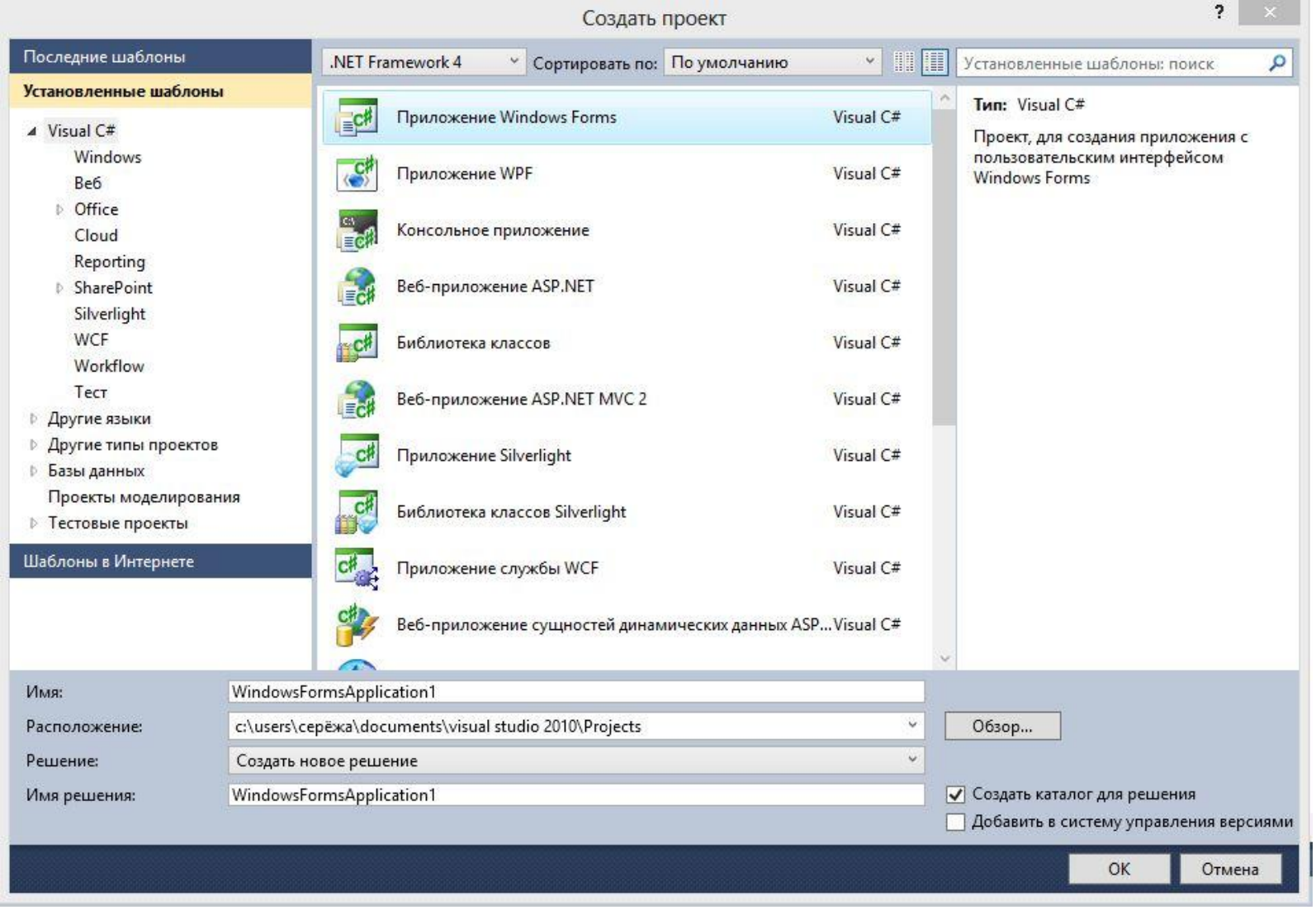
