Раздел 1. Программирование в визуальной среде с применением объектно-ориентированных технологий

Программирование в Microsoft Visual Studio 2012 (2013, 2015) в среде **Microsoft Visual C++/ Microsoft Visual C#**

С появлением **технологии .NET и платформы Microsoft .NET Framework for Windows** в **Visual C++ /C#** интегрированы возможности разработки **.NET – приложений.**

Основная идея **Технологии .NET** - позволяет создавать универсальный программный код, работающий в любой операционной системе.

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году, обеспечивающая поддержку **технологии .NET в ОС Microsoft Windows .**

Основа платформы (компонент .NET Framework) - общеязыковая среда исполнения Common Language Runtime (CLR), которая обеспечивает выполнение **.NET – приложений в ОС Microsoft Windows . CLR** подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

https://www.youtube.com/watch?v=a7XPMLbnpU0

Среда визуального программирования – среда быстрой разработки Приложений,
RAD – rapid application development

Основы RAD – технология визуального проектирования (проектирование интерфейса) и событийного программирования (написание функций обработки событий),
Применяются ОО-технологии.

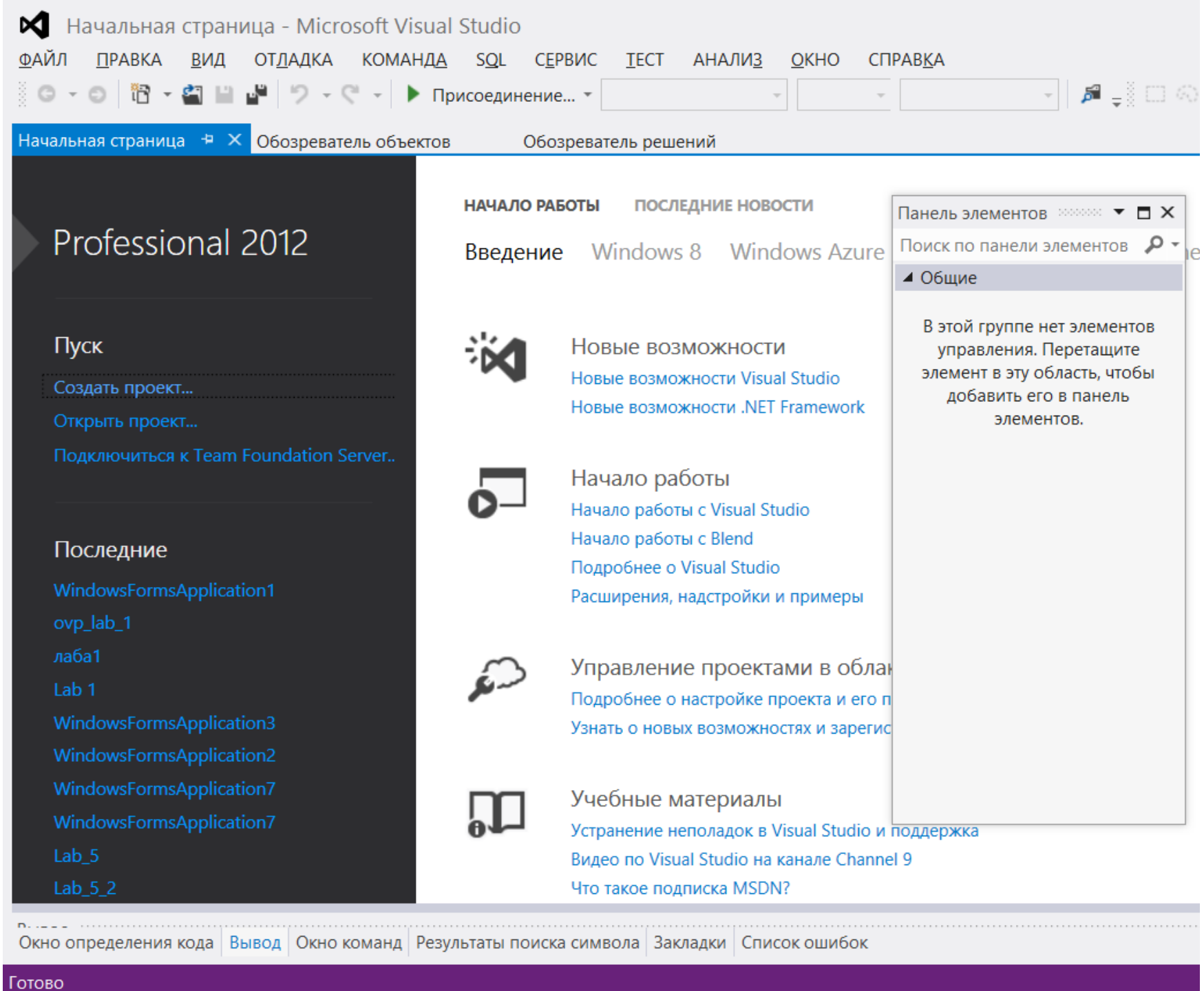
Этапы разработки приложений:

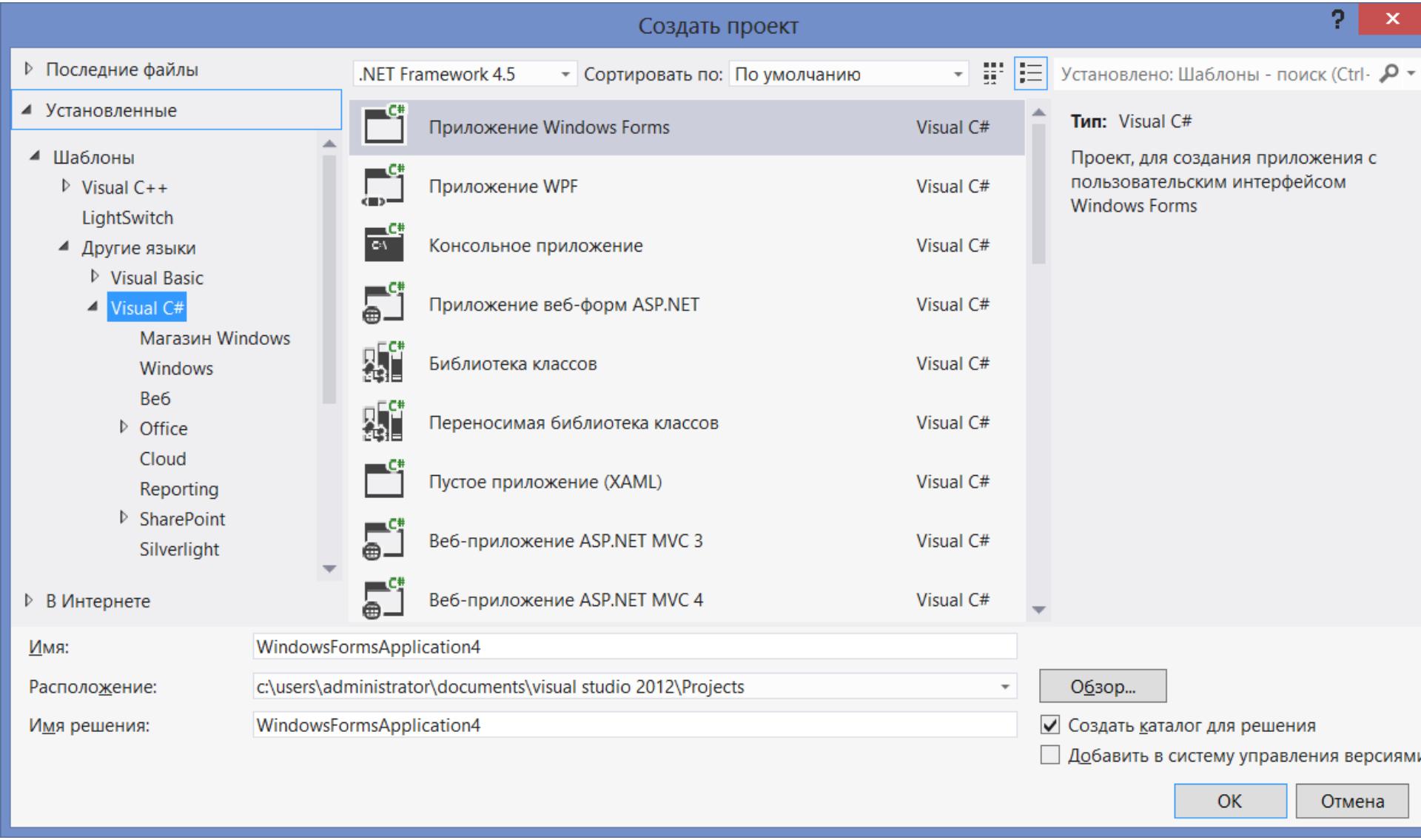
Разработка Интерфейса

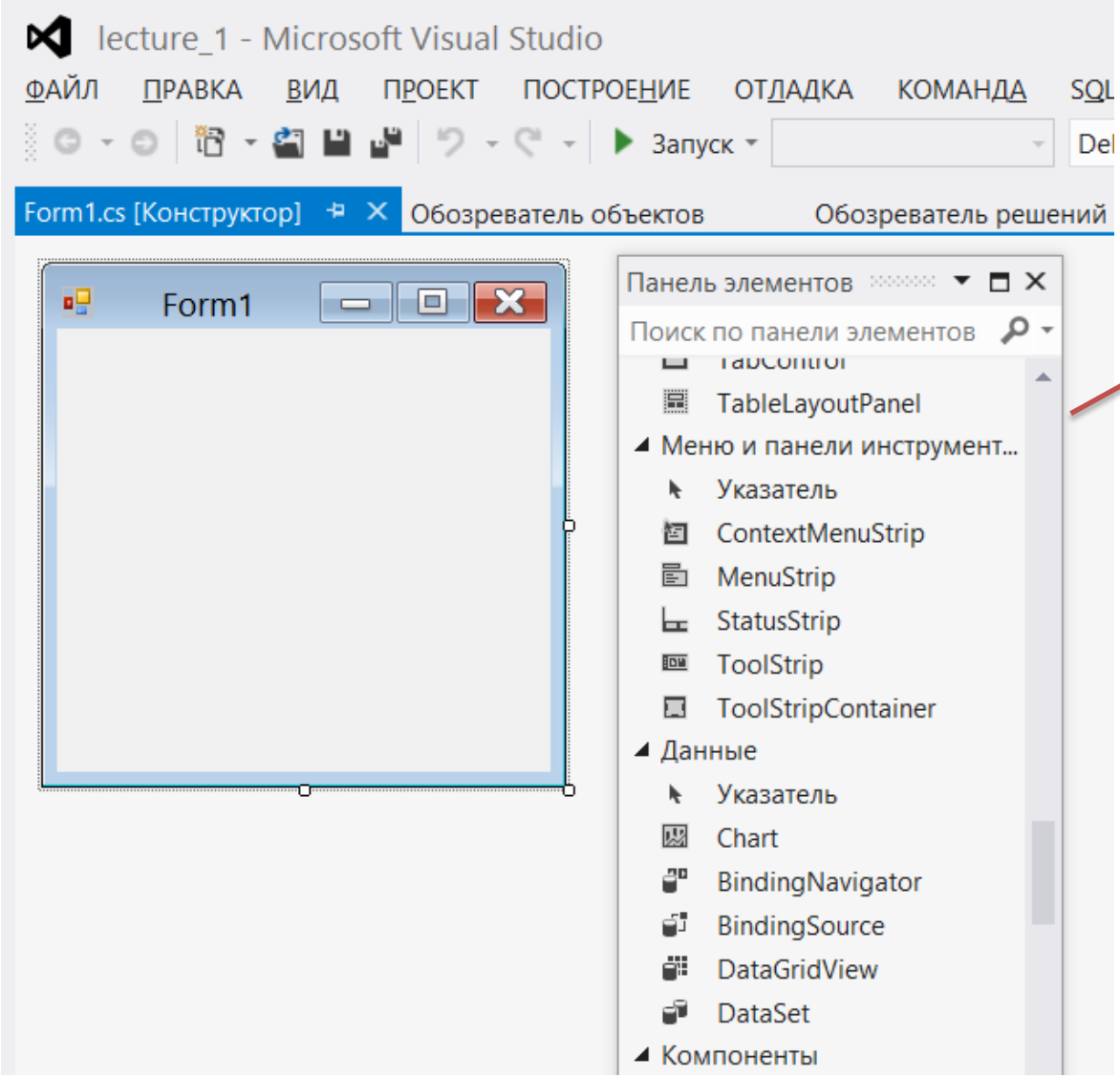
Написание кода

Основные составляющие среды **Microsoft Visual C++/C#**:

- конструктор форм
- специализированный редактор кода
- высокоскоростной оптимизирующий компилятор


Начальная страница - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ОТЛАДКА КОМАНДА SQL СЕРВИС ТЕСТ АНАЛИЗ ОКНО СПРАВКА
Присоединение...
Начальная страница
Обозреватель объектов
Обозреватель решений
Professional 2012
Пуск
Создать проект...
Открыть проект...
Подключиться к Team Foundation Server..
Последние
WindowsFormsApplication1
ovp_lab_1
лаба1
Lab 1
WindowsFormsApplication3
WindowsFormsApplication2
WindowsFormsApplication7
WindowsFormsApplication7
Lab_5
Lab_5_2
НАЧАЛО РАБОТЫ
ПОСЛЕДНИЕ НОВОСТИ
Введение
Windows 8
Windows Azure
Новые возможности
Новые возможности Visual Studio
Новые возможности .NET Framework
Начало работы
Начало работы с Visual Studio
Начало работы с Blend
Подробнее о Visual Studio
Расширения, надстройки и примеры
Управление проектами в облак
Подробнее о настройке проекта и его п
Узнать о новых возможностях и зарегис
Учебные материалы
Устранение неполадок в Visual Studio и поддержка
Видео по Visual Studio на канале Channel 9
Что такое подписка MSDN?
Панель элементов
Поиск по панели элементов
Общие
В этой группе нет элементов управления. Перетащите элемент в эту область, чтобы добавить его в панель элементов.
Окно определения кода
Вывод
Окно команд
Результаты поиска символа
Закладки
Список ошибок
Готово


Создать проект
Последние файлы
Установленные
Шаблоны
Visual C++
LightSwitch
Другие языки
Visual Basic
Visual C#
Магазин Windows
Windows
Веб
Office
Cloud
Reporting
SharePoint
Silverlight
В Интернете
.NET Framework 4.5
Сортировать по:
По умолчанию
Установлено: Шаблоны - поиск (Ctrl-
Приложение Windows Forms
Visual C#
Приложение WPF
Visual C#
Консольное приложение
Visual C#
Приложение веб-форм ASP.NET
Visual C#
Библиотека классов
Visual C#
Переносимая библиотека классов
Visual C#
Пустое приложение (XAML)
Visual C#
Веб-приложение ASP.NET MVC 3
Visual C#
Веб-приложение ASP.NET MVC 4
Visual C#
Тип: Visual C#
Проект, для создания приложения с пользовательским интерфейсом Windows Forms
Имя:
WindowsFormsApplication4
Расположение:
c:\users\administrator\documents\visual studio 2012\Projects
Обзор...
Имя решения:
WindowsFormsApplication4
Создать каталог для решения
Добавить в систему управления версиями
OK
Отмена

