
- В центре VS находится среда программирования (платформа) – **.NET Framework**
- Платформа **.NET Framework** представляет среду управляемого выполнения, возможности упрощения разработки и развертывания, а также возможности интеграции со многими языками программирования

VISUAL STUDIO (VS)

- Включает языки программирования:
 - Visual Basic (VB)
 - Visual C#
 - Visual C++
 - Visual F#
- Существенный положительный эффект достигается при групповой разработке
- Над одним проектом могут работать программисты на C#, VB, C++, при этом среда обеспечивает совместимость программных частей, написанных на разных языках

Последние файлы .NET Framework 4.5 Сортировать по: По умолчанию

Установлено: Шаблоны - поиск (Ctrl-F)

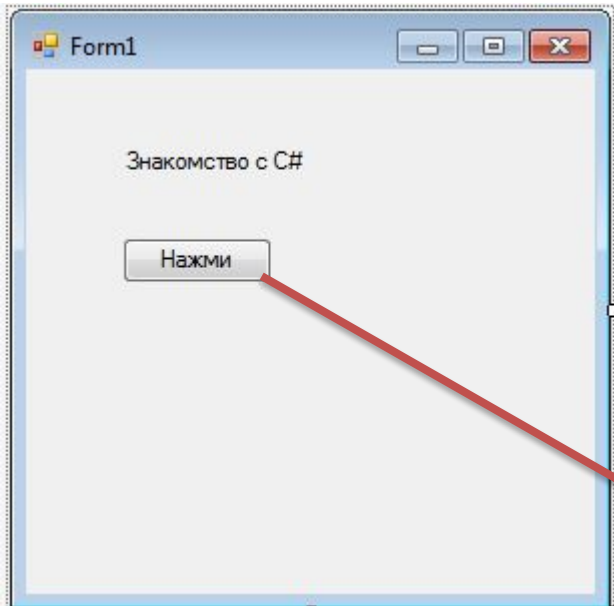
- Установленные
 - Шаблоны
 - Visual Basic
 - Windows
 - Веб
 - Office
 - Cloud
 - Reporting
 - SharePoint
 - Silverlight
 - WCF
 - Workflow
 - Тест
 - LightSwitch
 - Другие языки
 - Visual C#
 - Windows
 - Веб
 - Office
 - Cloud
 - Reporting
 - SharePoint
- В Интернете

Иконка	Название шаблона	Технология
	Приложение Windows Forms	Visual C#
	Приложение WPF	Visual C#
	Консольное приложение	Visual C#
	Библиотека классов	Visual C#
	Переносимая библиотека классов	Visual C#
	Приложение обозревателя WPF	Visual C#
	Библиотека настраиваемых элементов управления WPF	Visual C#
	Библиотека пользовательских элементов управления W...	Visual C#
	Пустой проект	Visual C#
	Служба Windows	Visual C#
	Библиотека элементов управления Windows Forms	Visual C#

Тип: Visual C#
Проект, для создания приложения с пользовательским интерфейсом Windows Forms

Имя: WindowsFormsApplication1

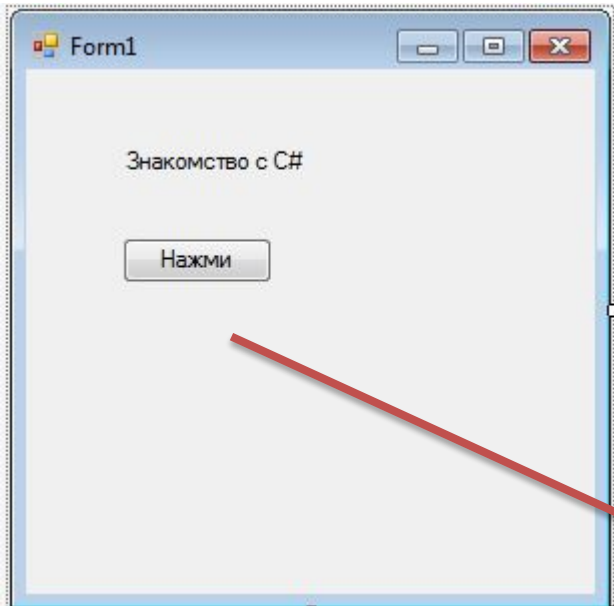
OK Отмена



```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProbaC1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```