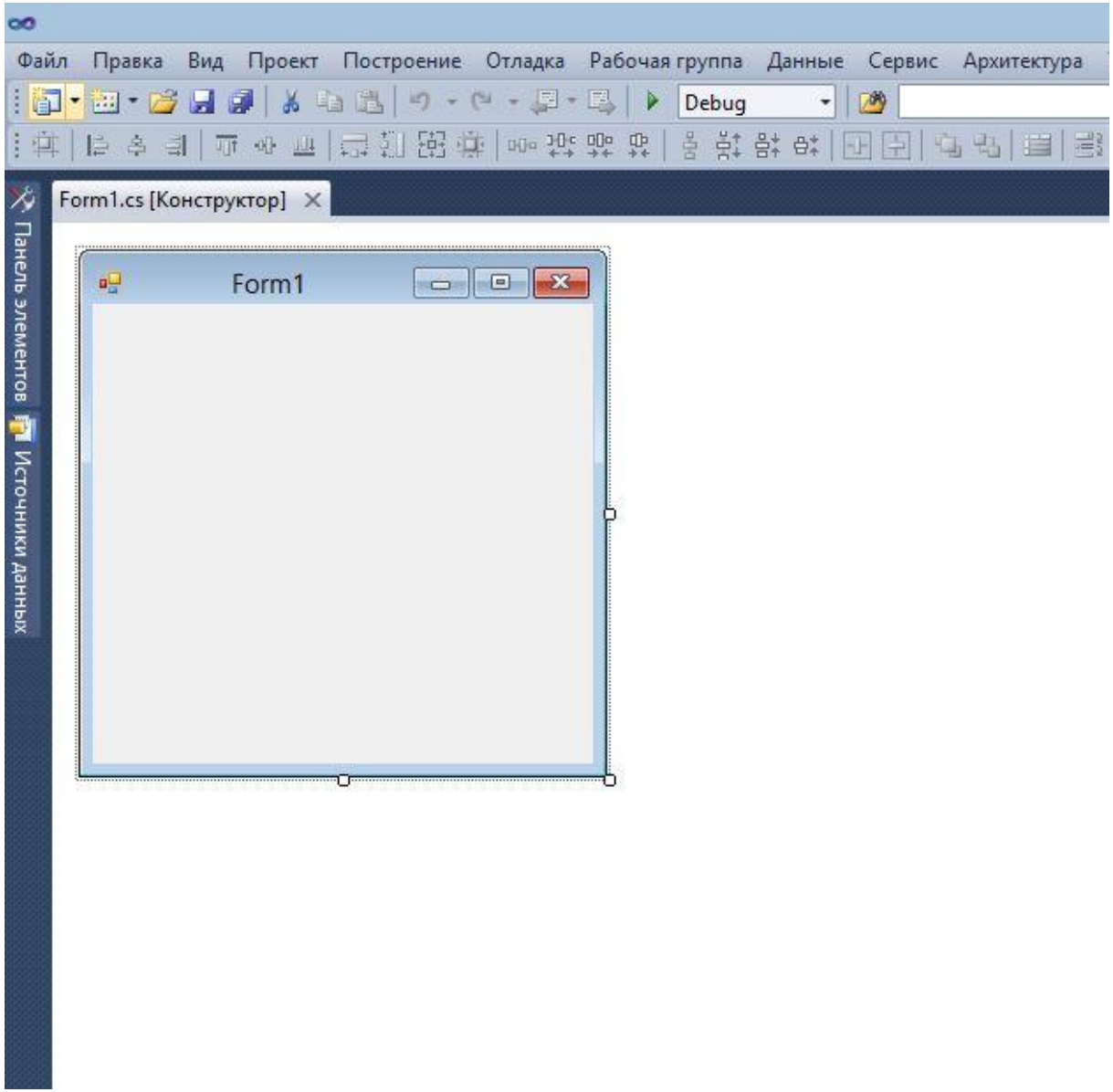