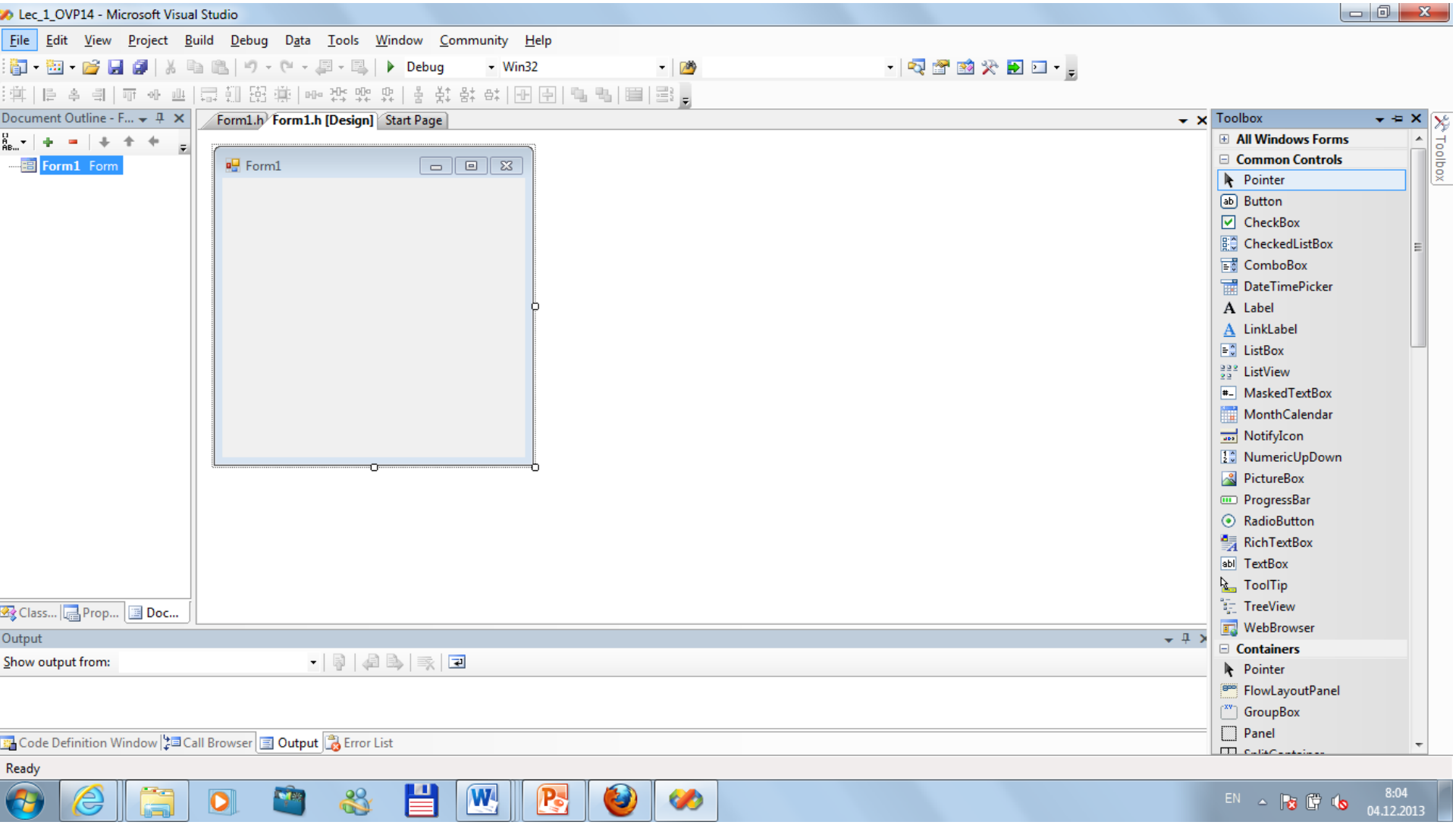


Если нет окна: пункт меню ВИД – Панель элементов

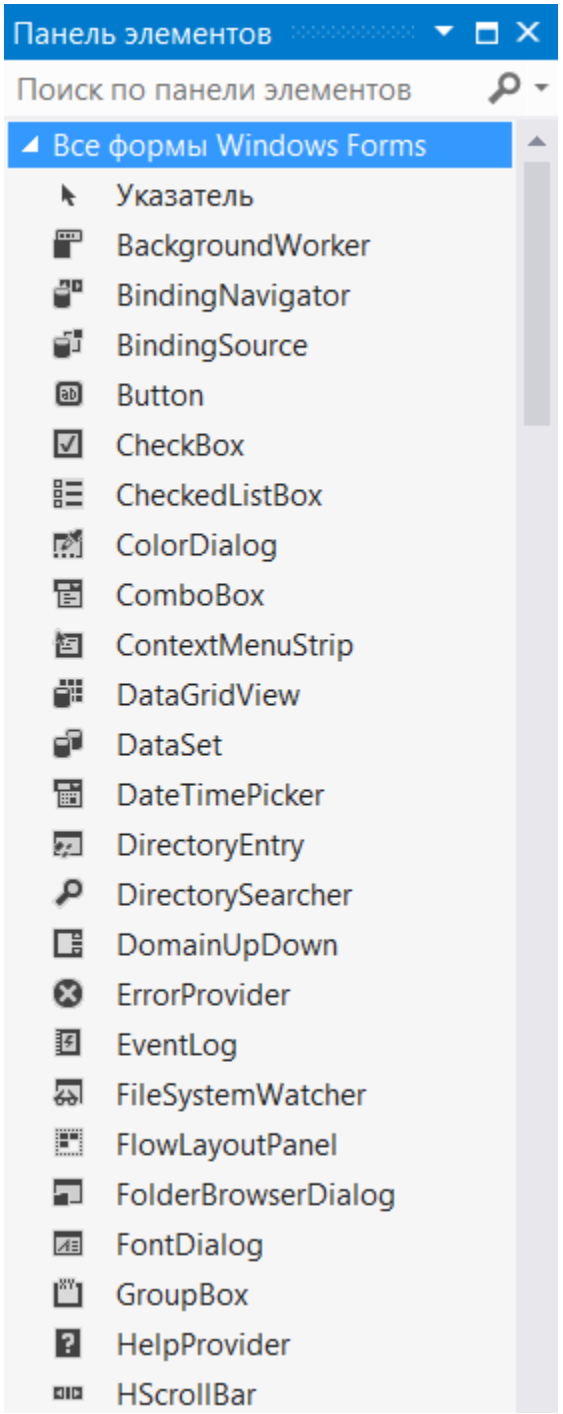
Проект в визуальных средах - совокупность файлов, необходимых для создания программы (**Приложение**)

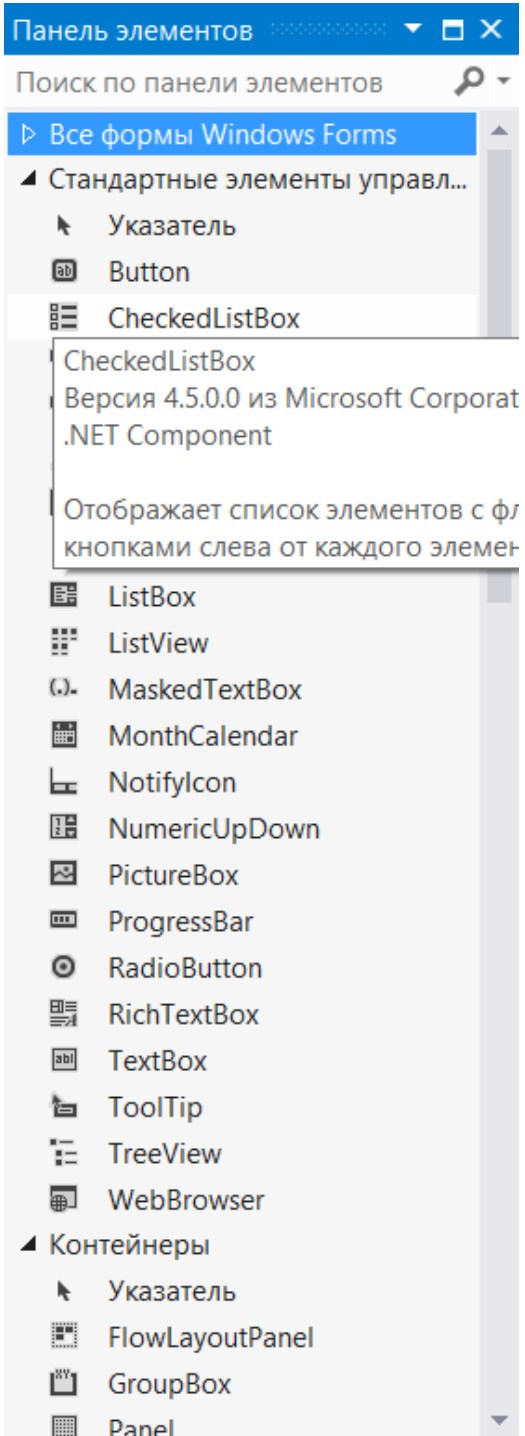
В окне конструктора (Designer) формы - расположена *Форма* - заготовка окна Приложения.

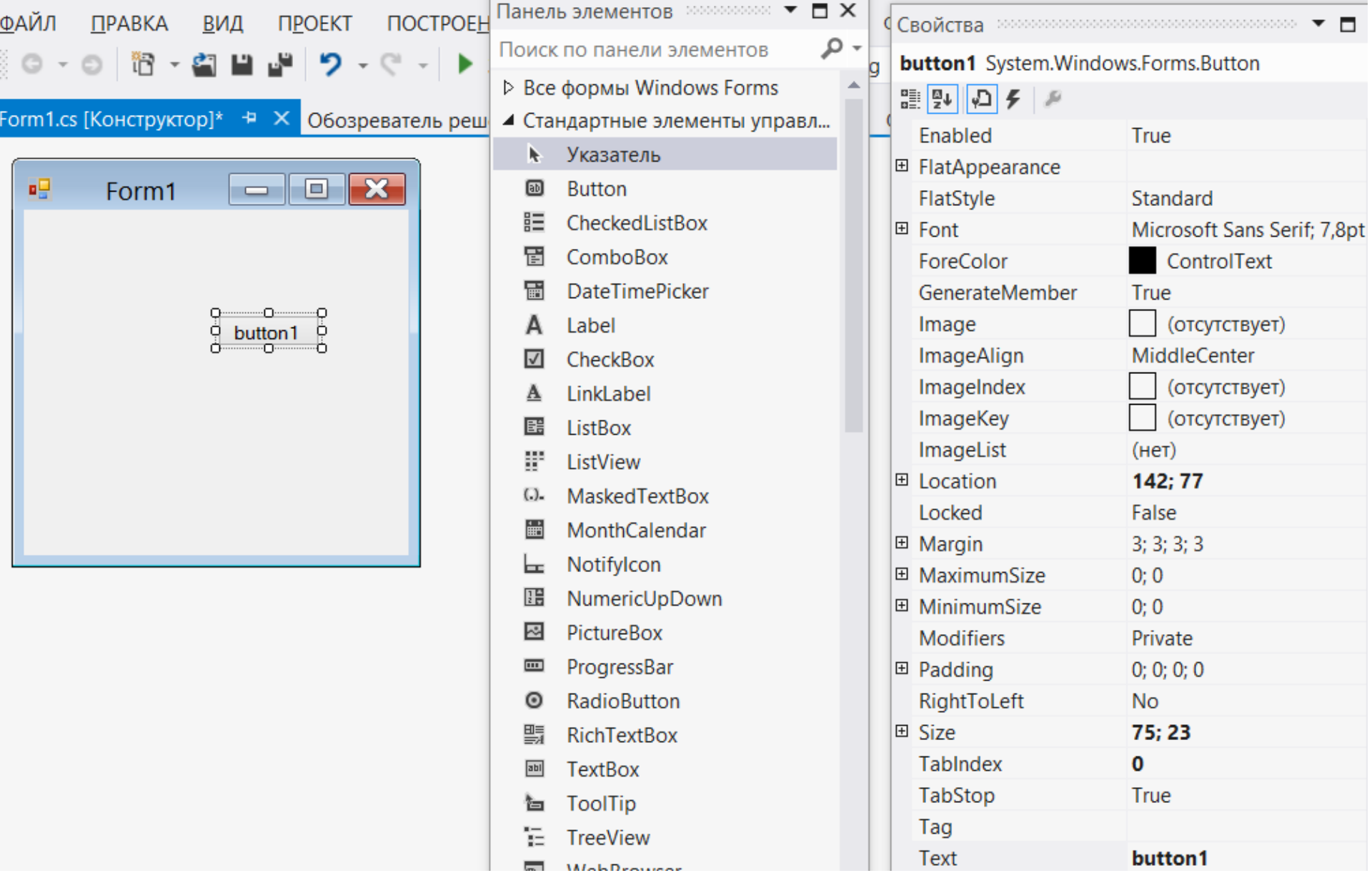
Работа над Приложением начинается с создания стартовой формы – главного окна программы

