

Лабораторная работа № 1

Знакомство с Visual Studio 2010. Создание простейших приложений с графическим интерфейсом

Цели работы:

1. Изучить основы работы в среде Visual Studio 2010.
2. Изучить основные свойства и методы элемента Windows Form
3. Создать простейшее приложение с графическим интерфейсом.
4. Проанализировать поведение созданного приложения, в зависимости свойств формы.

1. Знакомство с интегрированной средой разработки (IDE) Visual Studio 2010

1.1. Запуск Visual Studio 2010 Professional или Microsoft Visual Studio 2010 Express в Windows 7

В стартовом меню Windows, щелкнуть кнопку **Пуск**, выбрать **Все программы**, переместиться к пункту **Microsoft Visual Studio 2010** или. **Microsoft Visual Studio 2010 Express**. Щелкнуть и в открывшемся подменю активизировать **Microsoft C# 2010 Express**. Начнется запуск среды IDE.

1.2. Версии Visual Studio 2010

Версия Microsoft Visual Studio 2010 Express позволяет работать со следующими языками программирования: Visual Basic, C# и C++ , а также позволяет использовать технологию ASP.NET для создания Web-приложений. Для каждого из этих языков существует своя интегрированная среда разработки.

Кроме версии Microsoft Visual Studio 2010 Express существует несколько версий Visual Studio 2010. Каждая из них предназначена для определенного уровня навыков работы программиста и для определенной системы организации, в которой используется продукт.

Microsoft Visual Studio 2010 Professional	Для независимых разработчиков: выполнение основных задач по разработке. Все дальнейшее изложение материала будет основываться на данной среде. Рекомендуется получить легальную версию этой среды через DreamSpark.ru
Microsoft Visual Studio 2010 Premium	Для независимых разработчиков и рабочих групп: создание масштабируемых и высококачественных приложений.
Microsoft Visual Studio 2010 Ultimate	Средства управления жизненным циклом для рабочих групп: высокие результаты на всех этапах разработки — от проектирования до развертывания.
Microsoft Visual Studio 2010 Test Professional	Специализированный набор инструментов для отделов контроля качества: упрощение планирования тестирования и выполнение ручных тестов.
Microsoft Visual Studio Team Foundation Server 2010	Платформа взаимодействия, лежащая в основе решения ALM от Microsoft.
Microsoft Visual Studio Team Explorer Everywhere 2010	Открывает возможности Team Foundation Server для Java™ и групп кроссплатформенной разработки.

Дополнительная информация:

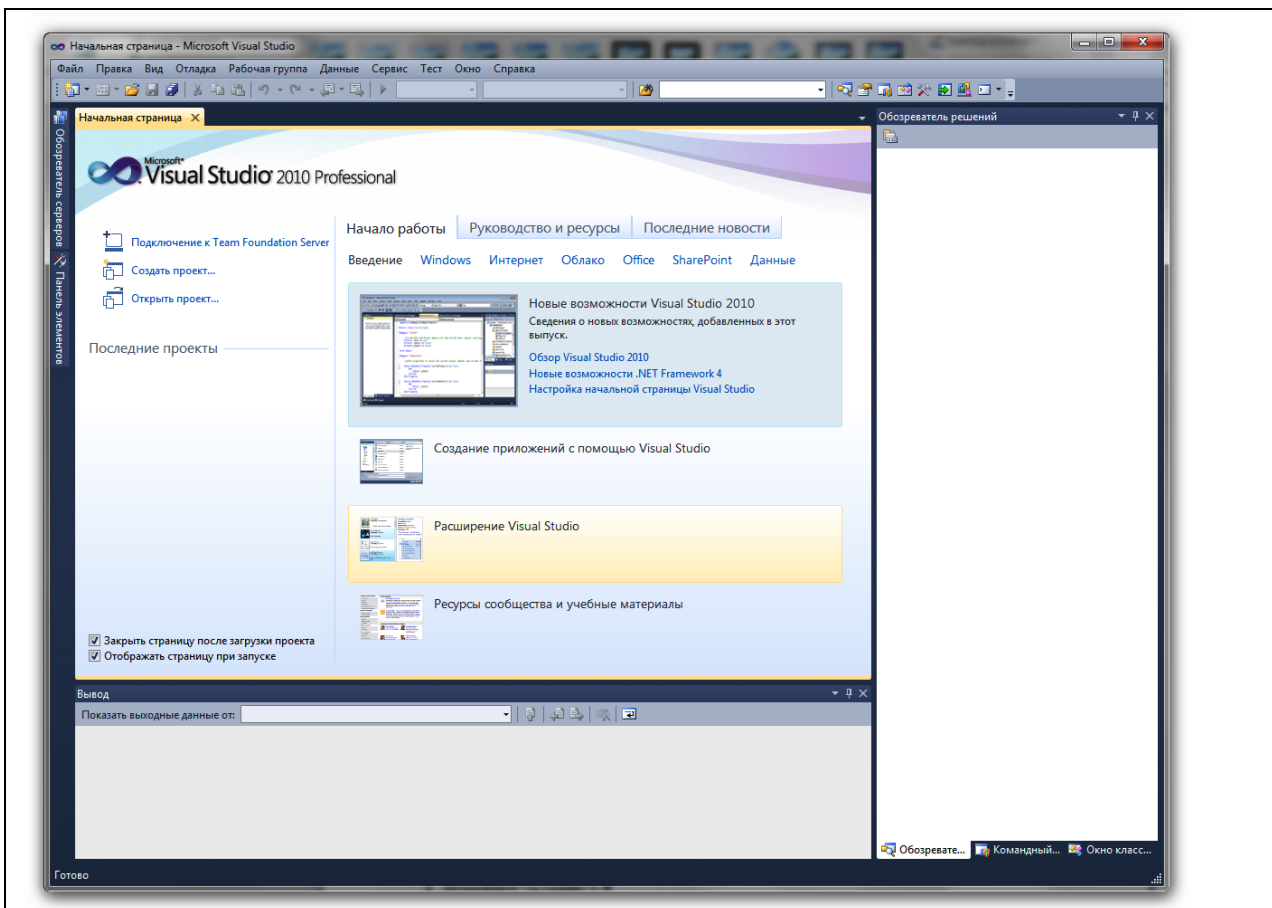
<http://www.microsoft.com/visualstudio/ru-ru/products>