```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProbaC1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Привет!");
        }
    }
}
```

Обратите внимание на ;

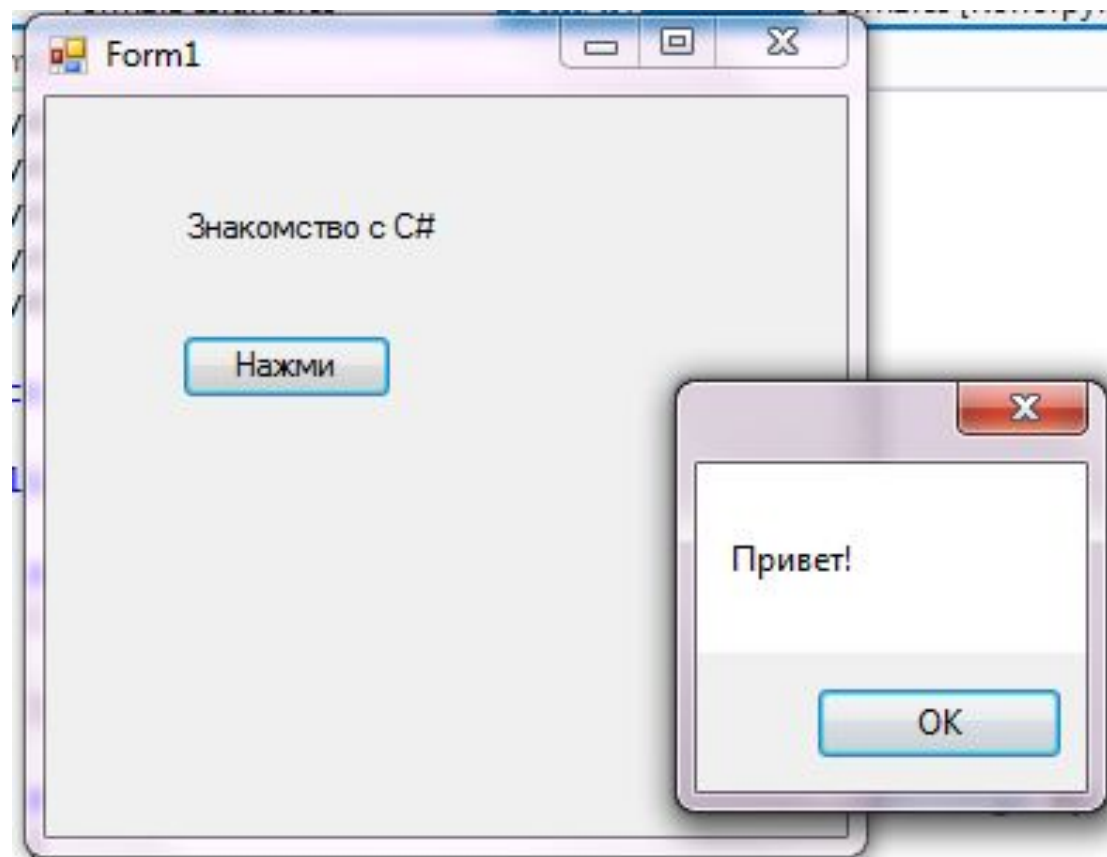
Язык C# чувствителен к регистру.

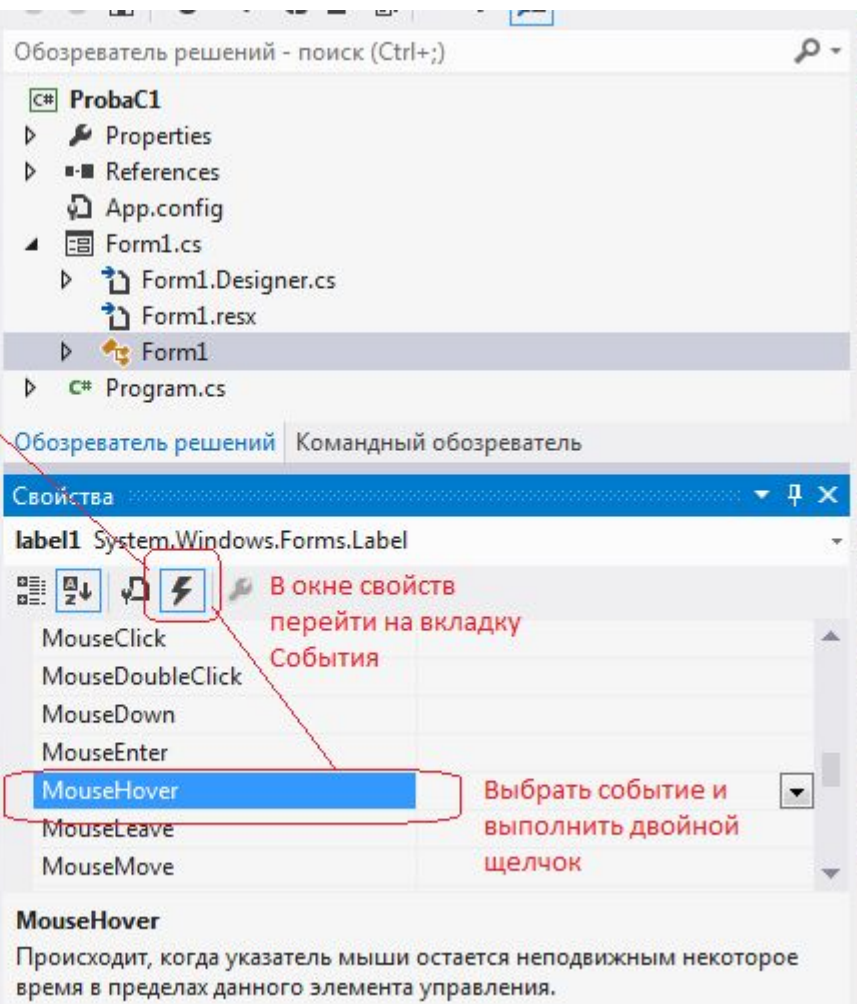