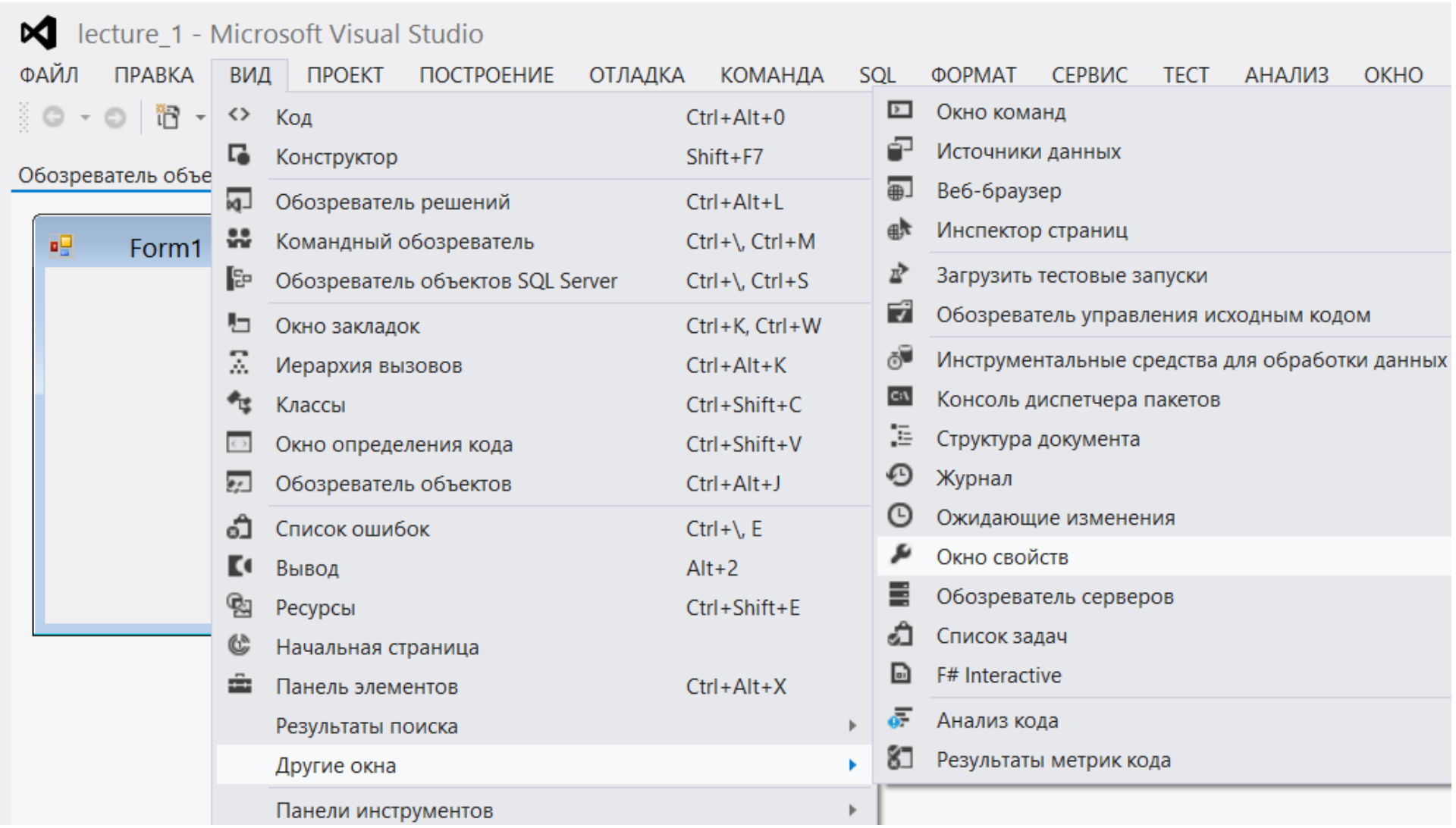


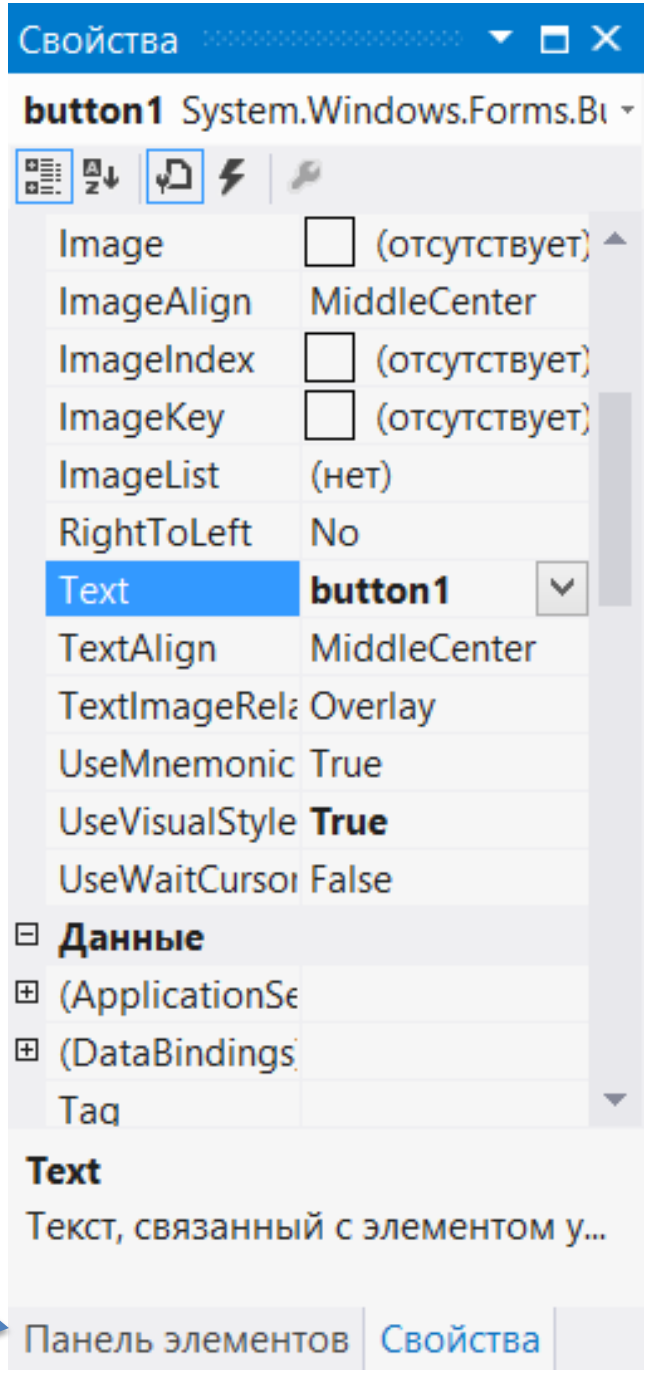
Окно ToolBox (Панель элементов) - содержит компоненты, которые можно поместить на форму

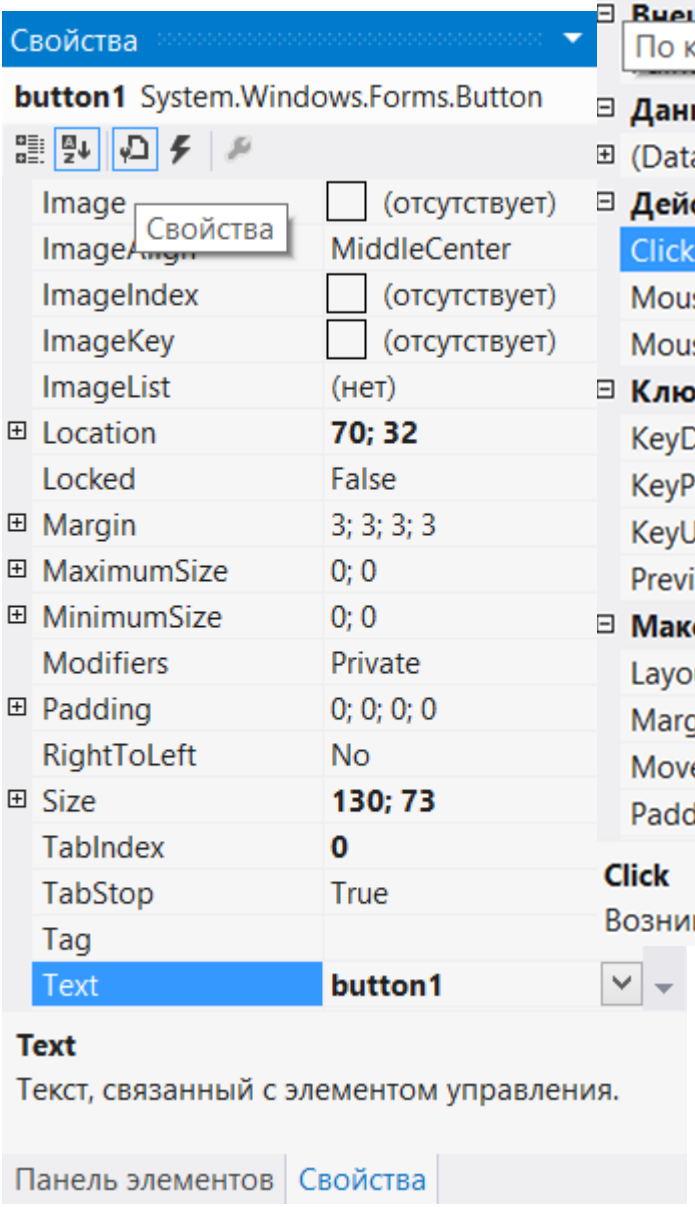

Панель элементов
Поиск по панели элементов
Все формы Windows Forms
Указатель
BackgroundWorker
BindingNavigator
BindingSource
Button
CheckBox
CheckedListBox
ColorDialog
ComboBox
ContextMenuStrip
DataGridView
DataSet
DateTimePicker
DirectoryEntry
DirectorySearcher
DomainUpDown
ErrorProvider
EventLog
FileSystemWatcher
FlowLayoutPanel
FolderBrowserDialog
FontDialog
GroupBox
HelpProvider
HScrollBar

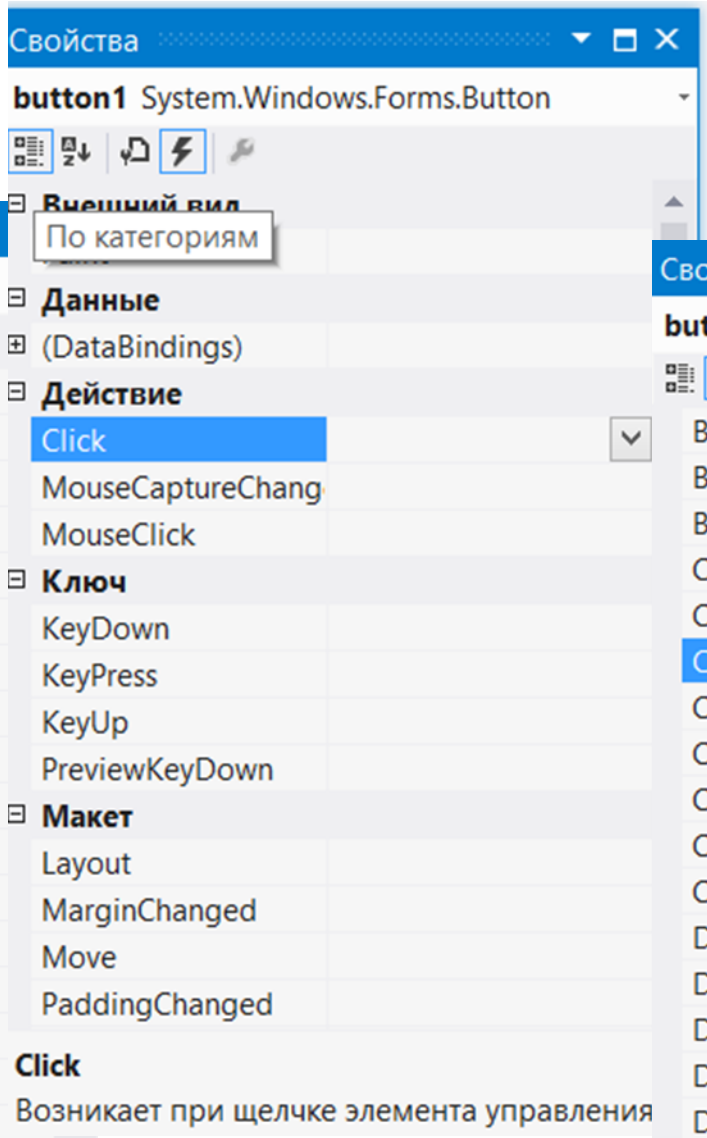

Панель элементов
Поиск по панели элементов
Все формы Windows Forms
Стандартные элементы управл...
Указатель
Button
CheckedListBox
CheckedListBox
Версия 4.5.0.0 из Microsoft Corporat
.NET Component
Отображает список элементов с фл
кнопками слева от каждого элемен
ListBox
ListView
MaskedTextBox
MonthCalendar
NotifyIcon
NumericUpDown
PictureBox
ProgressBar
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
WebBrowser
Контейнеры
Указатель
FlowLayoutPanel
GroupBox
Panel

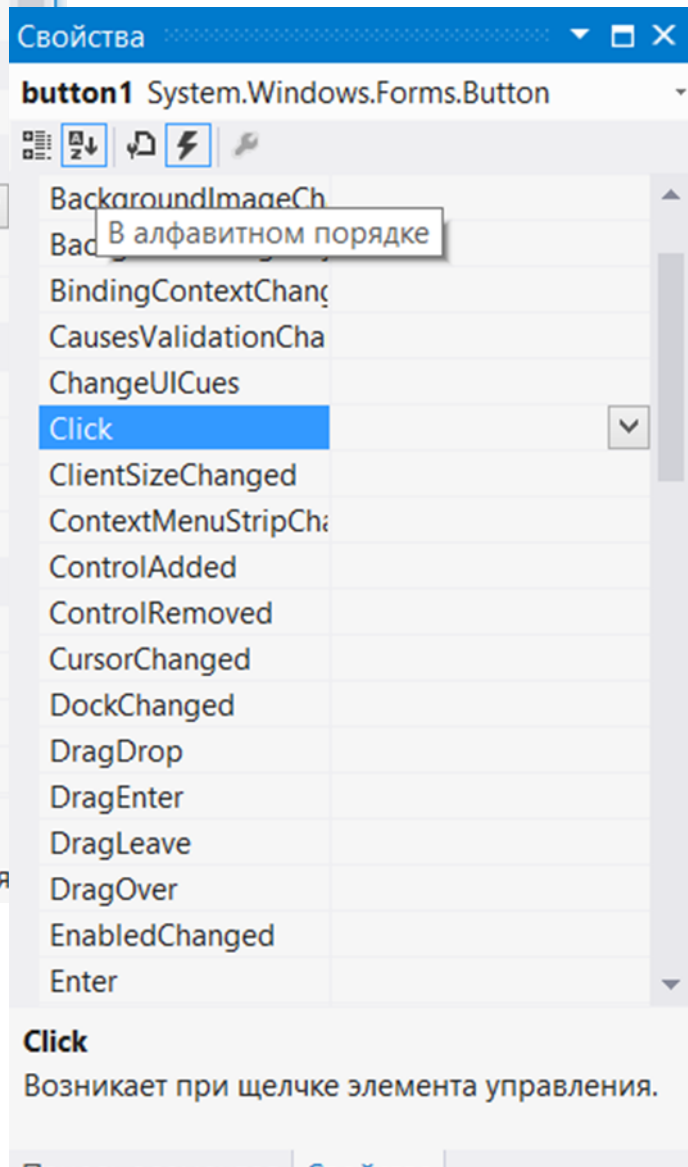

ФАЙЛ
ПРАВКА
ВИД
ПРОЕКТ
ПОСТРОЕН
Form1.cs [Конструктор]*
Обозреватель реш
Form1
button1
Панель элементов
Поиск по панели элементов
Все формы Windows Forms
Стандартные элементы управл...
Указатель
Button
CheckedListBox
ComboBox
DateTimePicker
Label
CheckBox
LinkLabel
ListBox
ListView
MaskedTextBox
MonthCalendar
NotifyIcon
NumericUpDown
PictureBox
ProgressBar
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
Свойства
button1 System.Windows.Forms.Button
Enabled True
FlatAppearance
FlatStyle Standard
Font Microsoft Sans Serif; 7,8pt
ForeColor ControlText
GenerateMember True
Image (отсутствует)
ImageAlign MiddleCenter
ImageIndex (отсутствует)
ImageKey (отсутствует)
ImageList (нет)
Location 142; 77
Locked False
Margin 3; 3; 3; 3
MaximumSize 0; 0
MinimumSize 0; 0
Modifiers Private
Padding 0; 0; 0; 0
RightToLeft No
Size 75; 23
TabIndex 0
TabStop True
Tag
Text button1


lecture_1 - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА SQL ФОРМАТ СЕРВИС ТЕСТ АНАЛИЗ ОКНО
Обозреватель объе
Form1
Код Ctrl+Alt+0
Конструктор Shift+F7
Обозреватель решений Ctrl+Alt+L
Командный обозреватель Ctrl+\, Ctrl+M
Обозреватель объектов SQL Server Ctrl+\, Ctrl+S
Окно закладок Ctrl+K, Ctrl+W
Иерархия вызовов Ctrl+Alt+K
Классы Ctrl+Shift+C
Окно определения кода Ctrl+Shift+V
Обозреватель объектов Ctrl+Alt+J
Список ошибок Ctrl+\, E
Вывод Alt+2
Ресурсы Ctrl+Shift+E
Начальная страница
Панель элементов Ctrl+Alt+X
Результаты поиска
Другие окна
Панели инструментов
Окно команд
Источники данных
Веб-браузер
Инспектор страниц
Загрузить тестовые запуски
Обозреватель управления исходным кодом
Инструментальные средства для обработки данных
Консоль диспетчера пакетов
Структура документа
Журнал
Ожидающие изменения
Окно свойств
Обозреватель серверов
Список задач
F# Interactive
Анализ кода
Результаты метрик кода

