

Тема 1. Базовые понятия и определения

Цель работы. Рассмотреть базовые понятия и определения, элементарные типы данных, основные операторы и выражения языка C#. Познакомиться с примерами простейших программ, демонстрирующих приемы работы с элементарными типами данных и операторами C#.

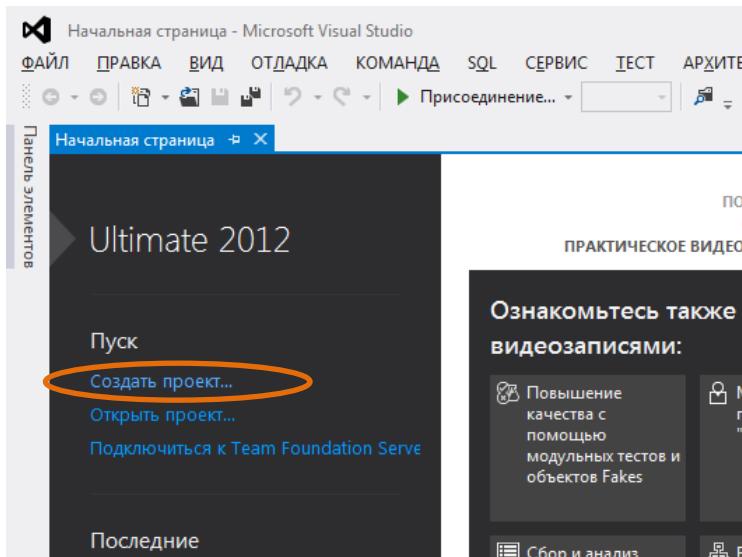
Программирование в Windows Forms. Создание Windows приложений на C#

Для создания графических интерфейсов с помощью платформы .NET применяются разные технологии - Window Forms, WPF, приложения для магазина Windows Store (для ОС Windows 8/8.1/10). Однако наиболее простой и удобной платформой до сих пор остается Window Forms или формы.

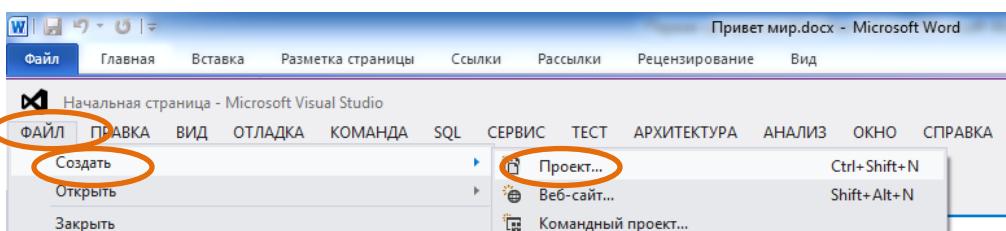
Задание 2. Создание графического приложения «Привет мир!»

Цель данной работы - понять принципы создания графических интерфейсов с помощью технологии WinForms и познакомиться с работой основных элементов управления.