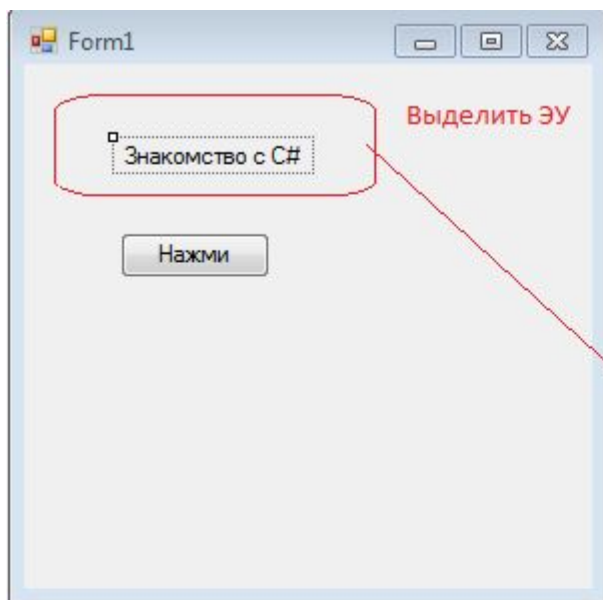
При вводе программ, написанных на языке C#, учитывайте *регистр*.

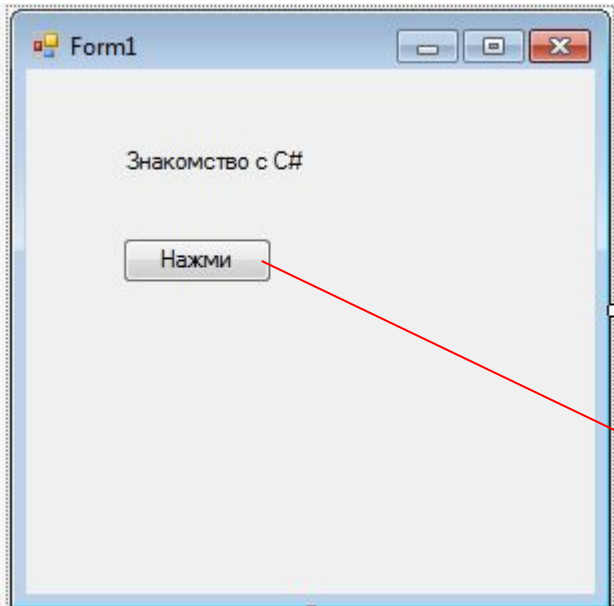
За именем функции следуют аргументы функции, заключенные в круглые скобки,

а после скобок стоит **точка с запятой**.

Аргументы разделяются запятыми.