1.3. Обзор интегрированной среды разработки (IDE)

После запуска программы **Microsoft Visual Studio 2010**, откроется окно IDE.

Интегрированная среда разработки (IDE), состоящая из нескольких элементов: строки меню, панели инструментов Стандартная, различных закрепленных или автоматически скрываемых окон инструментов в левой, нижней или правой областях, а также области редакторов. Набор доступных окон инструментов, меню и панелей инструментов зависит от типа проекта или файла, в котором выполняется разработка.

После загрузки центральное место в интегрированной среде занимает **Начальная страница**, на которой расположено несколько областей.



Домашняя страница Visual Studio — это страница Windows Presentation Foundation (WPF) на языке Extensible Application Markup Language (XAML), которая выполняется в окне инструментов Visual Studio.

Начальная страница разделена на три основных раздела: раздел команд, в котором отображаются команды **Создать проект** и **Открыть проект**; список **Последние проекты**; область содержимого с вкладками **Начало работы**, **Руководство и ресурсы** и **Последние новости** (RSS-канал). В нижней части страницы находятся параметры, которые задают, когда должна отображаться начальная страница.


Домашнее задание 1.

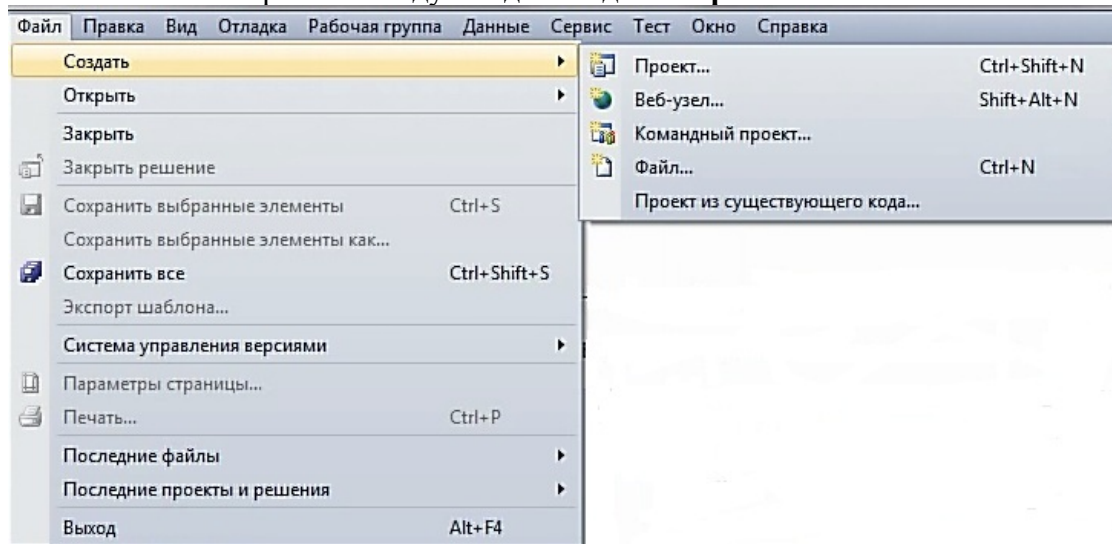
Самостоятельно ознакомиться с основными возможностями Начальной страницы. Подготовить краткий письменный обзор по каждому элементу интерфейса данной страницы. Этот обзор записать в тетрадь для домашних заданий. Срок сдачи домашнего задания две недели, после получения лабораторной работы. В тетради по самостоятельным работам может быть вклеен распечатанный вариант ответа.
Рекомендации. Воспользоваться библиотекой MSDN в Интернете или локальной справкой по Visual Studio