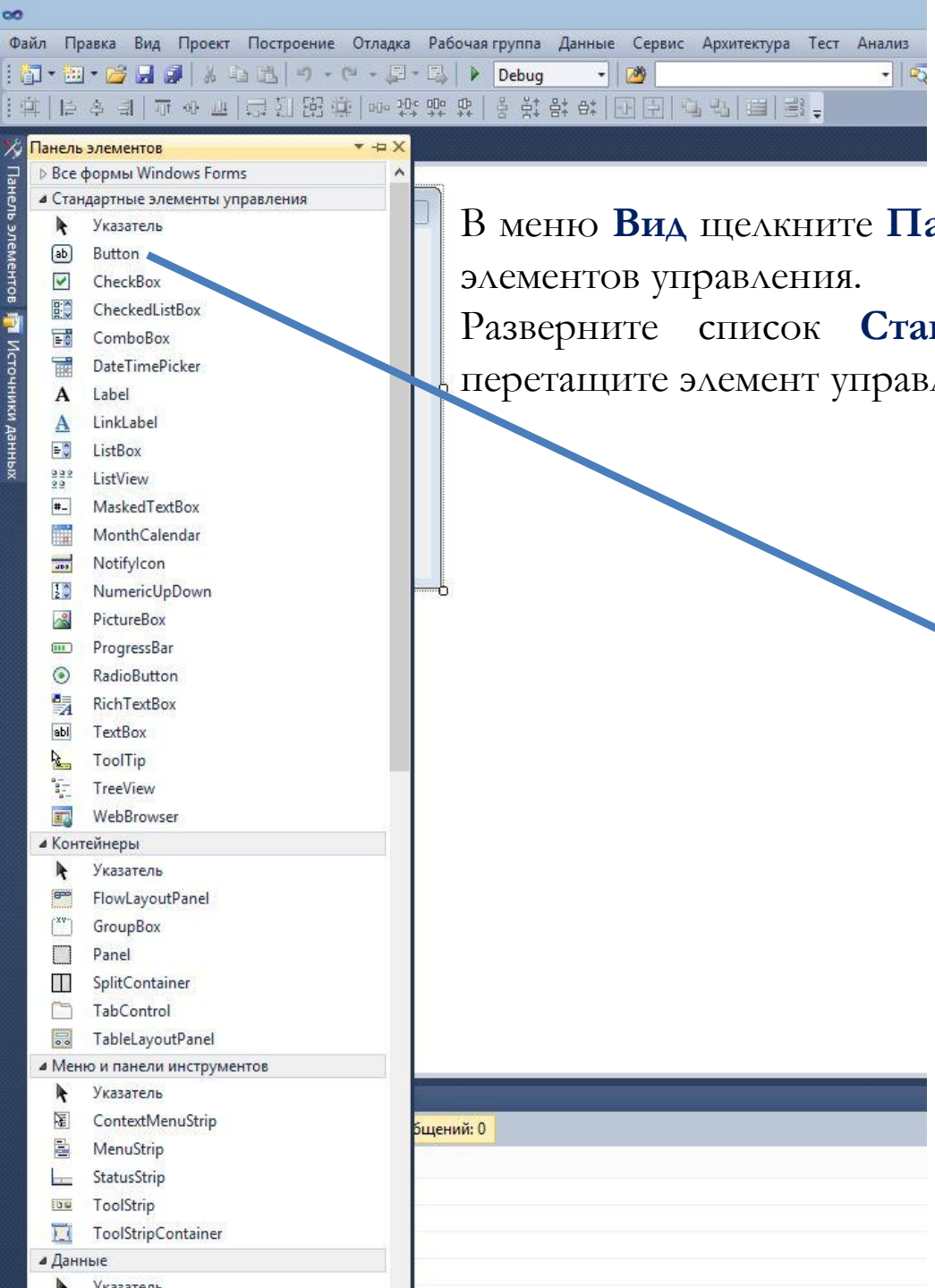