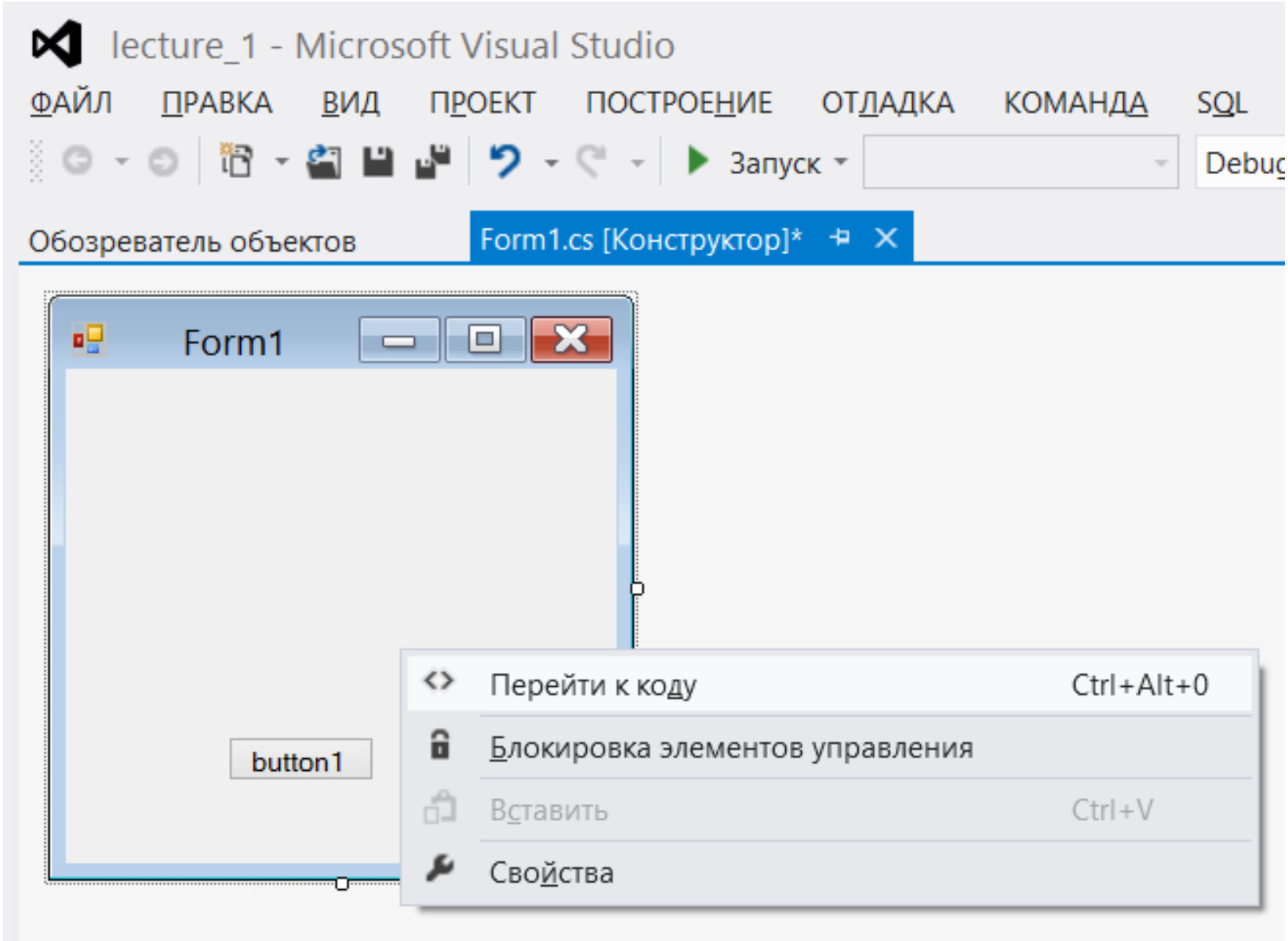

Свойства
button1 System.Windows.Forms.Bu
Image (отсутствует)
ImageAlign MiddleCenter
ImageIndex (отсутствует)
ImageKey (отсутствует)
ImageList (нет)
RightToLeft No
Text button1
TextAlign MiddleCenter
TextImageRela Overlay
UseMnemonic True
UseVisualStyle True
UseWaitCurso False
Данные
(ApplicationSe
(DataBindings
Tag
Text
Текст, связанный с элементом у...
Панель элементов
Свойства

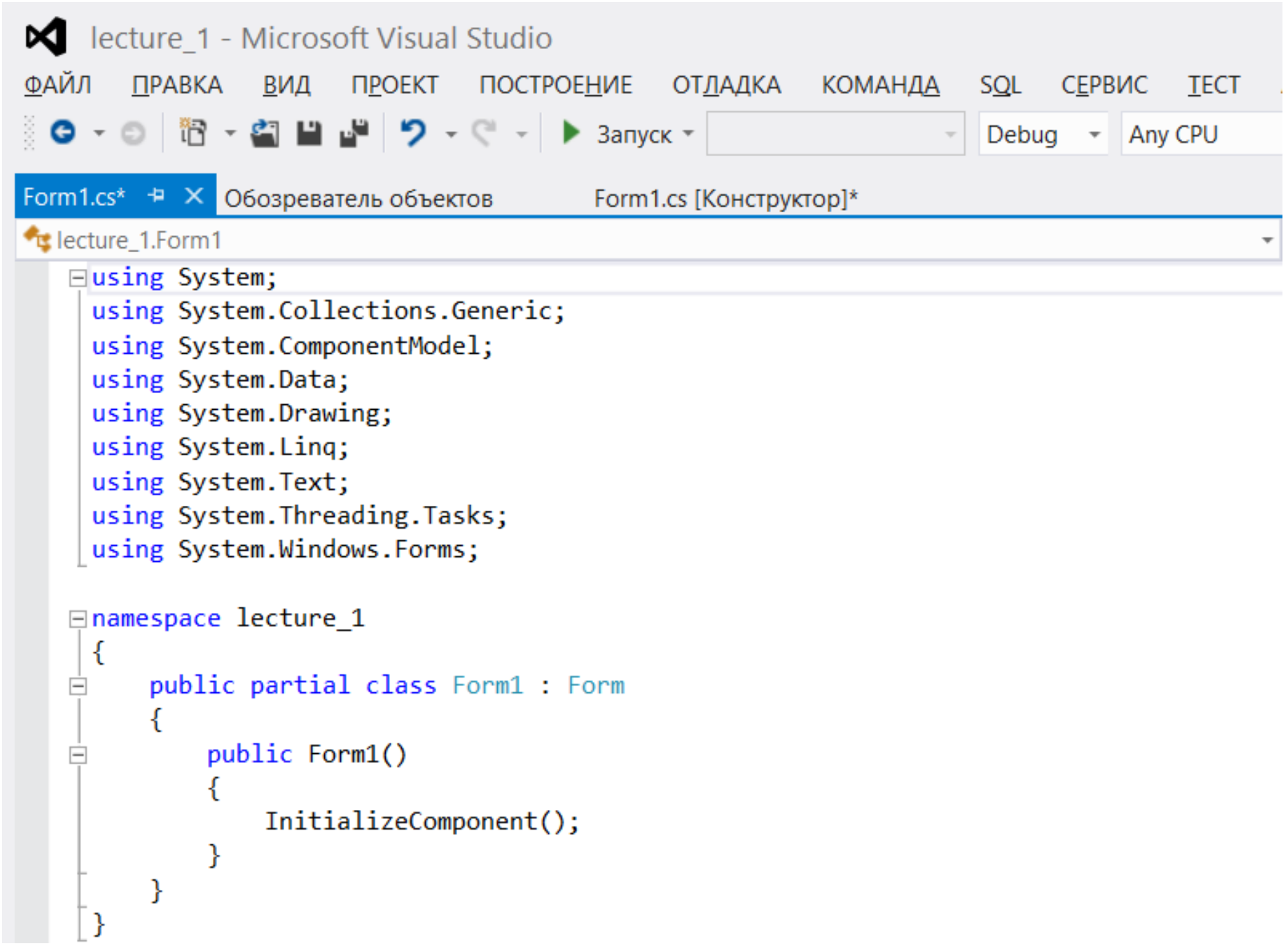

Свойства
button1 System.Windows.Forms.Button
Свойства
Image (отсутствует)
MiddleCenter
ImageIndex (отсутствует)
ImageKey (отсутствует)
ImageList (нет)
Location 70; 32
Locked False
Margin 3; 3; 3; 3
MaximumSize 0; 0
MinimumSize 0; 0
Modifiers Private
Padding 0; 0; 0; 0
RightToLeft No
Size 130; 73
TabIndex 0
TabStop True
Tag
Text button1
Text
Текст, связанный с элементом управления.
Панель элементов
Свойства

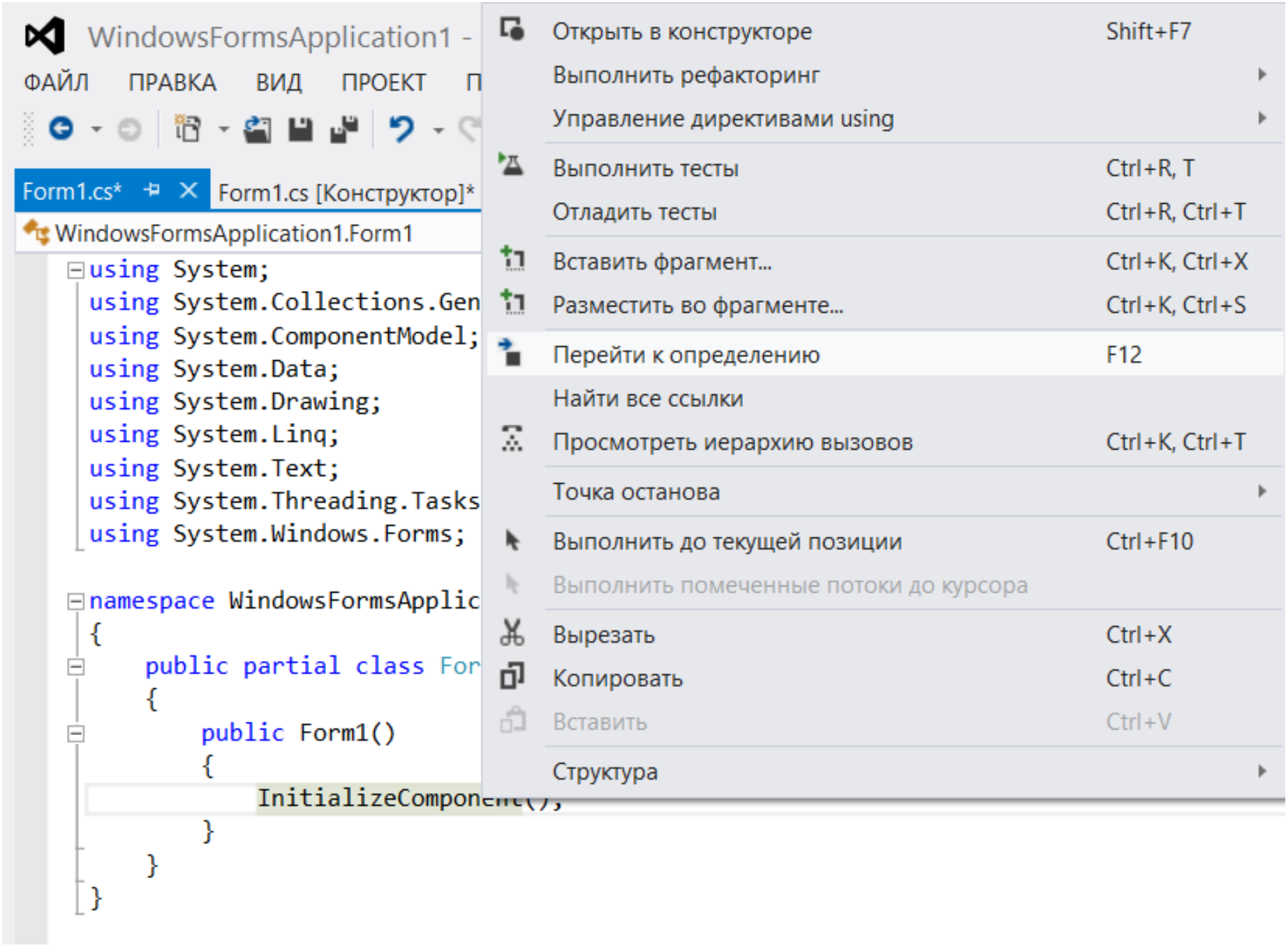

Свойства
button1 System.Windows.Forms.Button
По категориям
Данные
(DataBindings)
Действие
Click
MouseCaptureChang
MouseClick
Ключ
KeyDown
KeyPress
KeyUp
PreviewKeyDown
Макет
Layout
MarginChanged
Move
PaddingChanged
Click
Возникает при щелчке элемента управления


Свойства
button1 System.Windows.Forms.Button
В алфавитном порядке
BindingContextChanc
CausesValidationCha
ChangeUICues
Click
ClientSizeChanged
ContextMenuStripCha
ControlAdded
ControlRemoved
CursorChanged
DockChanged
DragDrop
DragEnter
DragLeave
DragOver
EnabledChanged
Enter
Click
Возникает при щелчке элемента управления.
Панель элементов
Свойства

В окне конструктора (Designer) формы можно посмотреть код создания формы:
Контекстное меню на форме – **Перейти к коду**
Обратно к форме – **открыть в конструкторе**