1.3.1. Создание проекта

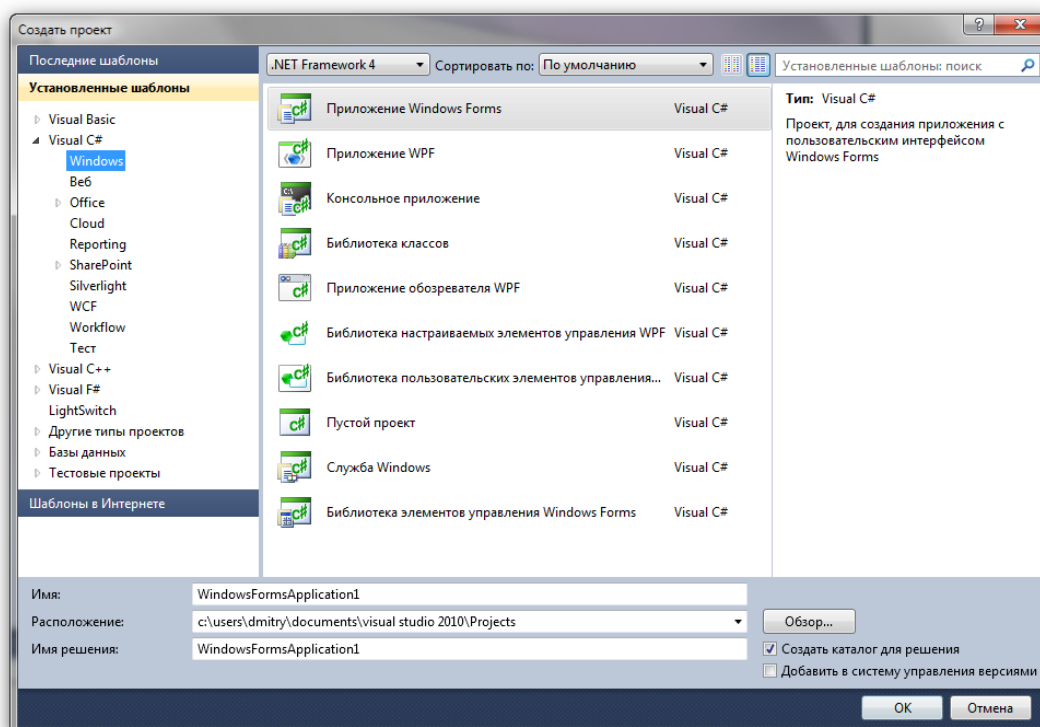
Запустить Visual Studio 2010 (VS).

Создать проект можно двумя способами:

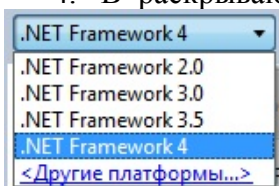
1. Выбрать  **Создать проект...** на Начальной странице.
2. В меню **Файл** выбрать команду **Создать** и далее **Проект**.



Откроется диалоговое окно **Создать проект**.



3. На панели **Установленные шаблоны** выбрать **Visual C#** и далее **Windows**.
4. В раскрывающемся списке можно выбрать требуемую версию .NET Framework.



5. В средней области надо выбрать шаблон **Приложение Windows Forms**.
6. В текстовом поле **Имя** задается имя проекта.
7. В текстовом поле **Расположение** указывается каталог для сохранения проекта.

8. Нажать кнопку **ОК**.

Откроется конструктор Windows Forms и отобразится форма Form1 проекта.