```
1  
2 public partial class Form1 : Form  
3 {  
4     public Form1()  
5     {  
6         InitializeComponent();  
7     }  
8  
9     private void button1_Click(object sender, EventArgs e)  
10    {  
11        MessageBox.Show("Привет!");  
12    }  
13  
14    private void label1_MouseHover(object sender, EventArgs e)  
15    {  
16    }  
17 }  
18 }
```

```
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
    }  
  
    private void button1_Click(object sender, EventArgs e)  
    {  
        MessageBox.Show("Привет!");  
    }  
}
```

```
private void label1_MouseHover(object sender, EventArgs e)  
{  
    //Обработка события, когда указатель мыши зависает над меткой  
    MessageBox.Show("Событие метки");  
}
```

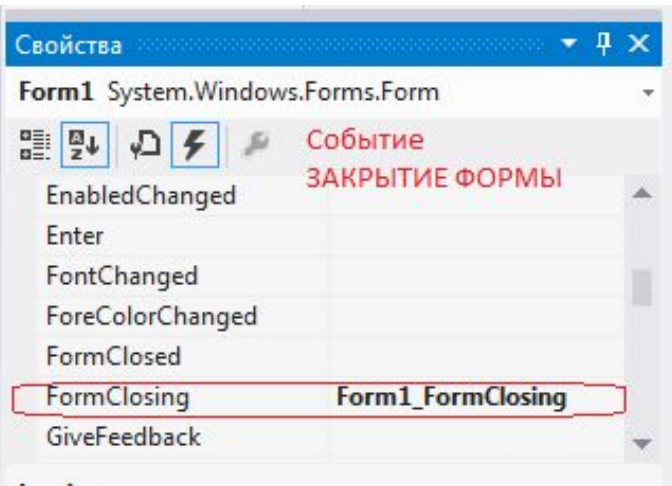
Комментарий
// одна строка
/* несколько строк
Несколько строк */

MessageBox

- Отображает окно сообщения, в котором могут содержаться текст, кнопки и символы, которые информируют пользователя и дают ему указания.

**MessageBox.Show(“текст_сообщения”,
“заголовок”, MessageBoxButtons.кнопки,
MessageBoxIcon.вид_значка)**

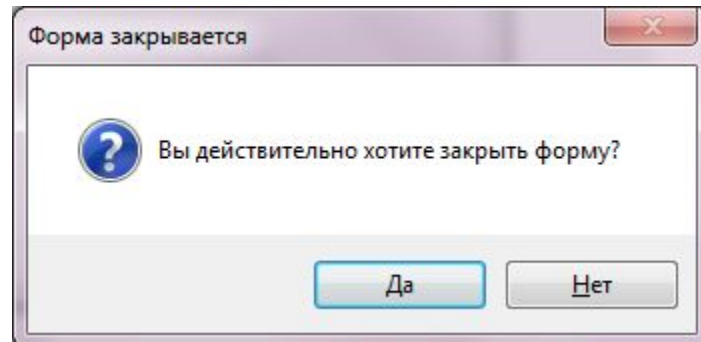
Использование свойства Заккрытие формы



```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    //описание констант
    const string message = "Вы действительно хотите закрыть форму?";
    const string caption = "Форма закрывается";
    //объявление переменной и присвоение ей значения
    var result = MessageBox.Show(message, caption, MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    // Если кнопка НЕТ нажата ...
    if (result == DialogResult.No)
    {
        // отмена закрытия формы

        e.Cancel = true;
    }
}
```



VB

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing
    Dim message As String = "Вы действительно хотите закрыть форму?"
    Dim caption As String = "Форма закрывается"
    Dim result = MessageBox.Show(message, caption, MessageBoxButtons.YesNo, MessageBoxIcon.Question)

    ' Если кнопка НЕТ нажата ...
    If (result = DialogResult.No) Then
        ' отмена закрытия формы.
        e.Cancel = True
    End If
End Sub
```

C#

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    //описание констант
    const string message = "Вы действительно хотите закрыть форму?";
    const string caption = "Форма закрывается";
    //объявление переменной и присвоение ей значения
    var result = MessageBox.Show(message, caption, MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    // Если кнопка НЕТ нажата ...
    if (result == DialogResult.No)
    {
        // отмена закрытия формы

        e.Cancel = true;
    }
}
```

Типы данных

VB	C#	.NET	
Byte	byte	Byte	8-битовое беззнаковое целое число
Sbyte	sbyte	Sbyte	8-битовое знаковое целое число
Integer	int	Int32	32-битовое знаковое целое число
Long	long	Int64	64-битовое знаковое целое число
Single	float	Single	32-битовое число с плавающей точкой
Double	double	Double	64-битовое число с плавающей точкой
Boolean	bool	Boolean	True/False
Char	char	Char	16 битовый символ unicode
Decimal	decimal	Decimal	96 битовое десятичное число (используется для указания денежных сумм)
String	string	String	Строка символов

Переменные

```
int a;  
// Объявление a.  
int b;  
// Объявление b.  
b = 10;  
// Инициализация b.  
a=b+b;  
// Инициализация a.  
int c = 0;  
// Объявление и инициализация c.
```

- `float temperature;`
- `string name;`
- `char firstLetter = 'C';`
- `var limit = 3;`
- `int[] source = { 0, 1, 2, 3, 4, 5 };`
- `Int32 age;`

- Ключевое слово **var** сообщает компилятору необходимости определения типа переменной из выражения, находящегося с правой стороны оператора

```
public struct CoOrds
{
    public int x, y;

    public CoOrds(int p1, int p2)
    {
        x = p1;
        y = p2;
    }
}
```


Константы

- `const float Pi = 3.14;`

Ветвление

C#