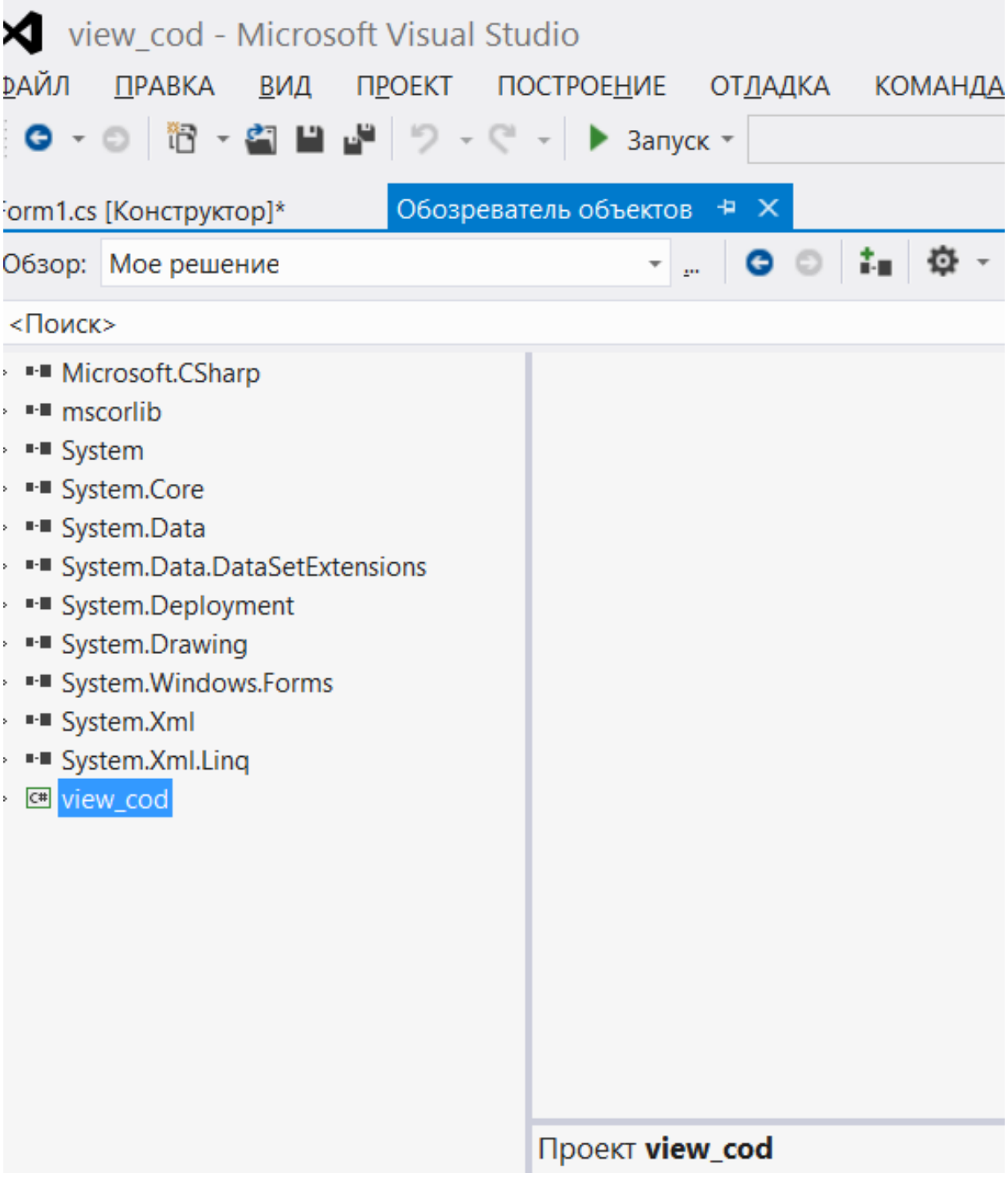


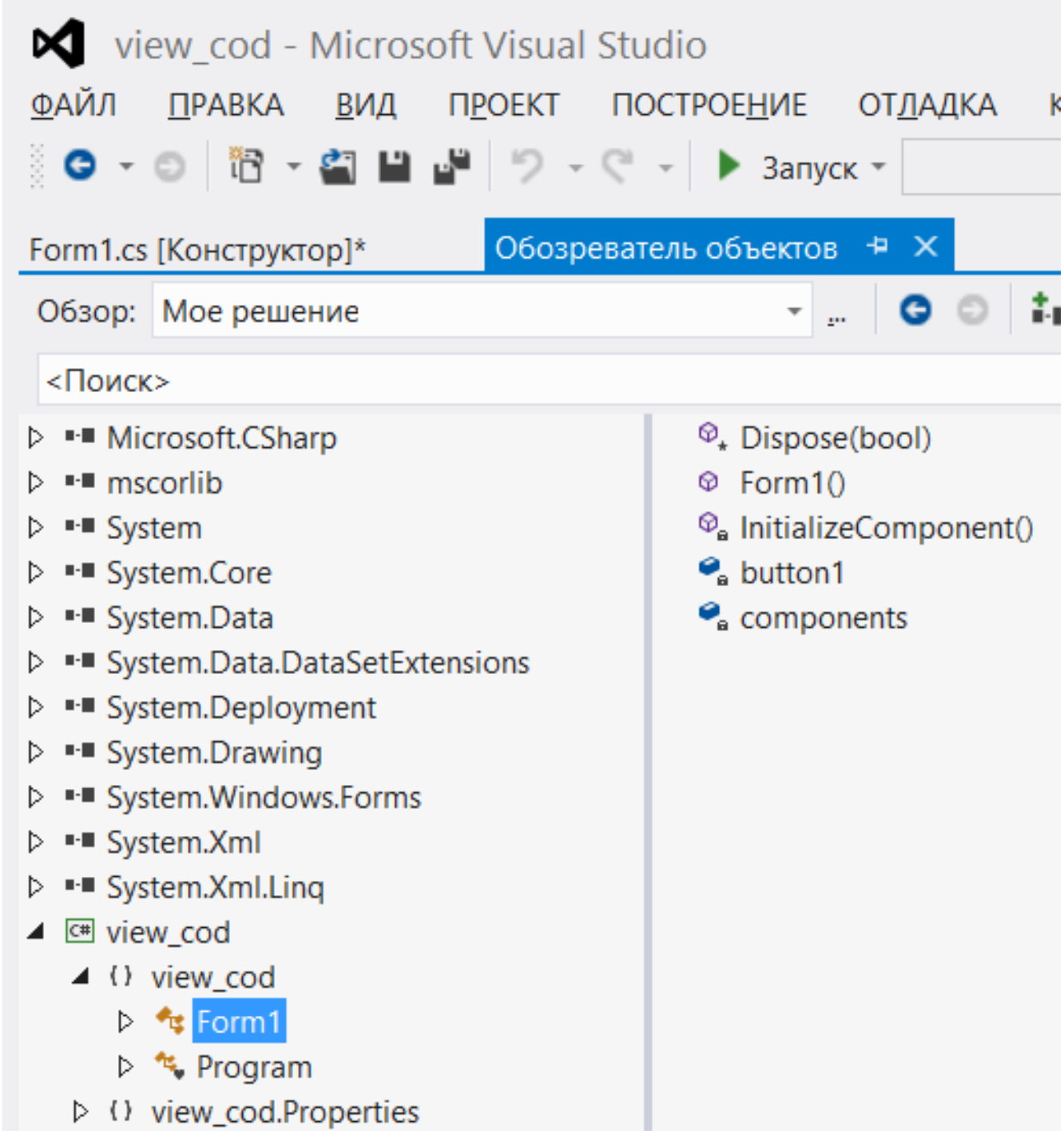

lecture_1 - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА SQL СЕРВИС ТЕСТ
Запуск
Debug
Any CPU
Form1.cs*
Обозреватель объектов
Form1.cs [Конструктор]*
lecture_1.Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace lecture_1
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
}
}

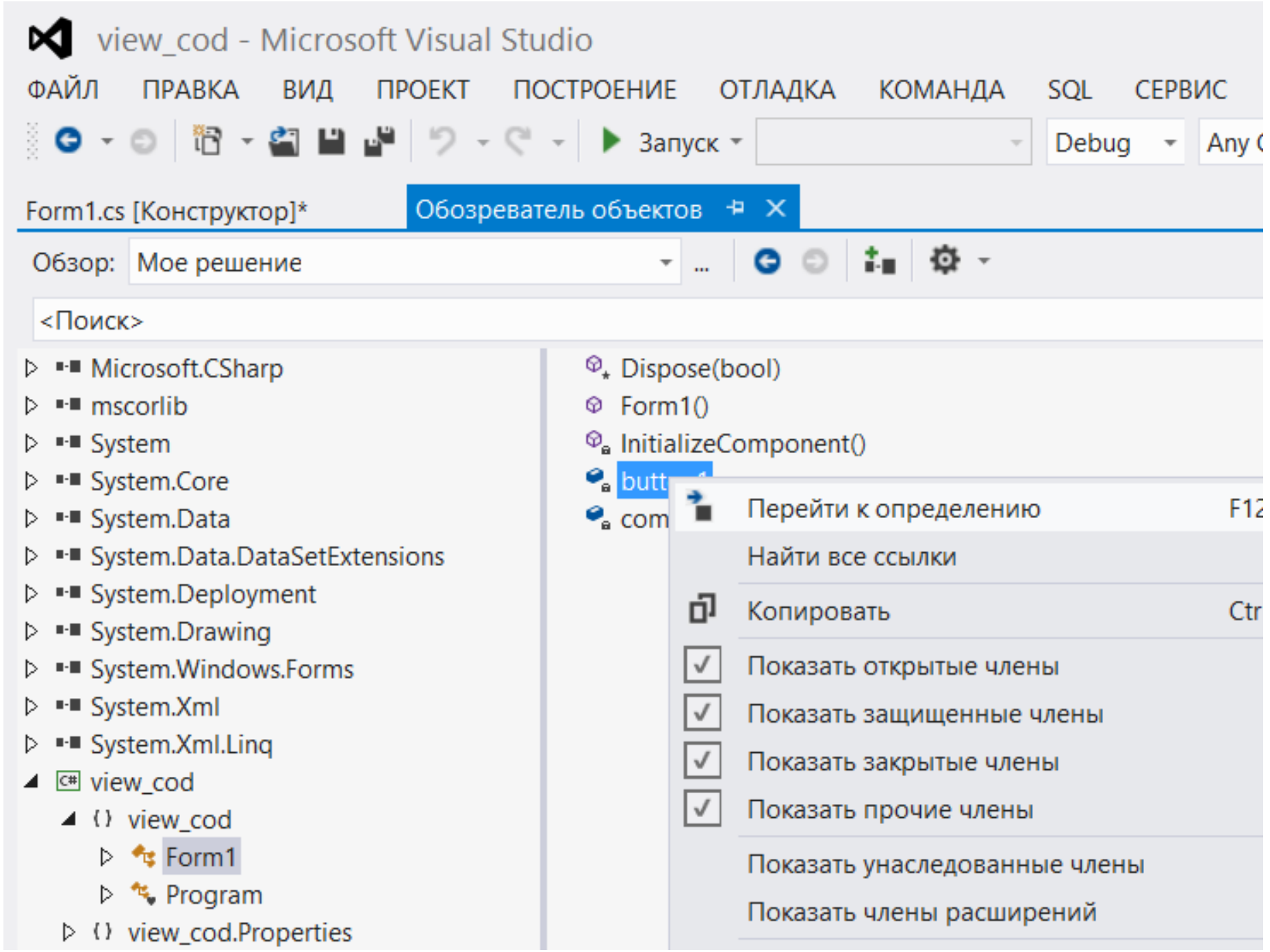

WindowsFormsApplication1 -
ФАЙЛ ПРАВКА ВИД ПРОЕКТ П
Form1.cs*
Form1.cs [Конструктор]*
WindowsFormsApplication1.Form1
using System;
using System.Collections.Gen
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks
using System.Windows.Forms;
namespace WindowsFormsApplic
{
public partial class For
{
public Form1()
{
InitializeComponent();
}
}
}
Открыть в конструкторе Shift+F7
Выполнить рефакторинг
Управление директивами using
Выполнить тесты Ctrl+R, T
Отладить тесты Ctrl+R, Ctrl+T
Вставить фрагмент... Ctrl+K, Ctrl+X
Разместить во фрагменте... Ctrl+K, Ctrl+S
Перейти к определению F12
Найти все ссылки
Просмотреть иерархию вызовов Ctrl+K, Ctrl+T
Точка останова
Выполнить до текущей позиции Ctrl+F10
Выполнить помеченные потоки до курсора
Вырезать Ctrl+X
Копировать Ctrl+C
Вставить Ctrl+V
Структура

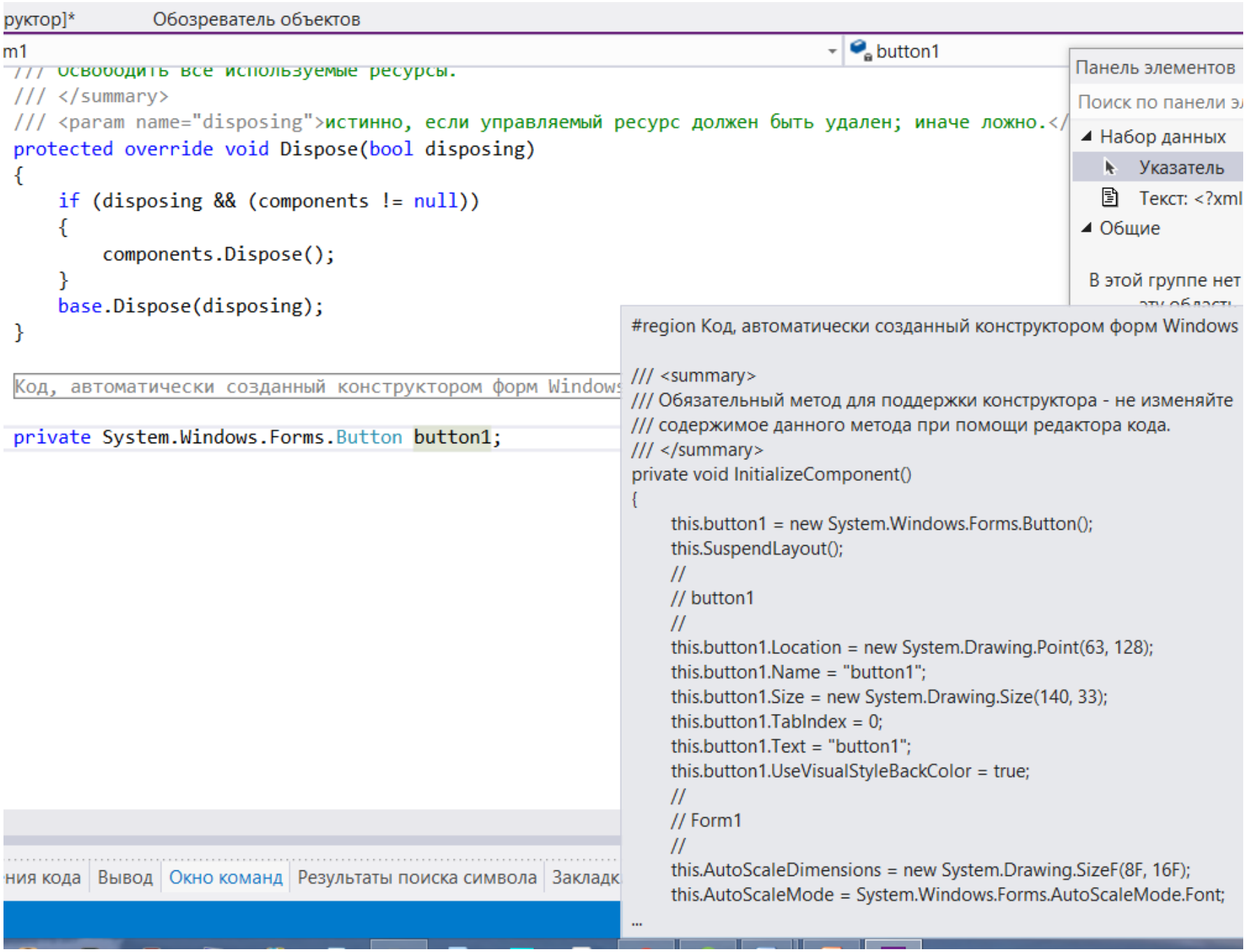
WindowsFormsApplication1.Form1

```
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Обязательный метод для поддержки конструктора - не изменяйте
        /// содержимое данного метода при помощи редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(142, 77);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75, 23);
            this.button1.TabIndex = 0;
            this.button1.Text = "button1";
            this.button1.UseVisualStyleBackColor = true;
            //
            // Form1
            //
```


view_cod - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА
Запуск
orm1.cs [Конструктор]*
Обозреватель объектов
Обзор: Мое решение
<Поиск>
Microsoft.CSharp
mscorlib
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Deployment
System.Drawing
System.Windows.Forms
System.Xml
System.Xml.Linq
view_cod
Проект view_cod


view_cod - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА
Запуск
Form1.cs [Конструктор]*
Обозреватель объектов
Обзор: Мое решение
<Поиск>
Microsoft.CSharp
mscorlib
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Deployment
System.Drawing
System.Windows.Forms
System.Xml
System.Xml.Linq
view_cod
view_cod
Form1
Program
view_cod.Properties
Dispose(bool)
Form1()
InitializeComponent()
button1
components

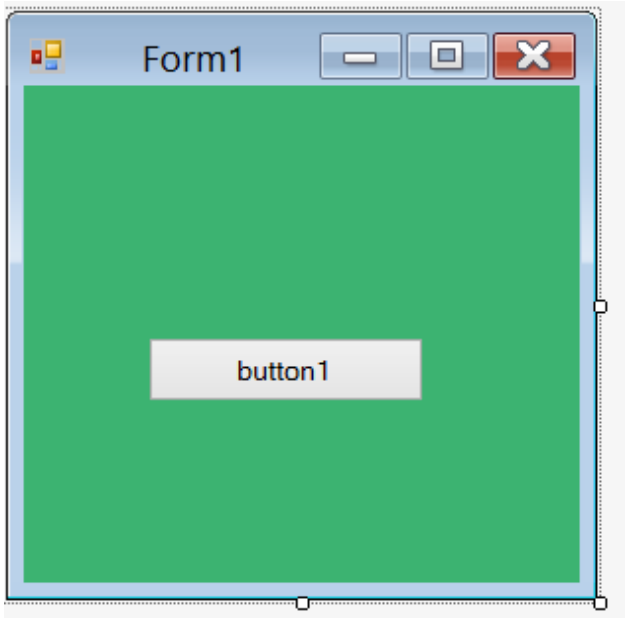

view_cod - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА SQL СЕРВИС
Запуск
Debug
Form1.cs [Конструктор]*
Обозреватель объектов
Обзор:
Мое решение
<Поиск>
Microsoft.CSharp
mscorlib
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Deployment
System.Drawing
System.Windows.Forms
System.Xml
System.Xml.Linq
view_cod
view_cod
Form1
Program
view_cod.Properties
Dispose(bool)
Form1()
InitializeComponent()
Перейти к определению
Найти все ссылки
Копировать
Показать открытые члены
Показать защищенные члены
Показать закрытые члены
Показать прочие члены
Показать унаследованные члены
Показать члены расширений


руктор]*
Обозреватель объектов
m1
button1
/// Освободить все используемые ресурсы.
/// </summary>
/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</
protected override void Dispose(bool disposing)
{
if (disposing && (components != null))
{
components.Dispose();
}
base.Dispose(disposing);
}
Код, автоматически созданный конструктором форм Windows
private System.Windows.Forms.Button button1;
Панель элементов
Поиск по панели эл
Набор данных
Указатель
Текст: <?xml
Общие
В этой группе нет
#region Код, автоматически созданный конструктором форм Windows
/// <summary>
/// Обязательный метод для поддержки конструктора - не изменяйте
/// содержимое данного метода при помощи редактора кода.
/// </summary>
private void InitializeComponent()
{
this.button1 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// button1
//
this.button1.Location = new System.Drawing.Point(63, 128);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(140, 33);
this.button1.TabIndex = 0;
this.button1.Text = "button1";
this.button1.UseVisualStyleBackColor = true;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
...
ния кода
Вывод
Окно команд
Результаты поиска символа
Закладк

```
private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(63, 128);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(140, 33);
            this.button1.TabIndex = 0;
            this.button1.Text = "button1";
            this.button1.UseVisualStyleBackColor = true;
            //
```

```
            // Form1                    //
            this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(282, 253);
            this.Controls.Add(this.button1);
            this.Name = "Form1";
            this.Text = "Form1";
            this.ResumeLayout(false);
        }
```

```
 // Form1
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
  this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.BackColor = System.Drawing.Color.MediumSeaGreen;
            this.ClientSize = new System.Drawing.Size(282, 253);
            this.Controls.Add(this.button1);
            this.Name = "Form1";
            this.Text = "Form1";
            this.ResumeLayout(false);
```