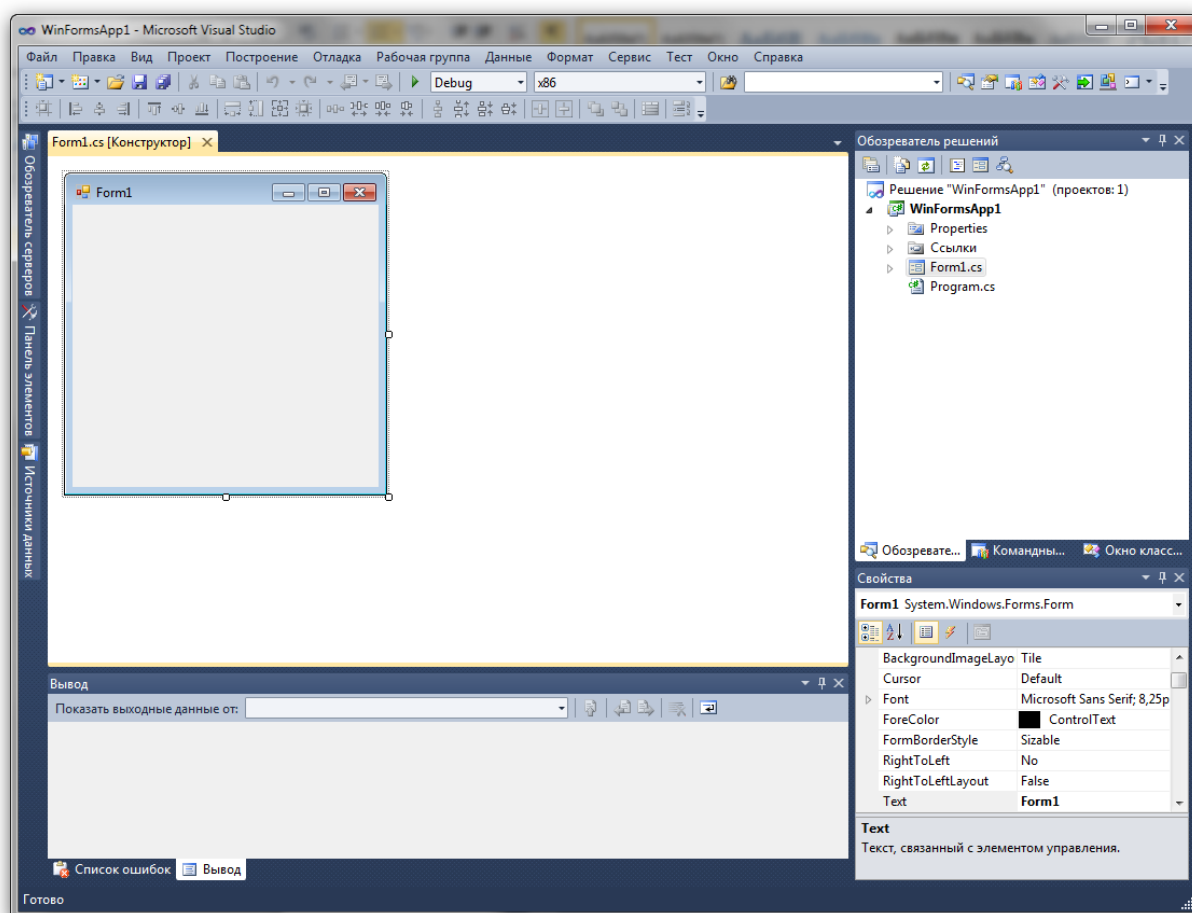


Рис. 1. Начальное состояние среды разработки VS при создании всех приложения Windows Form.

Во всех лабораторных работах курса в текстовом поле **Имя** должно указываться имя создаваемого приложения. Имя должно состоять из двух частей. Первая часть состоит сокращения Lab, знака подчеркивания, далее идет номер лабораторной работы (данная работа имеет номер 1) и через подчеркивание номер задания в лабораторной работе (z1). Вторая часть образуется из фамилии, имени и отчества в латинском написании. Например, Gramakov Dmitry Anatolevich – GDA.

Таким образом, первое задание в данной лабораторной работе должно иметь имя Lab_1_z1_GDA. В последующих лабораторных работах имена могут выглядеть следующим образом: Lab_2_z3_GDA – третье задание во второй лабораторной работе или Lab_10_z4_GDA – четвертое задание в десятой лабораторной работе.

В текстовом поле **Расположение** указывается каталог для сохранения проекта. Рекомендуется создать свой каталог на диске компьютера и размещать в нем свои проекты. Возможен вариант, когда в качестве места расположения используется Flash-устройство. Такой вариант предпочтительней, так как проще сохранять создаваемые приложения от случайного удаления на университетских компьютерах (Рис.2).