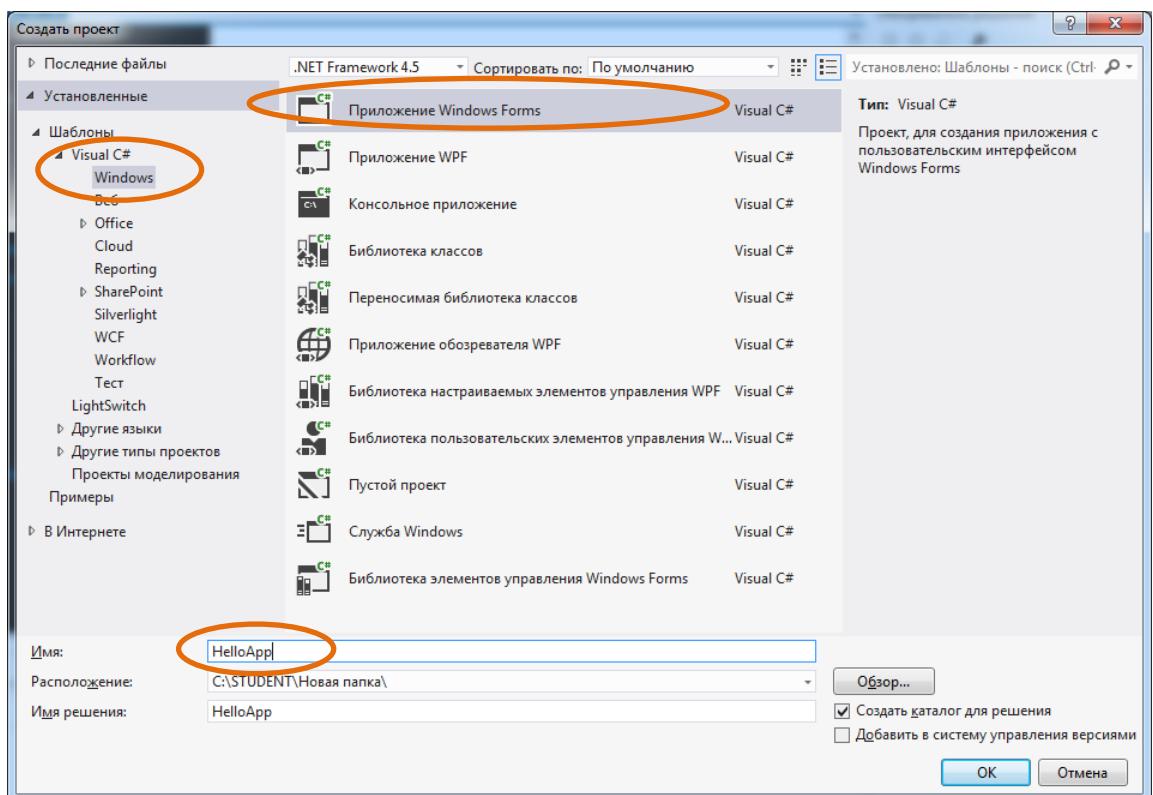
Для создания первого графического проекта потребуется среда разработки Visual Studio. Запускаем Visual Studio и создаем проект графического приложения.



либо

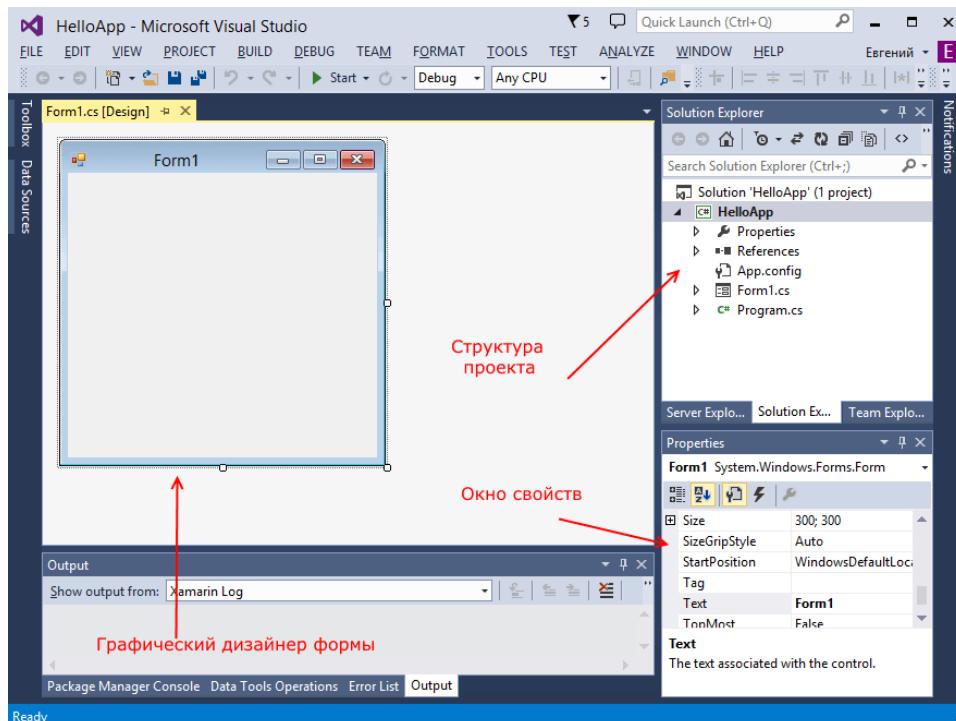


После этого откроется диалоговое окно создания нового проекта:



В левой колонке выберем язык С#, далее Windows, а в центральной части среди типов проектов - тип **Приложение Windows Forms** и дадим ему какое-нибудь имя в поле внизу. Например, назовем его *HelloApp*. После этого нажимаем OK.

После этого Visual Studio откроет проект с созданными по умолчанию файлами:

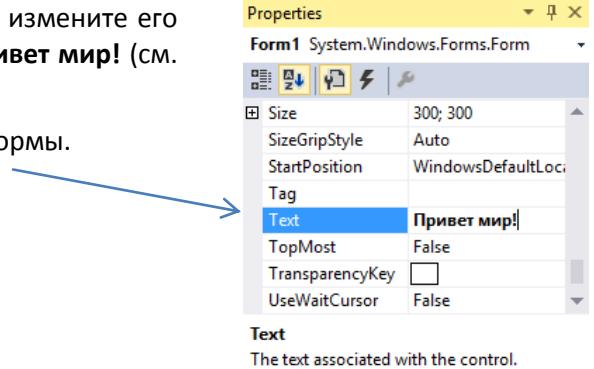


Большую часть пространства Visual Studio занимает графический дизайнер, который содержит форму будущего приложения. Пока она пуста и имеет только заголовок Form1. Справа находится окно файлов решения/проекта - Solution Explorer (Обозреватель решений). Там и находятся все связанные с нашим приложением файлы, в том числе файлы формы *Form1.cs*.

Внизу справа находится окно свойств - Properties. Так как в данный момент выбрана форма как элемент управления, то в этом поле отображаются свойства, связанные с формой.

Найдите в этом окне свойство формы **Text** и измените его значение на любое другое, например, на фразу **Привет мир!** (см. рис.).

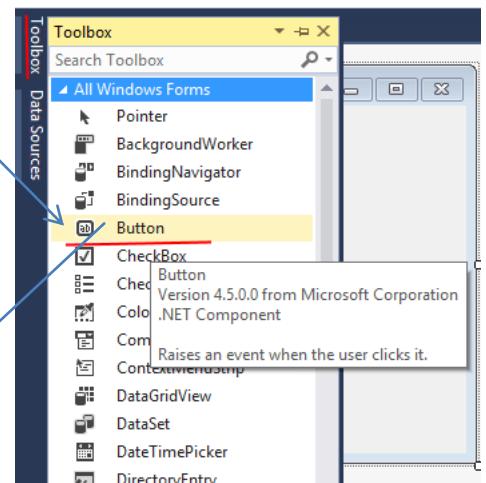
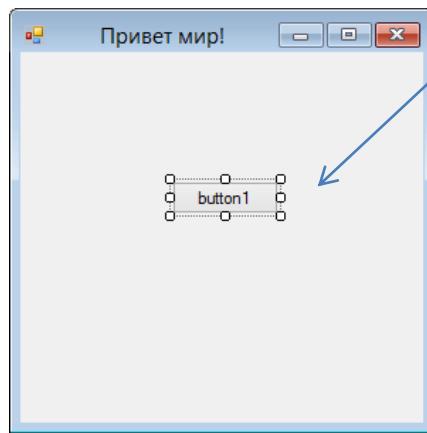
Таким образом, можно поменять заголовок формы.



Теперь перенесем на поле какой-нибудь элемент управления, например, **кнопку**.

Для этого найдите в левой части Visual Studio вкладку **Toolbox** (Панель инструментов).