Свойства

Visual Studio 2005, C++
Windows Forms Application

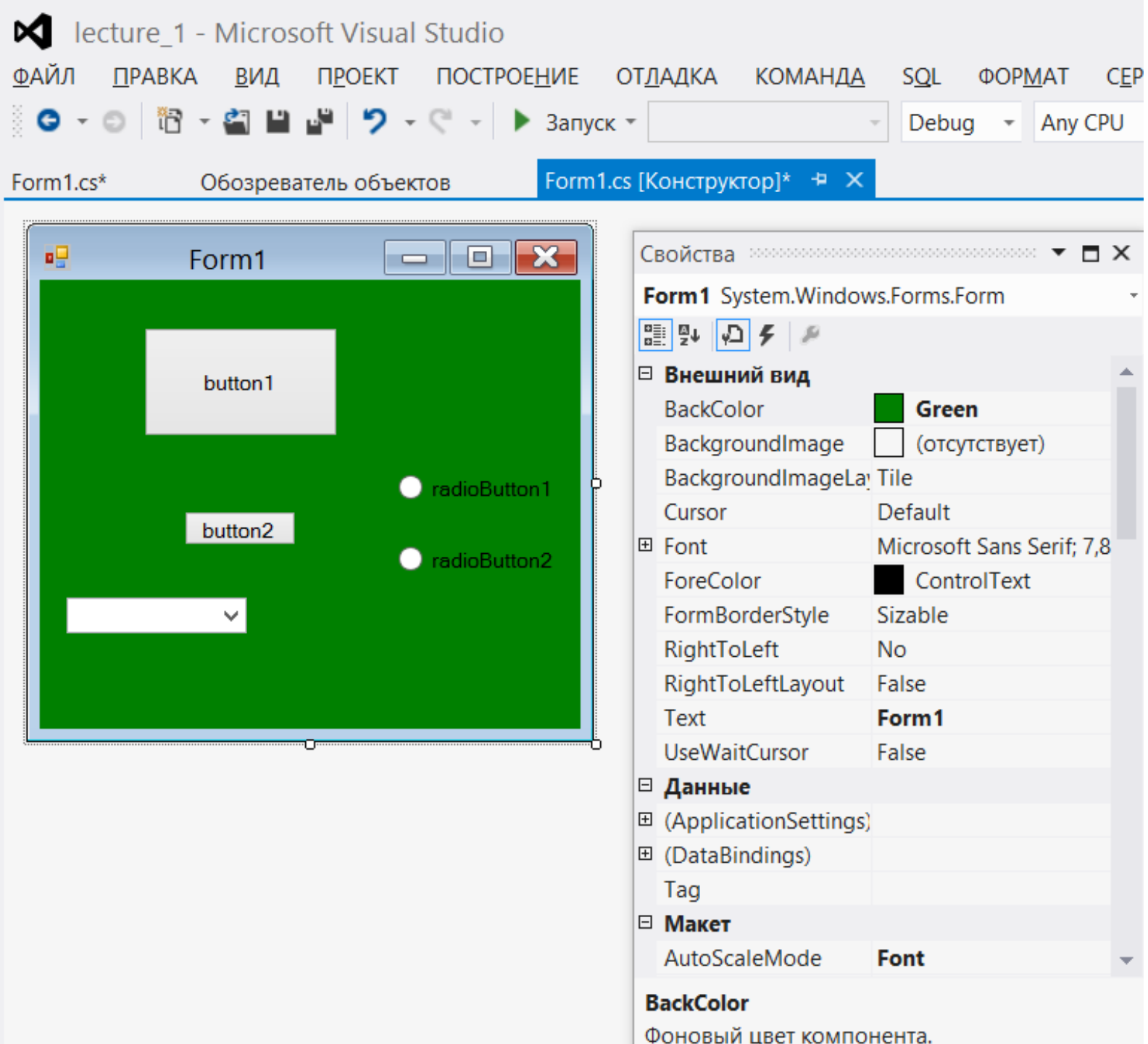
Пример описания стандартного свойства **BackColor** - цвет фона элемента управления

public:

```
virtual property Color BackColor  {
                    Color   get();
                     void  set  (Color value);
 }
```

Visual Studio 2012

public :override property **System::String^Text** { get; set; }


lecture_1 - Microsoft Visual Studio
ФАЙЛ ПРАВКА ВИД ПРОЕКТ ПОСТРОЕНИЕ ОТЛАДКА КОМАНДА SQL ФОРМАТ СЕР
Запуск
Debug
Any CPU
Form1.cs*
Обозреватель объектов
Form1.cs [Конструктор]*
Form1
button1
radioButton1
button2
radioButton2
Свойства
Form1 System.Windows.Forms.Form
Внешний вид
BackColor Green
BackgroundImage (отсутствует)
BackgroundImageLa Tile
Cursor Default
Font Microsoft Sans Serif; 7,8
ForeColor ControlText
FormBorderStyle Sizable
RightToLeft No
RightToLeftLayout False
Text Form1
UseWaitCursor False
Данные
(ApplicationSettings)
(DataBindings)
Tag
Макет
AutoScaleMode Font
BackColor
Фоновый цвет компонента.

```
private void button1_Click(object sender, EventArgs e)
        {
            this.BackColor = System.Drawing.Color.Green;
        }
```

```
public override Color BackColor { get; set; }
```

Работа со свойствами. Пример на C#

создадим класс `Client`

```
public class Client
{
    private string Name;
    private string Passport;
    private DateTime BirthDate;

   public Client()
   {

   }
}
```

Доступ к полям – элементам класса должен быть организован либо посредством методов, либо с помощью свойств класса.
Создадим свойства класса Client, обеспечивающие чтение и запись значений полей класса.

```
public string passport
{
  get
  {
    return Passport;
  }
  set
  {
    Passport = value;
  }
}
```

```
public string name
{
  get
  {
    return Name;
  }
  set
  {
    Name = value;
  }
}
```

Свойство состоит из методов **set** и **get**. При этом свойство должно содержать хотя бы один из методов.

Set позволяет изменять значение поля класса, **get** - получать значение.

В метод **set** передается значение параметра с помощью переменной value.

Оба метода могут содержать произвольное количество операторов, описывающих алгоритм выполнения действий в процессе чтения или записи значения в поле класса.

В данном примере свойства **passport** и **name** позволяют просто получить доступ к полям класса, считывая или устанавливая значения соответствующих переменных.

```
public DateTime  birthdate
{
 get
 {
  return BirthDate;
 }
 set
 {
  if (DateTime.Now > value)
   BirthDate = value;
  else
   throw new Exception("Введена неверная дата рождения");
 }
}
```

Свойство **birthdate** также предназначено для чтения и записи значения поля класса BirthDate. **При чтении** значения *(операция **get**)* происходит просто передача значения полю BirthDate, **при записи** нового значения в это поле происходит проверка допустимости устанавливаемого значения поля.

В данном случае проверка сводится к сравнению нового значения даты рождения с текущей датой. Если устанавливаемое значение даты рождения больше либо равно текущей дате, генерируется исключение, которое не позволяет записать новое значение в поле класса.

```
public int age
{
  get
  {
    int a;
    a = DateTime.Now.Year - BirthDate.Year;
    return a;
  }
}
```

Свойство **age** применяется для получения текущего возраста клиента. Оно предназначено только для чтения значения, поэтому содержит только метод **get**.

При использовании свойства **age** происходит вычисление текущего значения возраста клиента в годах путем вычитания года рождения из текущего значения года.

Использование свойств аналогично использованию полей.

В следующем примере создается объект **a1** класса **Client**. Затем поля данного объекта заполняются значениями с использованием свойств:

```
Client  a1=new  Client();
a1.name = “Петр”;
a1.passport = “5001”;
a1.birthdate = new  DateTime(1995, 06, 09);
```

Свойство – это элемент класса, который предоставляет удобный механизм доступа к полю класса (чтение поля и запись).

Свойство – это нечто среднее между полем и методом класса.

Синтаксис обращения к свойству такой же, как к полю класса, но на самом деле компилятор преобразовывает это обращение к вызову соответствующего неявного метода.

Такой метод называется **аксессор** (accessor).

Существует два таких метода:

get (для получения данных)

set (для записи).

Объявление простого свойства:

```
[модификатор доступа] [тип] [имя_свойства]
{
  get
  {
    // тело аксессора для чтения из поля
  }
  set
  {
    // тело аксессора для записи в поле
  }
}
```

Свойство не определяет место в памяти для хранения поля, необходимо отдельно объявить поле, доступом к которому будет управлять свойство

```
public class Date
{
                                        //объявление поля
                                       //объявление свойства
        {
                                         // аксессор чтения поля
            {

            }
                                    // аксессор записи в поле
            {
                }
```

```
                    // создаем объект класса Date
                   // записываем в поле, используя аксессор set

                   // читаем поле, используя аксессор get
```

```
public class Date
{
        private int month = 7; //объявление закрытого поля
        public int Month        //объявление свойства
        {
            get             // аксессор чтения поля
            {
                 return month;
            }
            set             // аксессор записи в поле
            {
                if ((value > 0) && (value < 13))
                {
                        month = value;
                }
            }
        }
}
```

```
Date data = new Date();
    data.Month = 2; // записываем в поле, используя аксессор set

Console.WriteLine(data.Month); // читаем поле, используя аксессор get
```

Автоматические свойства

Автоматическое свойство – это очень простое свойство, которое, в отличии от обычного свойства, определяет место в памяти (создает неявное поле), но при этом не позволяет создавать логику доступа.

Структура объявления Автоматического свойства:

[модификатор доступа] [тип] [имя_свойства] { get; set; }

У аксессоров таких свойств отсутствует тело.

Автоматически реализуемые свойства используют, когда нет необходимости накладывать какие-либо ограничения на возможные значения неявного поля свойства.

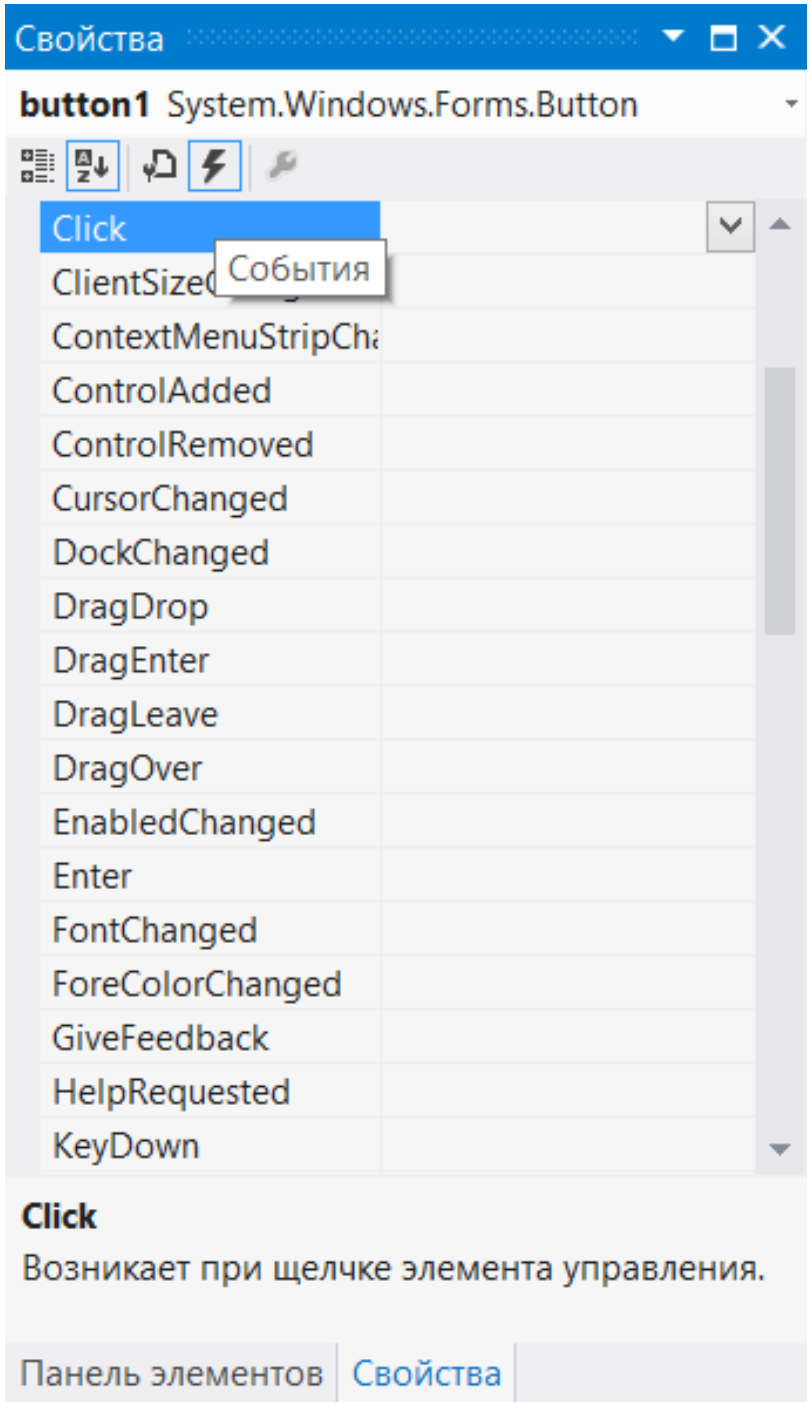
Пример использования:

```
public class Date

{
        // private int month = 7;
        public int Month {get; set;}

}
```

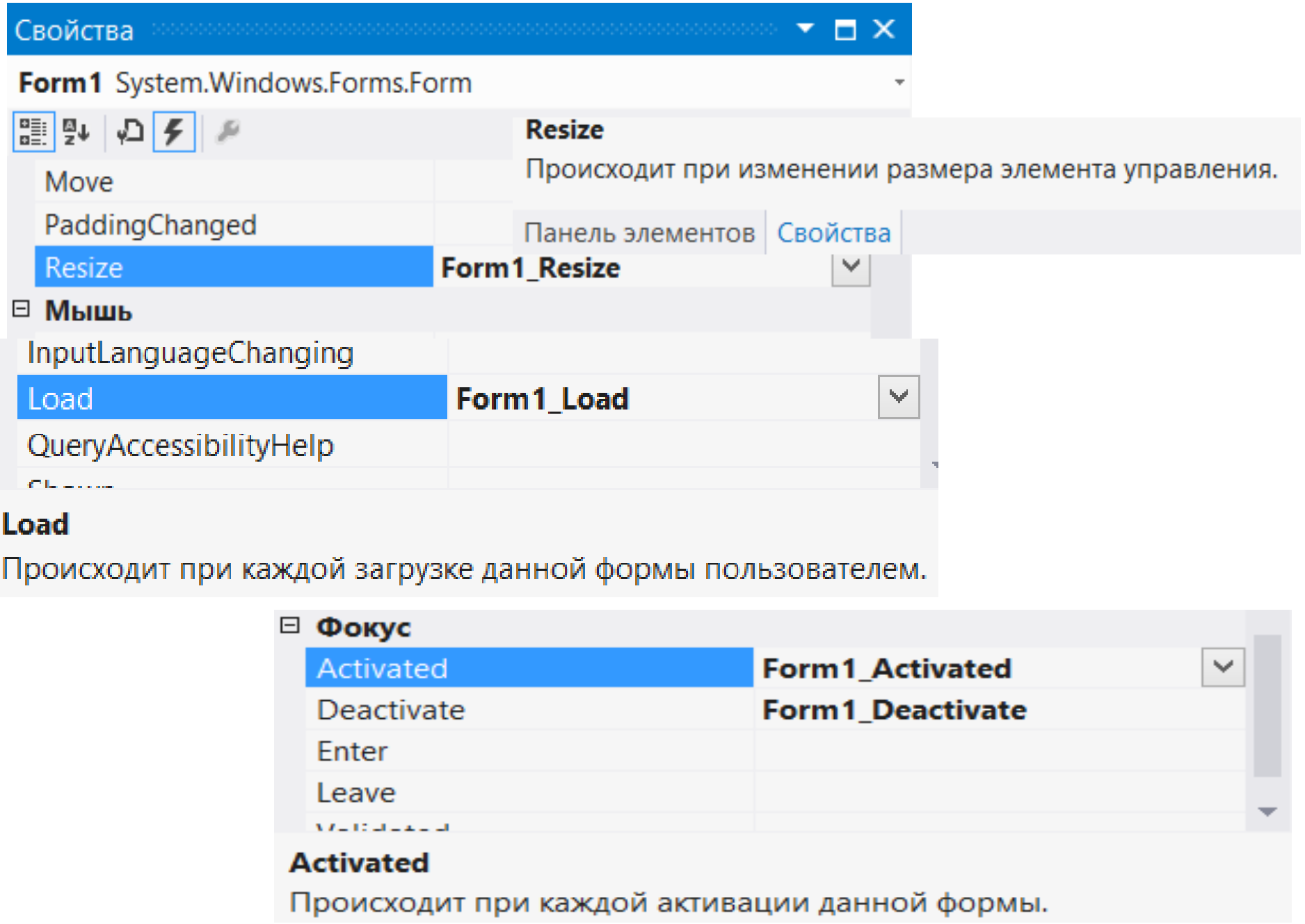
В отличии от простых открытых полей, у автоматических свойств есть возможность делать их только для чтения или только для записи.
Для этого используется модификатор доступа private перед именем аксессора:

```
public int Month  { private  get; set; } // свойство только на запись

public int Month  { get; private set; } // свойство только на чтение
```