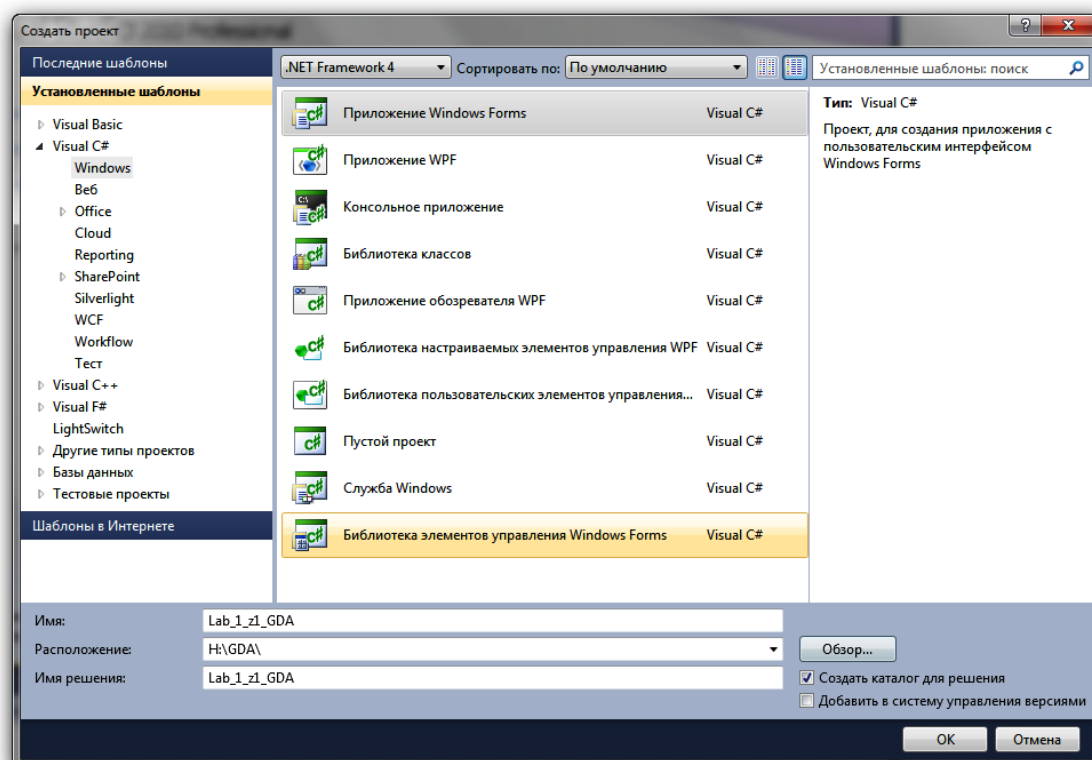


Рис.2. Пример создания приложения в лабораторной работе.

После указания в текстовых полях **Имя** и **Расположение** соответствующих имен, появится следующее окно (Рис. 3).