```
private void button2_Click(object sender, EventArgs e)
{
    int x, y;
    x = Convert.ToInt32(textBox1.Text);
    if (x < 40)
    {
        y=100;
    }
    else
    {
        y=200;
    }
    textBox2.Text = Convert.ToString(y);
}
```

VB

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim x, y As Integer
    x = Convert.ToInt32(TextBox1.Text)
    If x < 40 Then
        y = 100
    Else
        y = 200
    End If
    TextBox2.Text = Convert.ToInt32(y)
End Sub
```

Ветвление

C#

```
private void button3_Click(object sender, EventArgs e)
{
    string n;
    n = textBox1.Text;
    switch (n)
    {
        case "dog":
            textBox2.Text = "собака";
            break;
        case "cat":
            textBox2.Text = "кошка";
            break;
        case "bird":
            textBox2.Text = "птичка";
            break;
        case "fox":
            textBox2.Text = "лиса";
            break;
        default:
            textBox2.Text = "не знаю";
            break;
    }
}
```

VB

```
Private Sub Button2_Click(sender As Object, e As EventArgs)
    Dim n As String
    n = TextBox1.Text
    Select n

        Case "dog"
            TextBox2.Text = "собака"
        Case "cat"
            TextBox2.Text = "кошка"
        Case "bird"
            TextBox2.Text = "птичка"
        Case "fox"
            TextBox2.Text = "лиса"
        Case Else
            TextBox2.Text = "не знаю"
    End Select

End Sub
```

ЦИКЛЫ

C#

```
private void button4_Click(object sender, EventArgs e)
{
    int n, m, k;
    //n число
    n = Convert.ToInt32(textBox1.Text);
    //k степень
    k = Convert.ToInt32(textBox2.Text);
    //m результат возведения числа в степень
    m = 1;
    for (int i = 0; i < k; i++)
    {
        m = m * n;
    }
    textBox3.Text = Convert.ToString(m);
}
```

VB

```
Private Sub Button3_Click(sender As Object, e As
    Dim n, m, k As Integer
    n = TextBox1.Text
    k = TextBox2.Text
    m = 1
    For i = 1 To k
        m = m * n
    Next
    TextBox3.Text = m
End Sub
```

ЦИКЛЫ

```
private void button5_Click(object sender, EventArgs e)
{
    int n, m, k;
    //n число
    n = Convert.ToInt32(textBox1.Text);
    //k степень
    k = Convert.ToInt32(textBox2.Text);
    //m результат возведения числа в степень
    m = 1;
    while (k>0)
    {
        m=m*n;
        k--;
    }
    textBox3.Text = Convert.ToString(m);
}
```

ЦИКЛЫ

```
private void button6_Click(object sender, EventArgs e)
{
    int n, m, k;
    //n число
    n = Convert.ToInt32(textBox1.Text);
    //k степень
    k = Convert.ToInt32(textBox2.Text);
    //m результат возведения числа в степень
    m = 1;
    do
    {
        m = m * n;
        k--;
    } while (k > 0);
    textBox3.Text = Convert.ToString(m);
}
```

```
private void button7_Click(object sender, EventArgs e)
{
    int n, m;
    Random randomizer = new Random();
    m = 0;
    do
    {
        n = randomizer.Next(6)+1;
        //выпало число
        MessageBox.Show(Convert.ToString(n));
        m++;
    } while (n < 6);
    //количество попыток
    MessageBox.Show(Convert.ToString(m));
}
```

Задание стартовой формы

The image shows a screenshot of the Visual Studio IDE. The main window displays the code for `Program.cs` in the `ProbaC1` namespace. The code defines a static class `Program` with a `Main` method. The `Main` method is annotated with `[STAThread]` and contains the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace ProbaC1
8 {
9     static class Program
10    {
11        /// <summary>
12        /// Главная точка входа для приложения.
13        /// </summary>
14        [STAThread]
15        static void Main()
16        {
17            Application.EnableVisualStyles();
18            Application.SetCompatibleTextRenderingDefault(false);
19            Application.Run(new Form1());
20        }
21    }
22 }
```

A red arrow points to the `new Form1()` argument in the `Application.Run` call on line 19, with the text "задание стартовой формы" (assignment of the start form) written in red below it.

The right-hand side of the image shows the Solution Explorer. The project `ProbaC1` is expanded, showing the following files:

- Properties
- References
- App.config
- Form1.cs
 - Form1.Designer.cs
 - Form1.resx
- Form1
- Form2.cs
- Program.cs (highlighted)

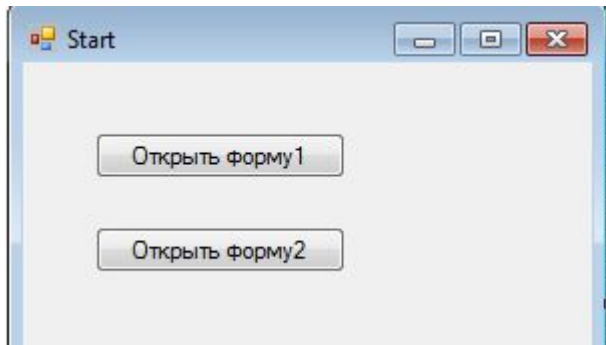
At the bottom of the Solution Explorer, the "Обозреватель решений" (Solution Explorer) tab is active, and the "Свойства" (Properties) window is partially visible.

Открытие формы

- Ошибка
- В C# **Form2** – это название класса, а не переменная
- Необходимо создать переменную и работать с ней