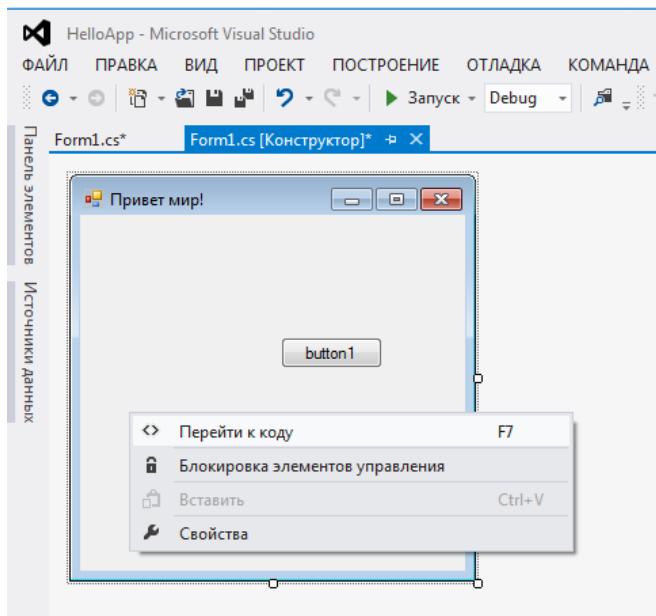
Нажмите на эту вкладку, чтобы открыть панель с элементами, откуда можно с помощью мыши перенести на форму любой элемент.



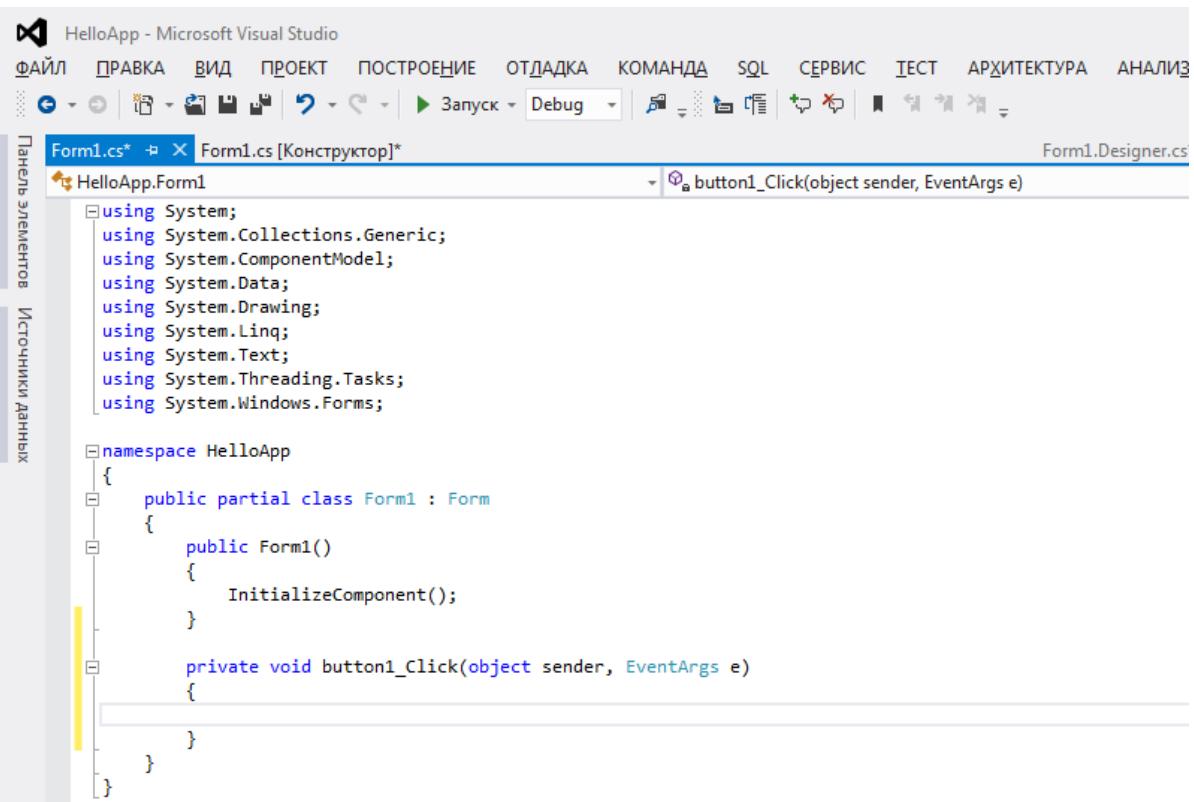
Найдем среди элементов кнопку и, захватив ее указателем мыши, перенесем на форму.