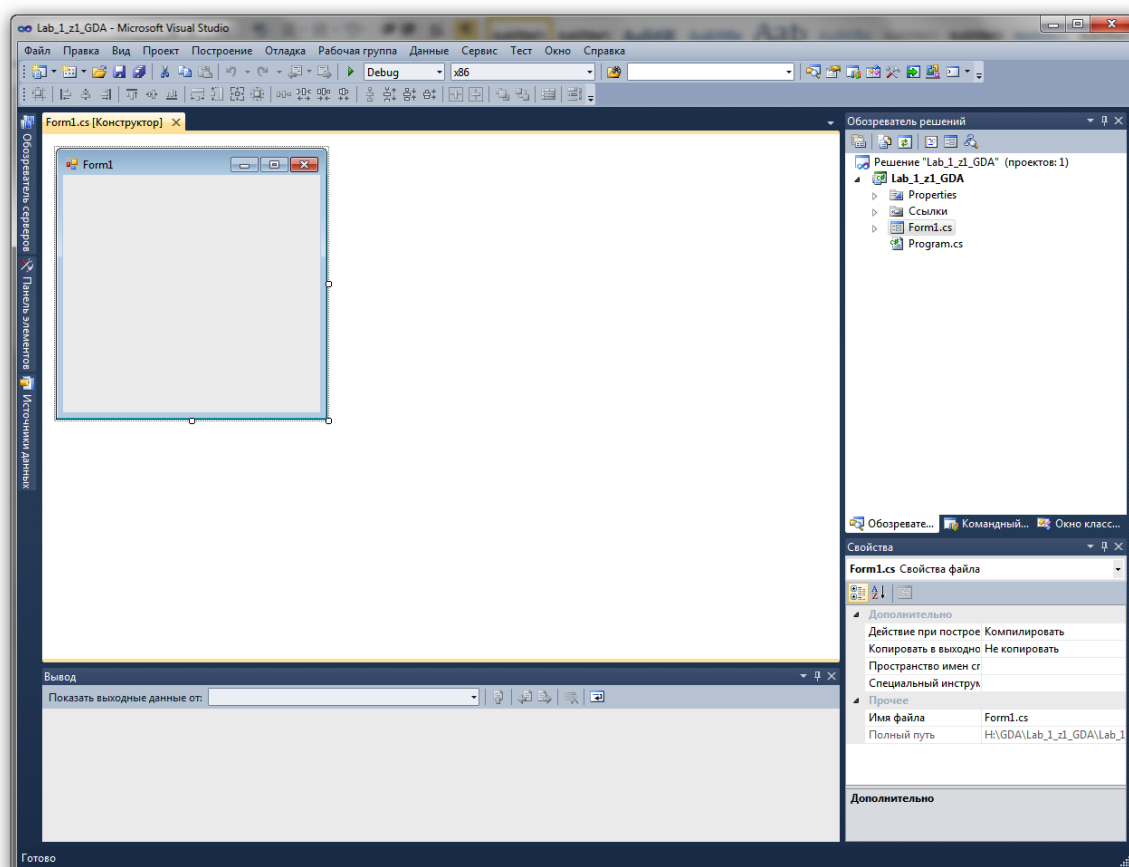


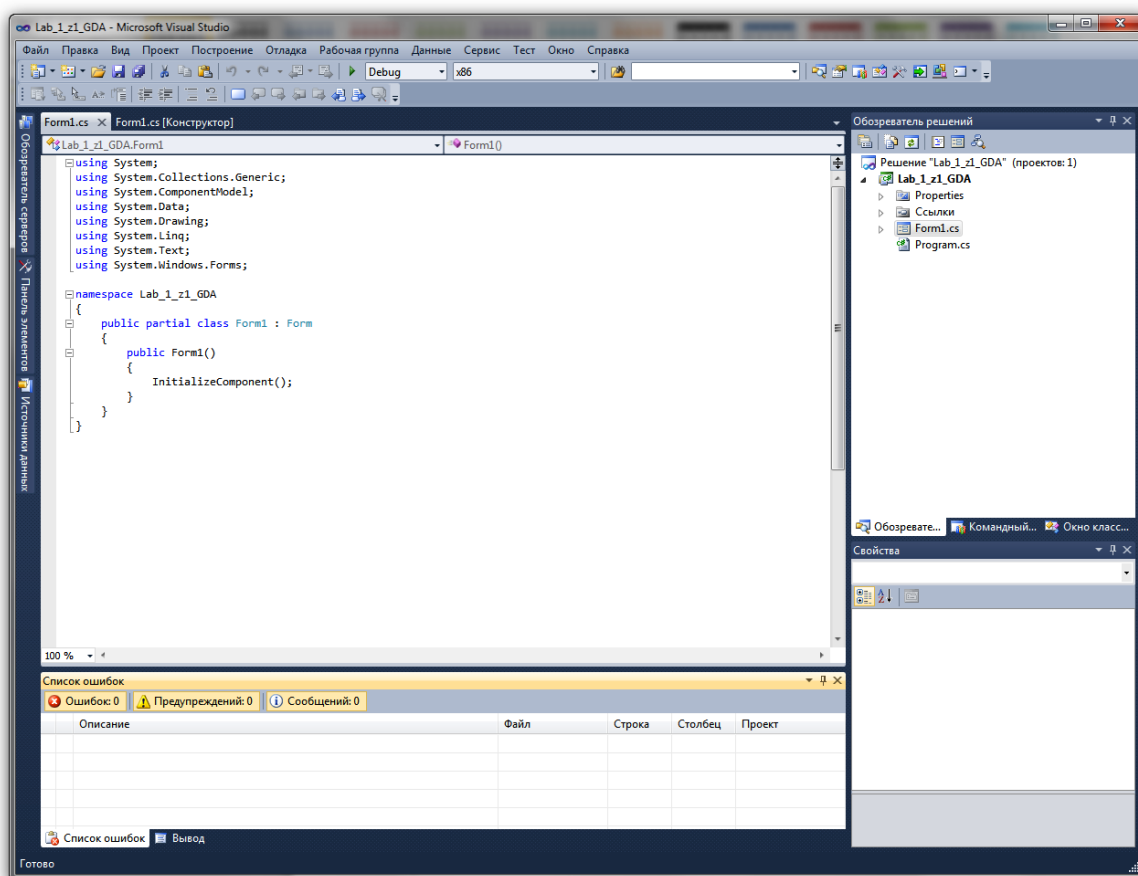
Рис. 3. Интегрированная среда разработки при создании приложения в лабораторной работе №1 задание 1.