В меню **Файл** выберите команду **Создать** и щелкните **Проект**.

Удостоверьтесь, что выбран шаблон **Приложение Windows Forms**, в поле **Имя** введите **КАЛЬКУЛЯТОР** и нажмите кнопку ОК.

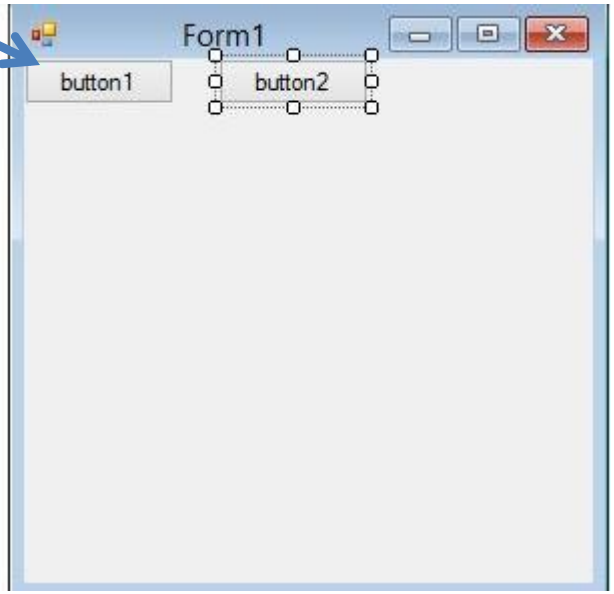


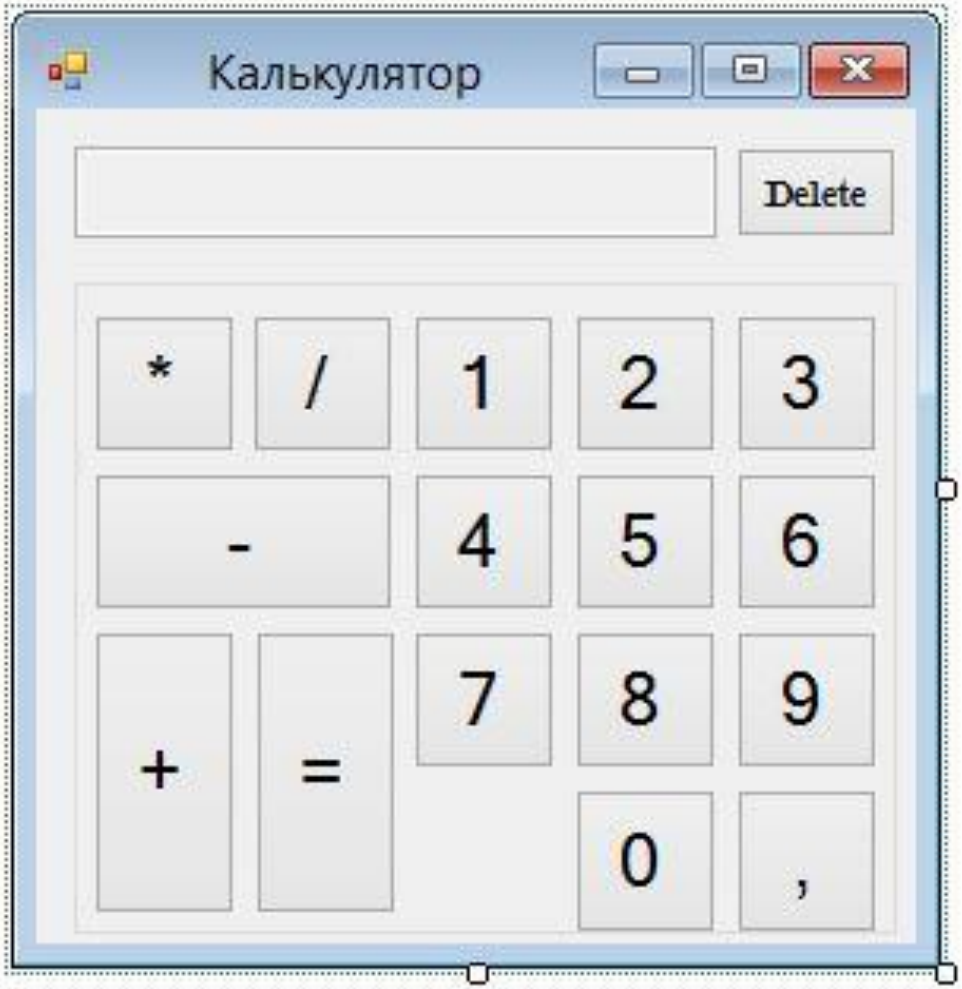
Откроется конструктор **Windows Forms** с формой **Windows**.
Это пользовательский интерфейс для создаваемого приложения.



В меню **Вид** щелкните **Панель элементов**, чтобы открыть список элементов управления.