Это визуальная часть создания приложения. Теперь приступим непосредственно к программированию. Добавим простейший код на языке C#, который бы выводил сообщение по нажатию кнопки.

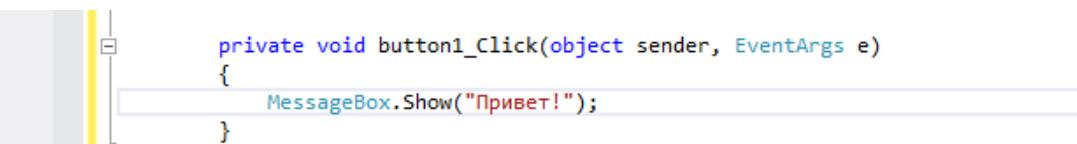
Для этого нужно перейти в файл кода, который связан с формой. Чтобы открыть файл кода, можем нажать на форму правой кнопкой мыши и в появившемся меню выбрать **View Code** (Посмотреть файл кода):



Однако чтобы не писать много лишнего кода, удобно воспользоваться другим способом. Наведем указатель мыши на кнопку и щелкнем по ней двойным щелчком. Мы автоматически попадаем в файл кода **Form1.cs**, который выглядит так:

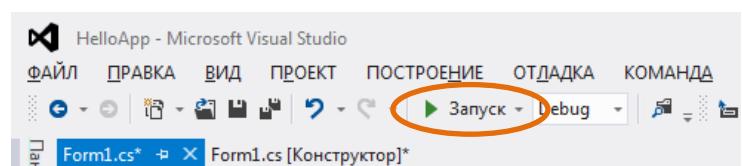


Добавим вывод сообщения по нажатию кнопки, изменив код следующим образом:



Запуск приложения

Чтобы запустить приложение в режиме отладки, нужно нажать клавишу F5 или на зеленую стрелочку на панели Visual Studio.



После этого запустится наша форма с одинокой кнопкой. И если мы нажмем на кнопку на форме, то нам будет отображено сообщение с приветствием.

После запуска приложения студия компилирует его в файл с расширением exe. Найти данный файл можно, зайдя в папку проекта и далее в каталог bin/Debug или bin/Release.

Самостоятельно

Попробуйте изменить название кнопки, например, на «**Нажми меня...**».

Задание 3. Дизайн формы и создание события

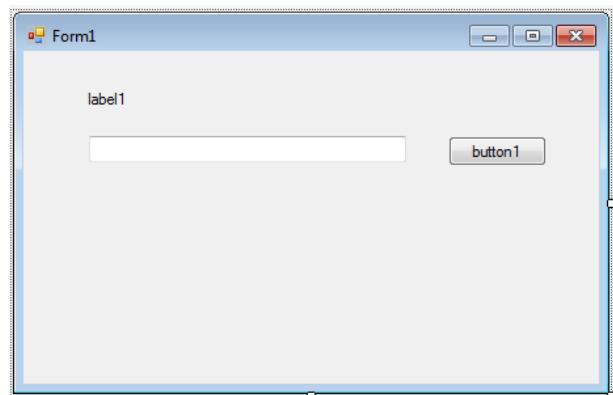
Создадим простую программу, которая будет иметь строковый редактор текста, кнопку и метку (текстовое поле). Программа должна будет, при нажатии на кнопку, скопировать текст, введённый в редактор текста на метку.

Создайте новый проект (имя придумайте самостоятельно).

Для оформления формы понадобятся следующие объекты:

- Label (метка)
- Textbox (Текстовое поле)
- Button (Кнопка)

Поместите эти объекты на форму как показано на рисунке.