```
private void button1_Click(object sender, EventArgs e)
{
    Form2.Show();
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 secondForm = new Form2();
    secondForm.Show();
}
```

```
ProbaC1.Program Main()
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace ProbaC1
8 {
9     static class Program
10    {
11        /// <summary>
12        /// Главная точка входа для приложения.
13        /// </summary>
14        [STAThread]
15        static void Main()
16        {
17            Application.EnableVisualStyles();
18            Application.SetCompatibleTextRenderingDefault(false);
19            Application.Run(new Start());
20        }
21    }
22 }
```

```
private void button1_Click(object sender, EventArgs e)
{
    Form1 myform = new Form1();
    myform.Show();
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    Form2 myform = new Form2();
    myform.Show();
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    //закреть приложение (в VB me.Close())
    this.Close();
}
```

Открытие формы в модальном режиме

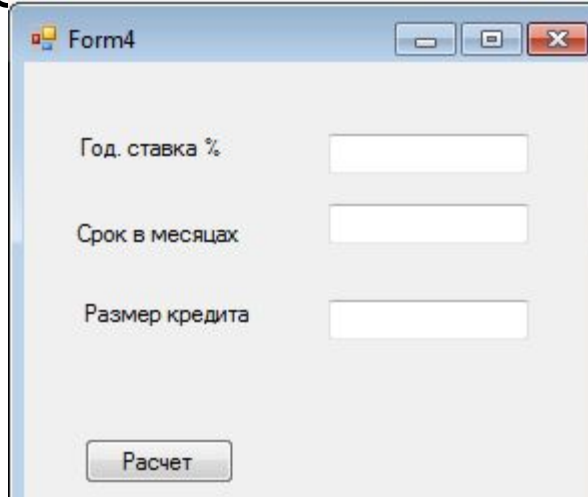
- Для открытия формы в модальном режиме, т.е., чтобы нельзя было переключиться на другую, пока открыта текущая необходимо вызвать метод **ShowDialog()**.