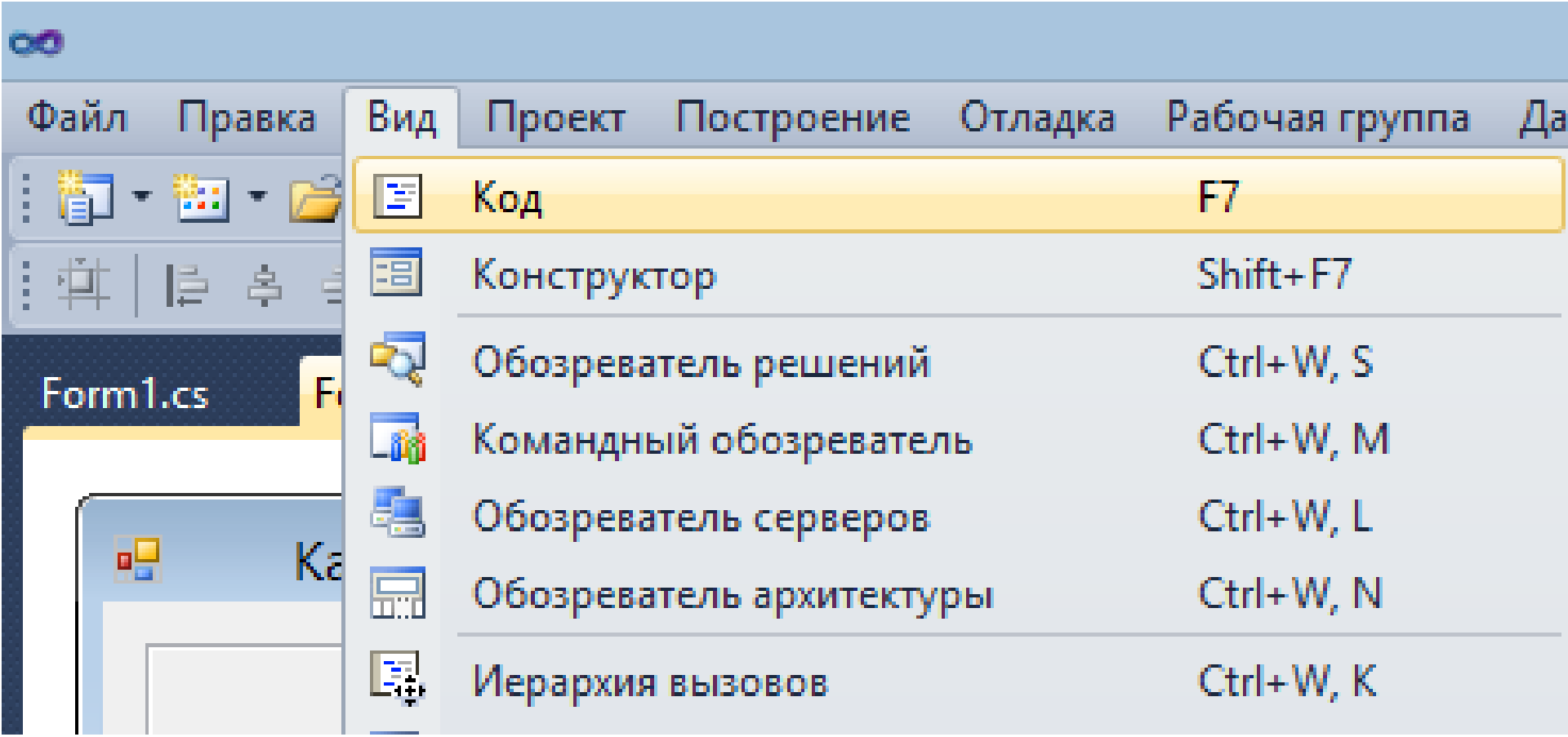
Разверните список **Стандартные элементы управления** и перетащите элемент управления кнопку





Создайте все нужные нам элементы.

В данном случае это : кнопки, текстовое поле.



Для перехода к написанию коду нажмите Вид/Код

- Подключаем нужные нам библиотеки , в данном случае идут стандартные:

```
using System;
using System.Text;
using System.Windows.Forms;
```

- Глобальные переменные:

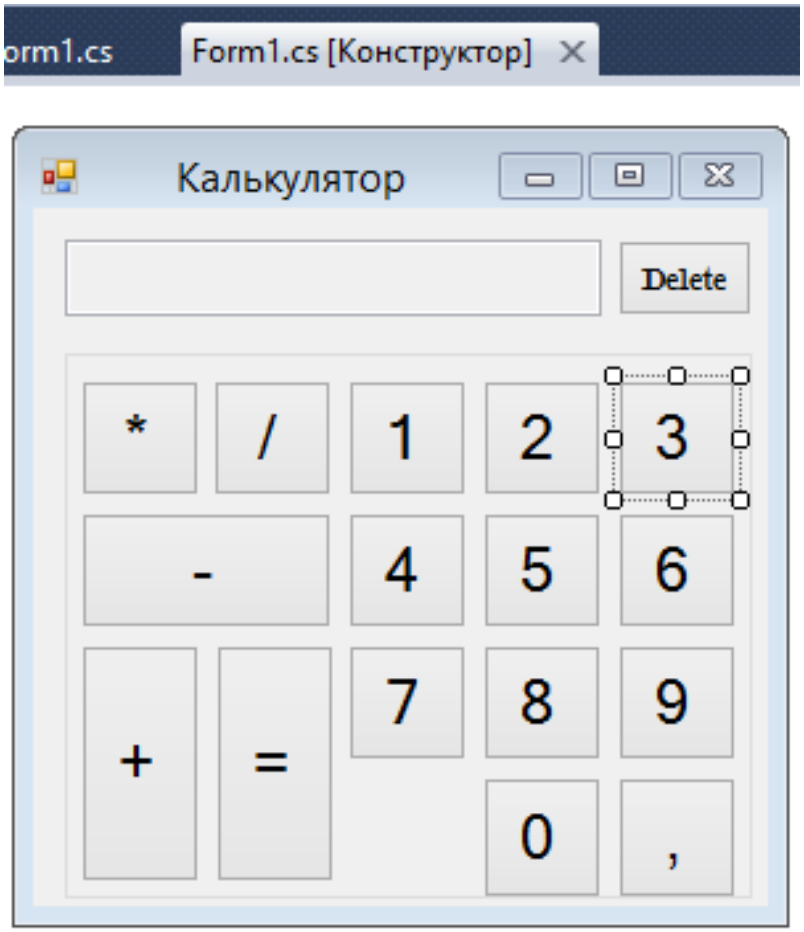
```
double first, second,result;
 Operat oper;
 bool fl = false, perv=false;
```

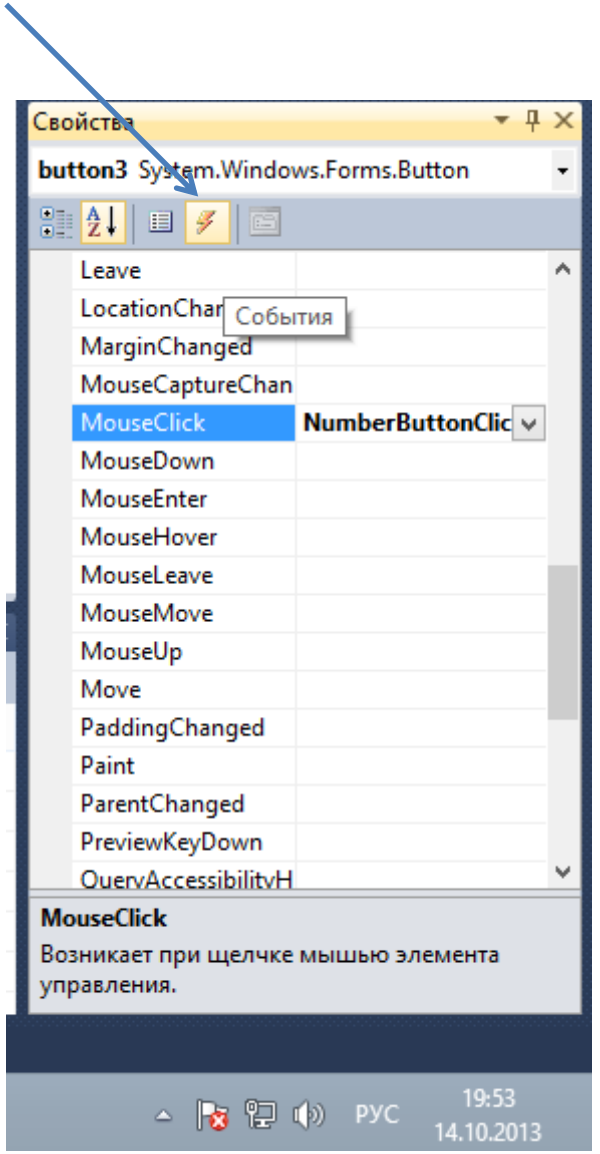
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Калькулятор
{
    public partial class Form1 : Form
    {
        enum Operat
        {
            Multiply,Divide,Plus,Minus
        }
        public Form1()
        {
            InitializeComponent();
        }

        double first, second,result;
        Operat oper;
        bool fl = false, perv=false;
```

Объявим перечисления отдельного типа с помощью ключевого слова **enum,** состоящего из набора именованных констант, который называется списком перечислителя.