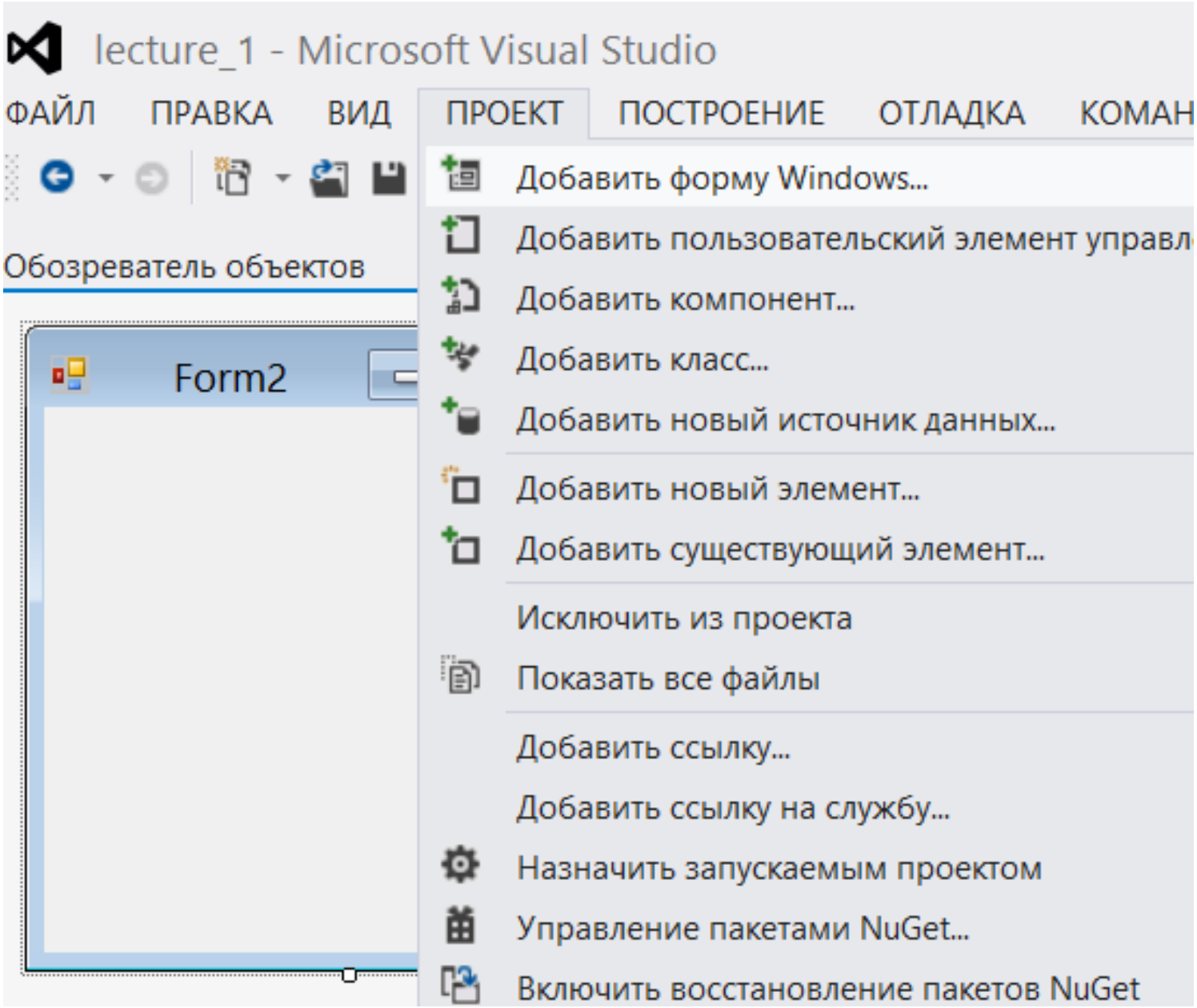
СОБЫТИЯ

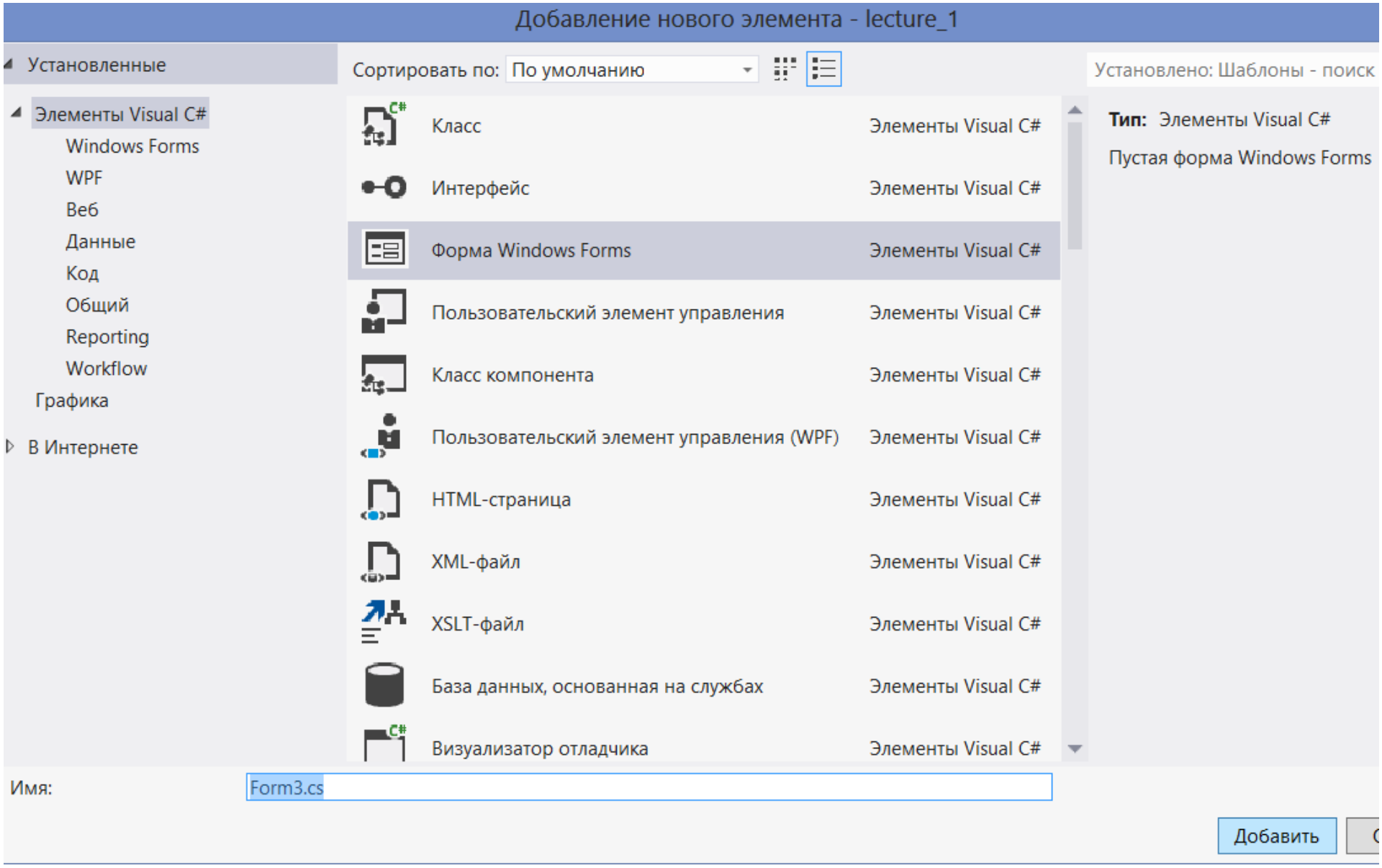

Свойства
button1 System.Windows.Forms.Button
Click
События
ClientSize
ContextMenuStripCha
ControlAdded
ControlRemoved
CursorChanged
DockChanged
DragDrop
DragEnter
DragLeave
DragOver
EnabledChanged
Enter
FontChanged
ForeColorChanged
GiveFeedback
HelpRequested
KeyDown
Click
Возникает при щелчке элемента управления.
Панель элементов
Свойства

Некоторые события формы



Добавим к Приложению новую форму




Добавление нового элемента - lecture_1
Установленные
Элементы Visual C#
Windows Forms
WPF
Веб
Данные
Код
Общий
Reporting
Workflow
Графика
В Интернете
Сортировать по: По умолчанию
Установлено: Шаблоны - поиск
Класс Элементы Visual C#
Интерфейс Элементы Visual C#
Форма Windows Forms Элементы Visual C#
Пользовательский элемент управления Элементы Visual C#
Класс компонента Элементы Visual C#
Пользовательский элемент управления (WPF) Элементы Visual C#
HTML-страница Элементы Visual C#
XML-файл Элементы Visual C#
XSLT-файл Элементы Visual C#
База данных, основанная на службах Элементы Visual C#
Визуализатор отладчика Элементы Visual C#
Тип: Элементы Visual C#
Пустая форма Windows Forms
Имя:
Form3.cs
Добавить

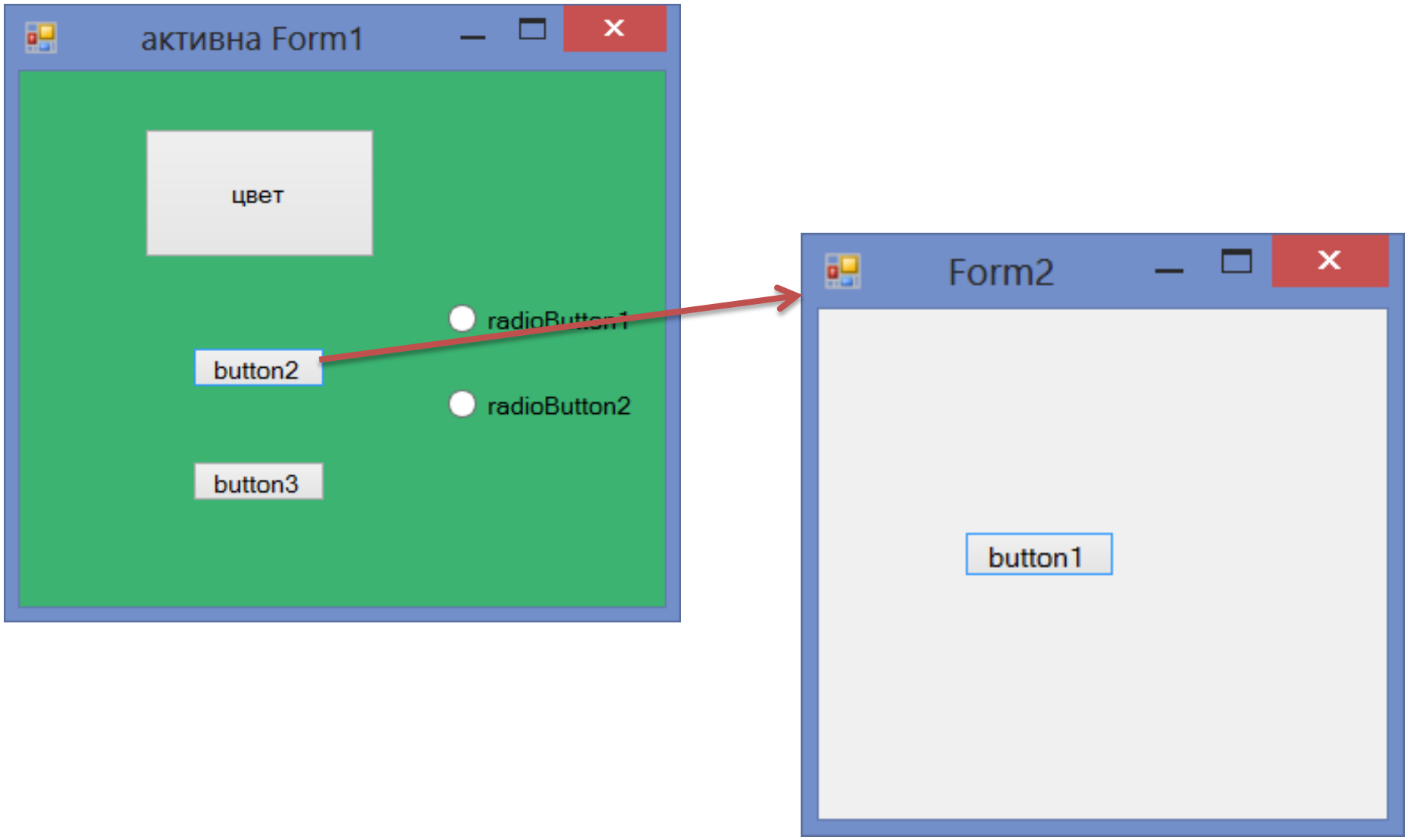
Вариант 1

```
private void button2_Click(object sender, EventArgs e)
        {
            Form2 form2 = new Form2();
            form2.Show();

        }
```