```
private void button2_Click(object sender, EventArgs e)
{
    Form2 myform = new Form2();
    myform.Show();
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    //модальная форма
    Form2 myform = new Form2();
    myform.ShowDialog();
}
```

Использование финансовых функций Excel

- Расчет кредита
 - функция Excel ПЛТ()
 - функция (метод) Pmt() класса WorksheetsFunction



The image shows a screenshot of a Windows application window titled "Form4". The window contains three input fields for data entry, each with a corresponding label to its left:

- Label: "Год. ставка %" (Annual interest rate %)
- Label: "Срок в месяцах" (Term in months)
- Label: "Размер кредита" (Loan amount)

At the bottom of the form, there is a button labeled "Расчет" (Calculate).

Использование финансовых функций Excel

Менеджер ссылок - ProbaC1

Целевая платформа: .NET Framework 4.5

Сборки - поиск (Ctrl+E)

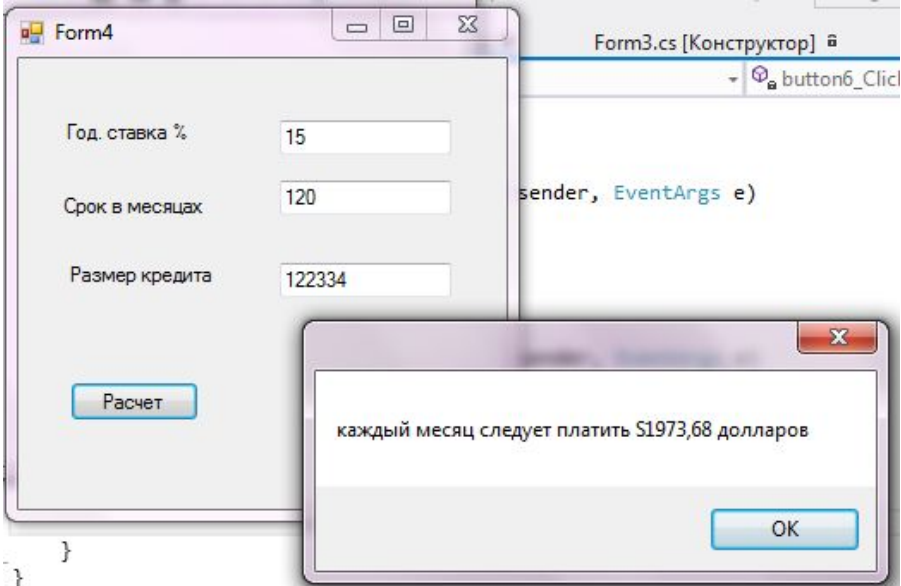
Имя: Microsoft.Office.Interop.Excel
Автор: Microsoft Corporation
Версия: 14.0.0.0
Версия файла: 14.0.4756.1000

Платформа	Имя	Версия
	Microsoft.Expression.Prototyping.Interactivity	4.5.0.0
	Microsoft.Expression.Prototyping.Interactivity	4.0.0.0
	Microsoft.Expression.Prototyping.SketchContr...	4.0.0.0
	Microsoft.Expression.Prototyping.SketchContr...	4.5.0.0
	Microsoft.Expression.SilverlightPlatform	5.0.30709.0
	Microsoft.Expression.SourceControl.TFS	5.0.30709.0
	Microsoft.Expression.Utility	5.0.30709.0
	Microsoft.Expression.VisualStudioAutomation	5.0.30709.0
	Microsoft.Expression.WebLanguageServices	5.0.30709.0
	Microsoft.Expression.WebPlatform	5.0.30709.0
	Microsoft.Expression.WindowsPhone	5.0.30709.0
	Microsoft.Expression.WindowsPhonePlatform	5.0.30709.0
	Microsoft.Expression.WindowsXamlPlatform	5.0.30709.0
	Microsoft.Expression.WpfPlatform	5.0.30709.0
	Microsoft.mshtml	7.0.3300.0
	Microsoft.mshtml	7.0.3300.0
	Microsoft.MSXML	8.0.0.0
	Microsoft.Office.InfoPath.Permission	14.0.0.0
	Microsoft.Office.Interop.Access	14.0.0.0
	Microsoft.Office.Interop.Access.Dao	14.0.0.0
	Microsoft.Office.Interop.Excel	14.0.0.0
	Microsoft.Office.Interop.Graph	14.0.0.0
	Microsoft.Office.Interop.InfoPath	14.0.0.0
	Microsoft.Office.Interop.InfoPath.SemiTrust	11.0.0.0
	Microsoft.Office.Interop.InfoPath.Xml	14.0.0.0
	Microsoft.Office.Interop.MSProject	14.0.0.0

Добавить ссылку...

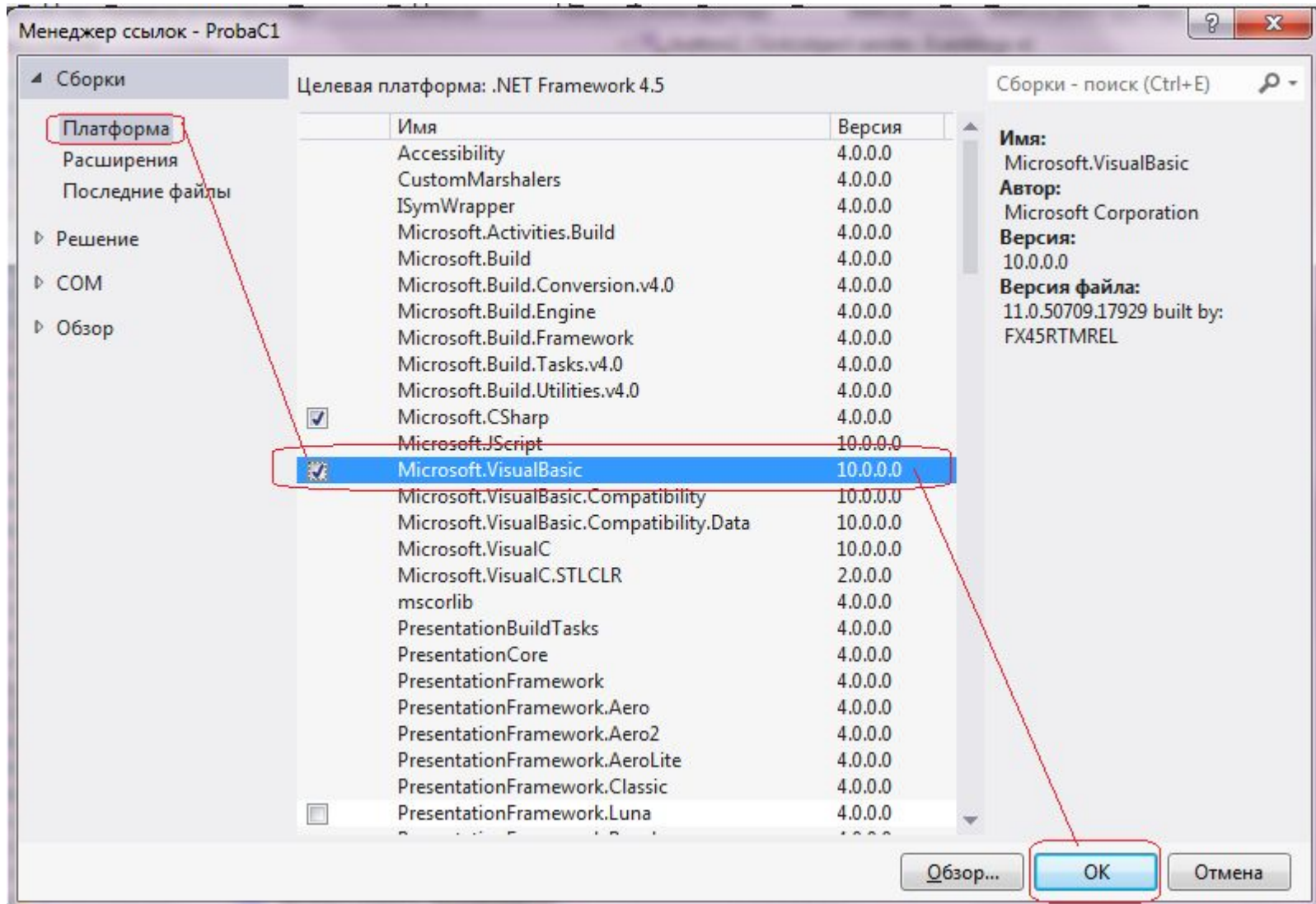
OK

Использование финансовых функций Excel