Большую часть открытого окна занимает область, в которой располагается или редактор кода, или конструктор. Выбор используемых редакторов и конструкторов зависит от типа создаваемого файла или документа. В нашем случае, в этом окне открыт **Конструктор** формы, который открывается автоматически при создании приложений Windows Form.

Для открытия **Редактора кода** можно использовать несколько способов:

- В **Обозревателе решений** выбрать форму или модуль и нажать кнопку **Перейти к коду**.
- Щелкнуть правой кнопкой мыши любой элемент управления в области представления **Конструктор** и выбрать из контекстного меню пункт **Перейти к коду**.
- В меню **Файл** выбрать команду **Открыть файл** или **Создать файл** и открыть файл исходного кода для редактирования.
- Можно воспользоваться клавишей **F7**.

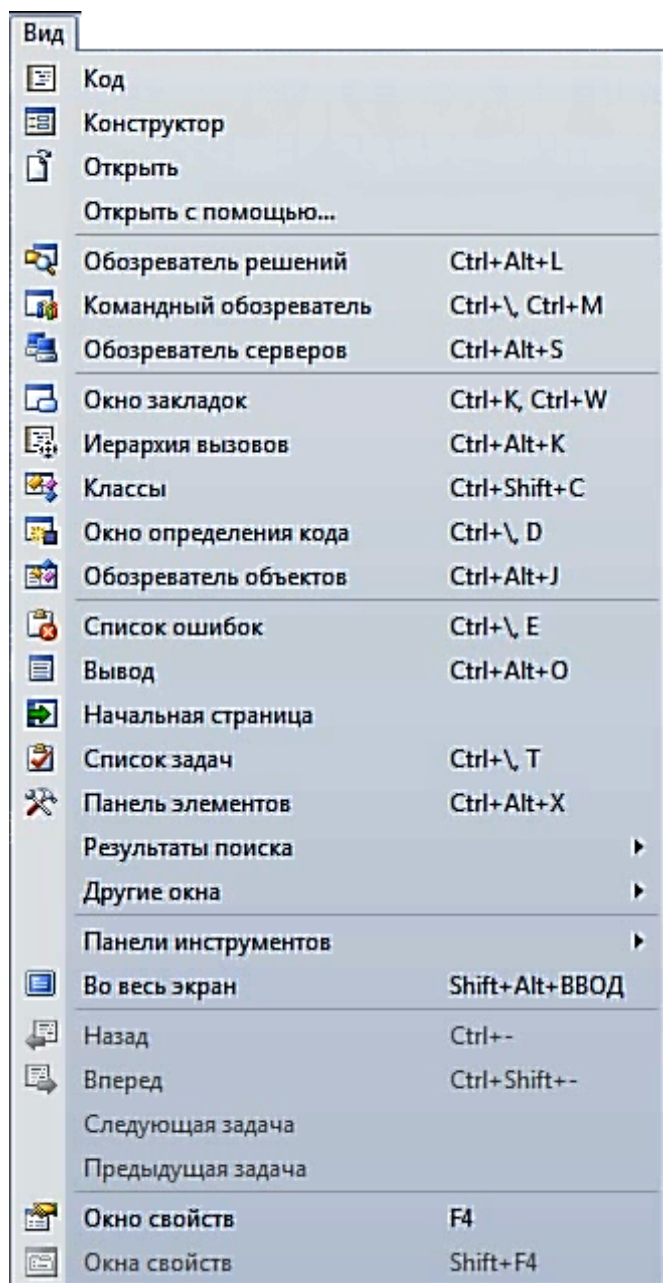
При открытии Редактора кода окно будет выглядеть следующим образом:



В окне интегрированной среды разработки располагаются еще несколько важных окон при разработке Windows приложений:

- **Обозреватель решений;**
- **Окно Свойства;**
- **Список ошибок;**
- **Панель элементов.**

Всеми отображаемыми окнами и другими окнами, которые не отображаются при начальной загрузке, можно управлять через меню **Вид**.



Домашнее задание 2.

Самостоятельно ознакомиться с основными возможностями меню Вид.