```
enum Operat//перечислимый тип для операций может
{//принимать 4 состояния по числу доступных операций
    Multiply,Divide,Plus,Minus
}
```

Для каждой кнопки создадим обработчик нажатия мыши, выделив кнопку и в свойствах кнопки нажав «**События**»:





Можно для каждой кнопки цифры задать свой метод. Однако, т.к. нажатие каждой кнопки обрабатывается одинаково, мы уменьшим количества кода. Для всех цифр обработчик (метод) будет следующий:

```
private void NumberButtonClick(object sender, MouseEventArgs e)//Событие нажатия цифры
{
    if (Terminal.Text != result.ToString() || fl)//Проверяем не
        //результат ли вычислений на табле дублируем проверку
        //припомощи флага,потому как пользователь может просто
        //ввести последний полученный результат
    {
        fl = true;
        Terminal.Text += (sender as Button).Text;//Дописываем текст(цифру)с
        //нажатой кнопки(с цифрой)
    }
    else
        Terminal.Text = (sender as Button).Text;//в случае,если на табле
    //все же последний результа,то затираем его и пишем сверху
}
```

Также сделаем обработчик нажатия кнопки DEL:

```
private void Delete_Click(object sender, EventArgs e)//Событие нажатия кнопки Delete
{
    if(!string.IsNullOrEmpty(Terminal.Text))//Проверка не пустое ли табло(здесь и далее)
    Terminal.Text = Terminal.Text.Substring(0, Terminal.Text.Length - 1);//Удаляем один символ
}
```

Опишем метод который будет проводить вычисления с операндами:

```
void Operatio(Operat op)//Метод для упрощения кода,который помогает обрабатывать нажатия кнопок мат операций
{
    if (!string.IsNullOrEmpty(Terminal.Text))
        if (!perv)//проверка нету ли незаконченых операций
        {
            first = Double.Parse(Terminal.Text);//первым операндом становится число из табла
            oper = op;//запоминаем последнюю операцию
            Terminal.Text = "";//Очищаем табло
            perv = !perv;//меняем состояние флага "идет цепочка операций" на обратное
        }
        else//если происходит "длинная" операция
        {
            second = Double.Parse(Terminal.Text);//второй операнд запоминаем  табла
            switch (oper)//Выполняем последнюю операцию,но в отличии от кнопки
                //равно запоминаем результат в первый операнд
            {
                case Operat.Divide: first = (first / second);
                    break;
                case Operat.Multiply: first = (first * second);
                    break;
                case Operat.Plus: first = (first + second);
                    break;
                case Operat.Minus: first = (first - second);
                    break;
            }
            oper = op;//Запоминаем операцию
            Terminal.Text = "";//Очищаем табло
        }
}
```

Для кнопки «=», создадим обработчик нажатия:

```
private void Result_Click(object sender, EventArgs e)//Обработчик кнопки равно
{
    if (!string.IsNullOrEmpty(Terminal.Text))
    {
        fl = false;//ставим флаг в положение"последний результат считан"
        second = Double.Parse(Terminal.Text);//Считываем и запоминаем второй операнд с табла
        switch (oper)//производим последнюю операцию
        {
            case Operat.Divide: result = (first / second);
                break;
            case Operat.Multiply: result = (first * second);
                break;
            case Operat.Plus: result = (first + second);
                break;
            case Operat.Minus: result = (first - second);
                break;
        }
        Terminal.Text=result.ToString();//записываем на табло резльтат вычислений
        perv = !perv;//меняем состояние флага "идет цепочка операций" на обратное
    }
}
```

Как видно из кода это обработчик вызывает выполнение метода описанного в предыдущем слайде.

По нажатию кнопок «*», «-», «+», «/» создадим по одному обработчику нажатия клавиши:

```
//Далее следует 4 обработчика события нажатия кнопки мат операций
//с использованием метода Operatio в качестве параметра передается тип операции
private void Multiply_Click(object sender, EventArgs e)
{
    Operatio(Operat.Multiply);
}

private void Divide_Click(object sender, EventArgs e)
{
    Operatio(Operat.Divide);
}

private void Minus_Click(object sender, EventArgs e)
{
    Operatio(Operat.Minus);
}

private void Plus_Click(object sender, EventArgs e)
{
    Operatio(Operat.Plus);
}
```

Как видно из кода, эти обработчики также вызывают выполнение метода Operatio.