Далее, создадим **обработчик события** (далее просто событие) при нажатии на кнопку Button1. Для этого щелкните двойным щелчком по выделенной кнопке. На экране появится редактор кода.

```

namespace проект2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

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

```

Прежде чем начать программировать, сохраните создаваемый проект в файлах проекта (Файл – Сохранить все).

Вернёмся к программированию и рассмотрим содержимое текста в редакторе кода.

Директива **using** подключает модули, необходимые для поддержки работоспособности нашей программы.

Namespace (пространство имён проекта) и открывающаяся фигурная скобка. Открывающиеся и закрывающиеся фигурные скобки задают начало и конец какого-либо процесса.

За скобкой следует строка создания класса с именем Form1 наследника класса Form:

public partial class Form1 : Form

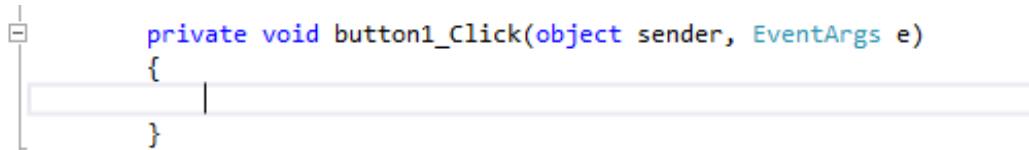
Класс **Form** описан в одном из подключенных модулей и у нас есть доступ только к его свойствам и событиям.

Form1, это наша форма, на которой располагаются визуальные объекты программы.

В строке **public Form1()** объявляется функция инициализации формы с кодом **InitializeComponent();**.

Эти строки создаются автоматически при создании проекта.

А вот строки:



создали мы сами при двойном щелчке на выделенной кнопке Button1; это и есть функция обработчика события.

Допишем событие, которое должно произойти при нажатии на кнопку:

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = textBox1.Text;
}
```

Рассмотрим строку `label1.Text = textBox1.Text;`

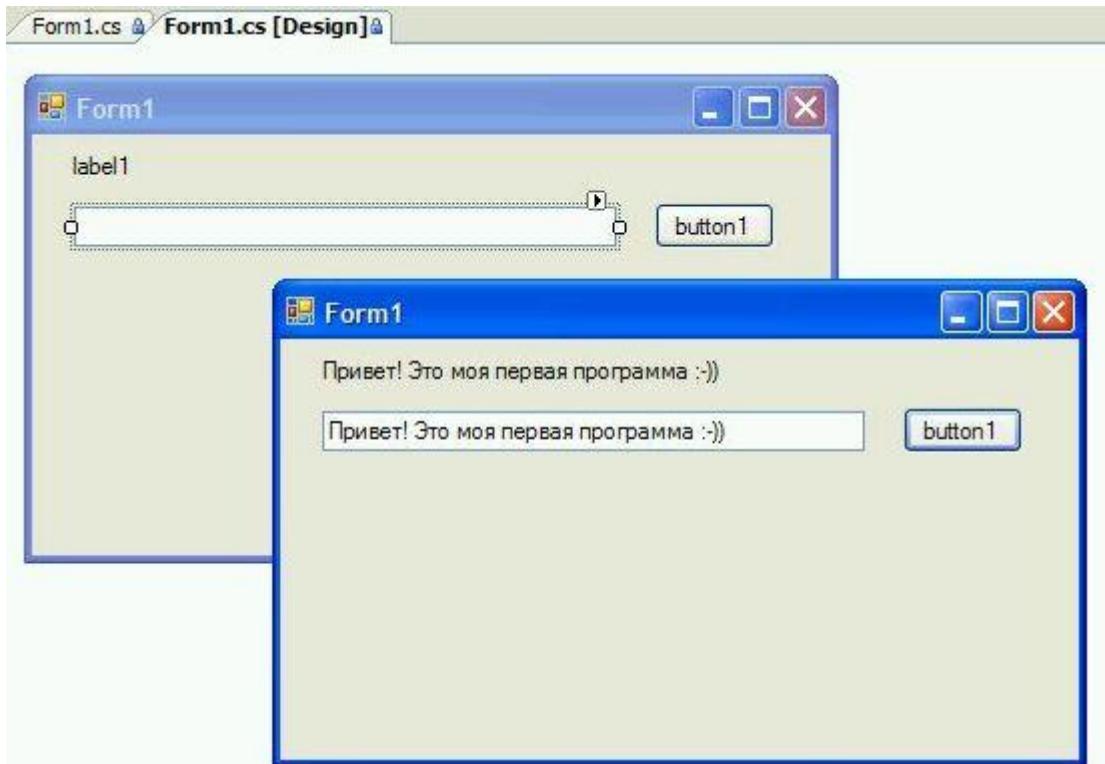
Нам необходимо, чтобы при нажатии на кнопку, текст из текстового поля `textBox1` был скопирован на этикетку `label1`, то есть текст этикетки должен быть равен тексту текстового поля, что мы и записываем в программной строке.

В любом объектно-ориентированном языке программирования принято указывать свойство объекта через точку после имени объекта.

имя объекта - точка - свойство объекта

Ещё один немаловажный аспект - имена объектов, свойств, переменных и тому подобное, пишутся именно в том регистре, в каком они зарезервированы в данном языке программирования.

Запустите программу и проверьте, как она работает.



Закроем запущенную программу и создадим ещё одно событие, которое будет "облагораживать" форму при старте программы. Для этого переключимся на страничку дизайна формы и двойным щелчком мыши на самой форме создадим событие:

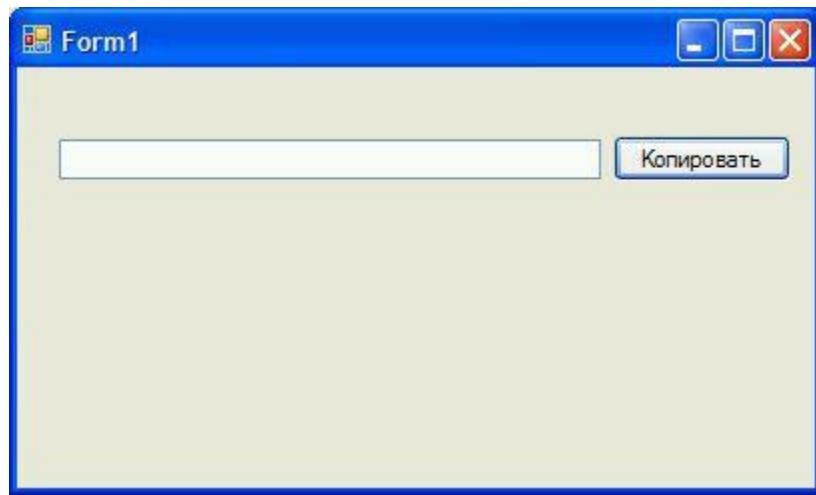
```
private void Form1_Load(object sender, EventArgs e)
{
}
```

Это событие (функция) Form1_Load означает что произойдут какие то изменения с дизайном формы во время запуска программы на выполнение. В этой функции, уберём с метки label1 надпись которая может ввести пользователя нашей программы в заблуждение, а на кнопку button1 выведем надпись, например "Копировать". Наша функция - обработчик примет вид:

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = "";
    button1.Text = "Копировать";
}
```

Обратите внимание, что в конце строки каждого выражения ставится точка с запятой.

Запустите приложение и убедитесь, что форма приняла нормальный вид:



Совет. Для пояснения работы тех или иных участков программы, рекомендуется снабжать код программы комментариями. Комментарии записываются через две наклонные черты и пишутся до конца строки. Например:

```

Form1.cs* Form1.cs [Design]*

Lesson1.Form1
Form1_Load(object sender, EventArgs e)

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    } // Конец инициализации класса
    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = textBox1.Text;
    } // Конец обработчика события нажатия на кнопку
    private void Form1_Load(object sender, EventArgs e)
    {
        label1.Text = "";
        button1.Text = "Копировать";
    }
} // Конец создания класса
} // End of namespace

```

Комментарии никак не влияют на работу программы и могут быть объявлены в любом месте текста кода.