```
private void button1_Click(object sender, EventArgs e)
{
    double a, b, c;
    //обработчик ошибок
    try
    {
        a = Convert.ToDouble(textBox1.Text); //ставка
        b = Convert.ToDouble(textBox2.Text); //срок
        c = Convert.ToDouble(textBox3.Text); //размер
        var XL = new Microsoft.Office.Interop.Excel.Application();
        //переменная с пустым значением
        var t = Type.Missing;
        //получаем размер месячного платежа
        var pay = XL.WorksheetFunction.Pmt(a/1200, b, c, t, t);
        var stra = string.Format("каждый месяц следует платить {0:S#.##} долларов", Math.Abs(pay));
        MessageBox.Show(stra);
        XL.Quit();
    }
    //обработка ошибки
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "ошибка", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

Вызов функции Visual Basic



Вызов функции Visual Basic

```
private void button2_Click(object sender, EventArgs e)
{
    double a, b, c;
    //обработчик ошибок
    try
    {
        a = Convert.ToDouble(textBox1.Text); //ставка
        b = Convert.ToDouble(textBox2.Text); //срок
        c = Convert.ToDouble(textBox3.Text); //размер

        //использование функции из Visual Basic
        var FV = 0.0;
        var dt = Microsoft.VisualBasic.DueDate.EndOfPeriod;
        var pay = Microsoft.VisualBasic.Financial.Pmt(a / 1200, b, c, FV, dt);
        var stra = string.Format("каждый месяц следует платить {0:S#.##} долларов", Math.Abs(pay));
        MessageBox.Show(stra);
    }
    //обработка ошибки
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "ошибка", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

Обработки исключений

- Вначале выполняются все инструкции между операторами `try` и `catch`.
- Если между этими операторами вдруг возникает исключение, то обычный порядок выполнения останавливается и переходит к инструкции `catch`.
- Инструкция `catch` имеет следующий синтаксис: `catch (тип_исключения имя_переменной)`
- `catch (Exception ex)` будет обрабатывать практически все исключения
- В любом случае выполняется блок `finally`. Однако этот блок необязательный, и его можно при обработке исключений опускать.

Схема обработки исключений в C#

```
Try
  {//нормальное выполнение...}
catch (Exception e)
{
    MessageBox.Show("Ошибка: " + e.Message);
}

catch (T1 e1)
{...}
...
catch(Tk ek)
{...}
finally {...}
```