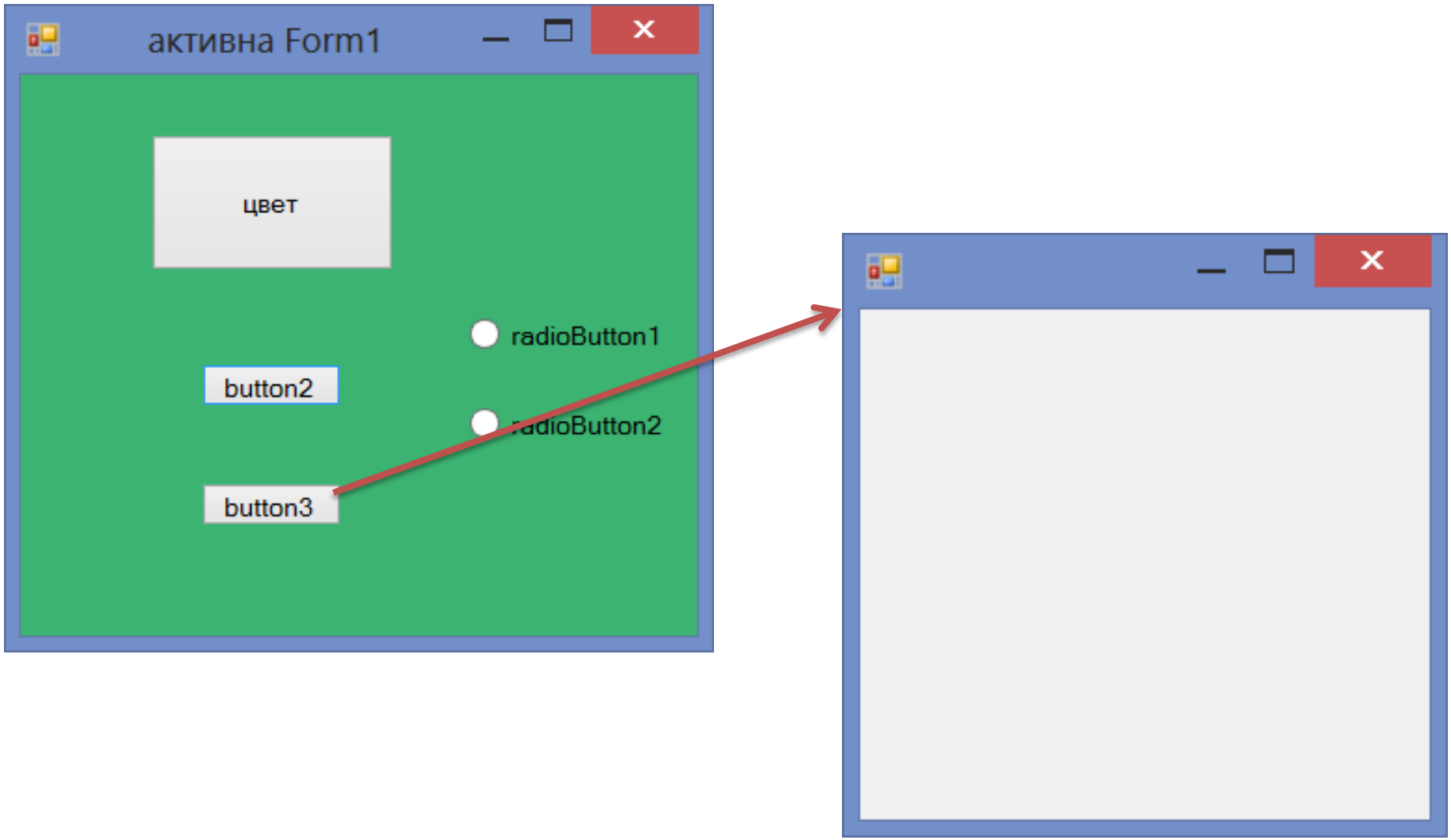
Вариант 2

```
private void button3_Click(object sender, EventArgs e)
        {
            Form f = new Form();
            f.Show();
        }
```

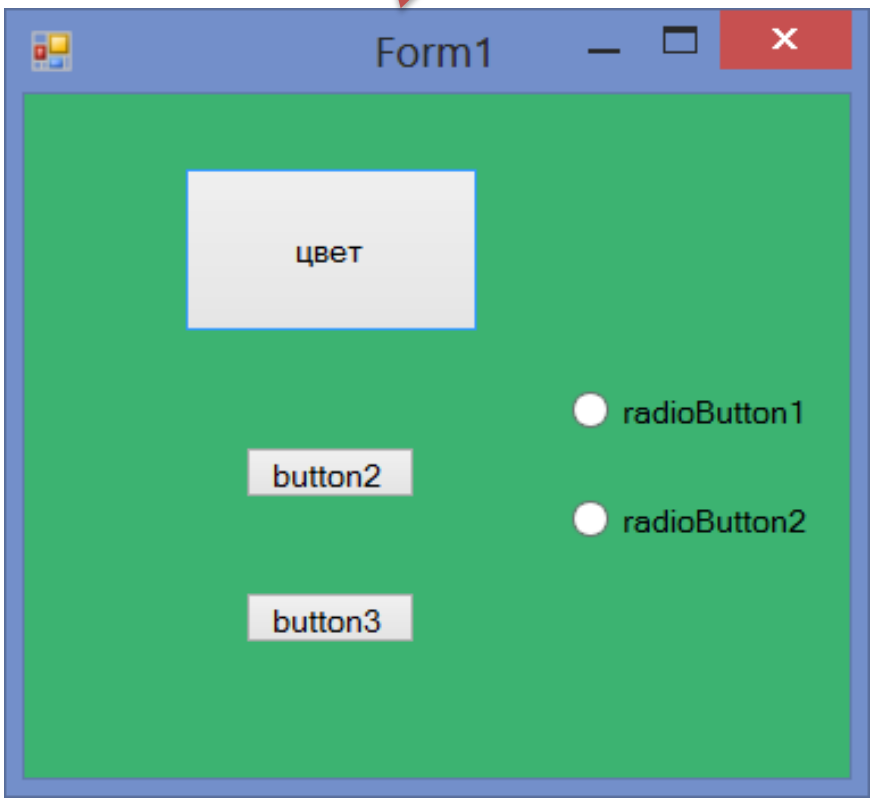
Вариант 1

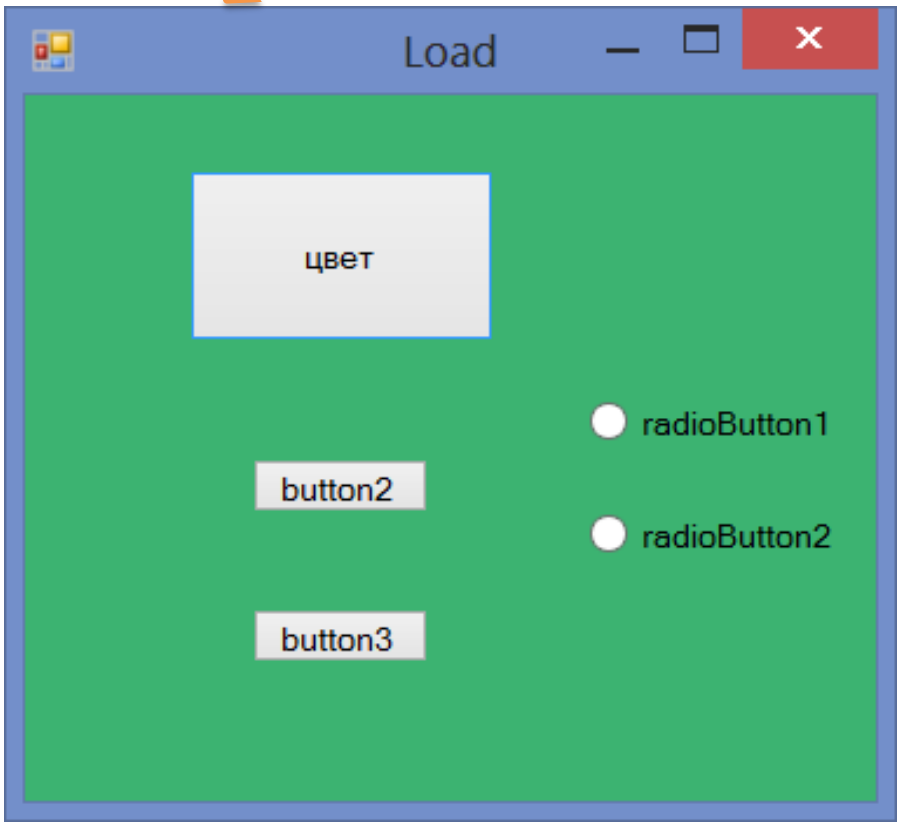

активна Form1
цвет
button2
radioButton1
radioButton2
button3
Form2
button1

Вариант 2

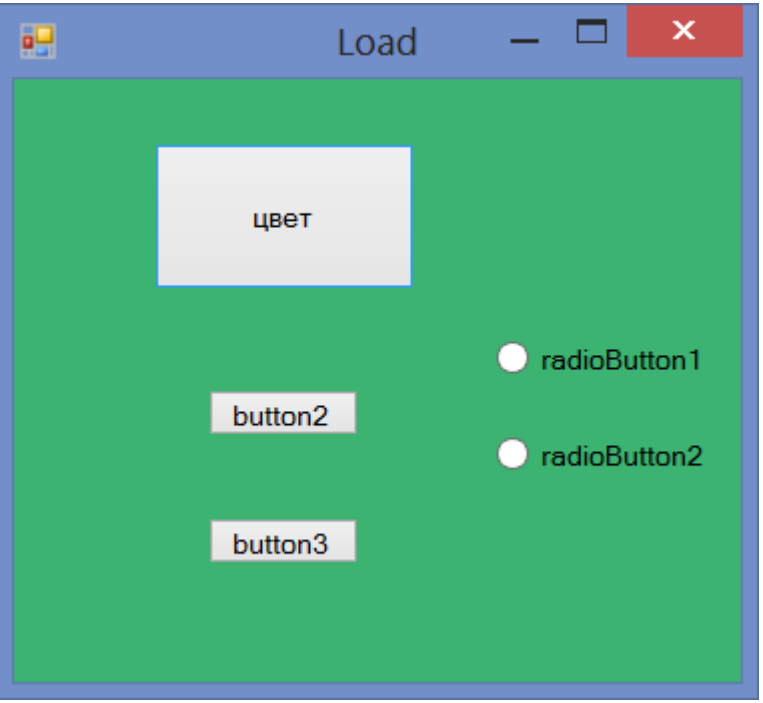


```
private void Form1_Load(object sender, EventArgs e)
        {
            //this.Text = "Load";
                this.Text = "Load";
        }
```



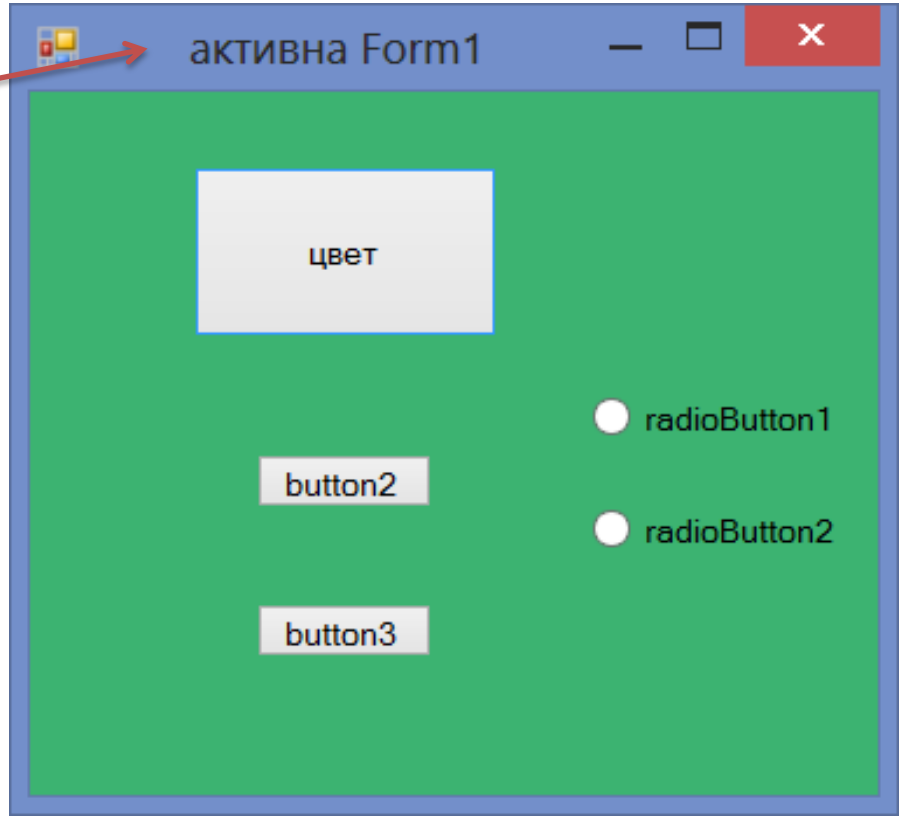


```
private void Form1_Resize(object sender, EventArgs e)
        {
            this.Text = "Resize";
        }
```



```
private void Form1_Activated(object sender, EventArgs e)
        {
            this.Text = "активна Form1";
        }
```

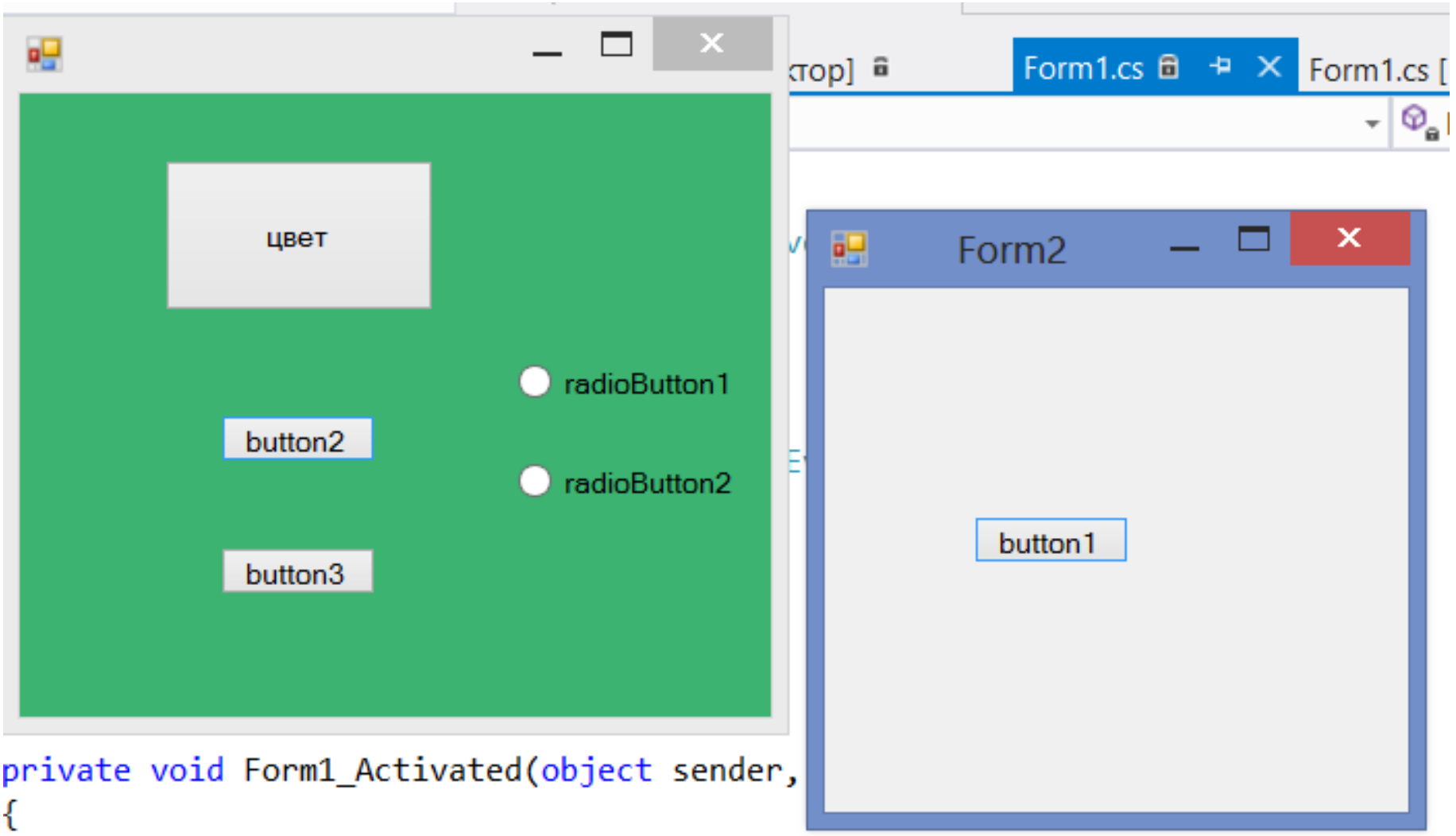
После запуска Приложения:




```
private void Form1_Load(object
sender, EventArgs e)
        {
           this.Text = "Load";
        }
```

Form1_Load ⟶ Form1_Activated

```
private void Form1_Deactivate(object sender, EventArgs e)
        {
            this.Text = "";
        }
```




```
private void Form1_Activated(object sender,
{
    this.Text = "активна Form1";
```