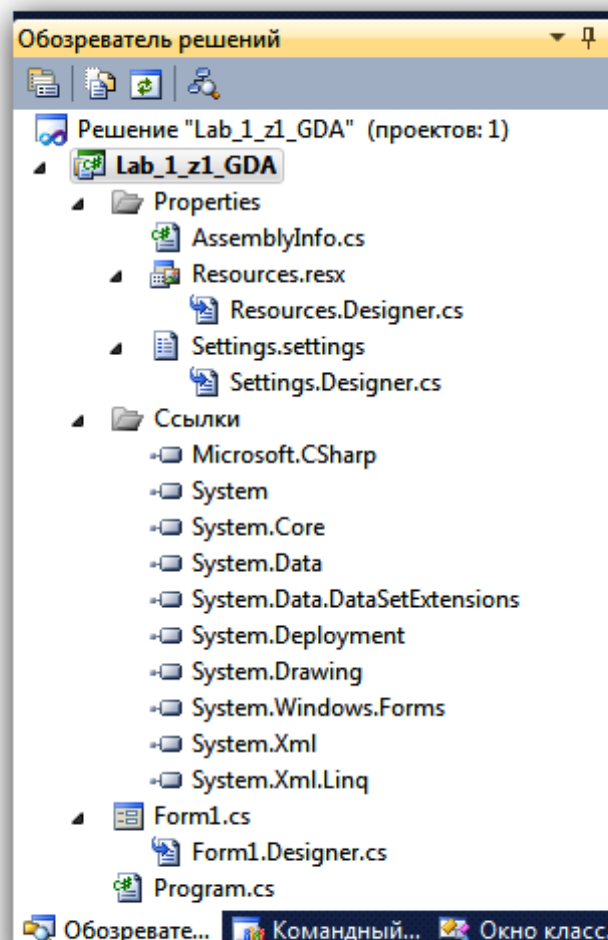
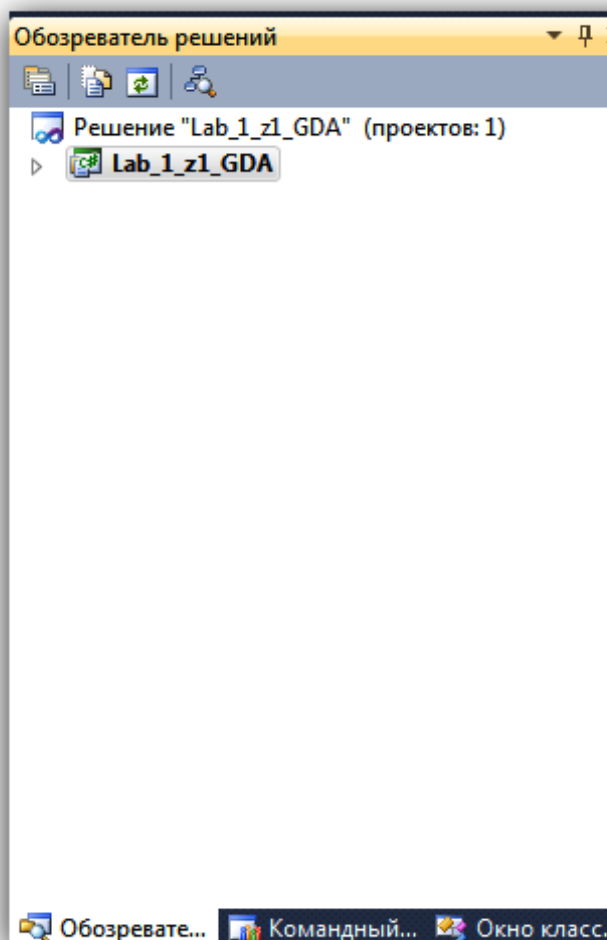
Подготовить краткий письменный обзор по каждому элементу данного меню. Этот обзор записать в тетрадь для домашних заданий. Срок сдачи домашнего задания две недели, после получения лабораторной работы. В тетради по самостоятельным работам может быть вклеен распечатанный вариант ответа.

Рекомендации. Воспользоваться библиотекой MSDN в Интернете или локальной справкой по Visual Studio

1.3.2.Обзорщик решений

Обзорщик объектов позволяет выбирать и анализировать элементы, доступные к использованию в рамках проектов. Для открытия **Обзорщик объектов**, выбрать соответствующую команду в меню **Вид** или нажать кнопку **Обзорщик объектов** на главной панели инструментов.

Вид Обзорщика решений в свернутом и развернутом виде. В развернутом виде показываются все элементы формируемые средой при создании проекта.

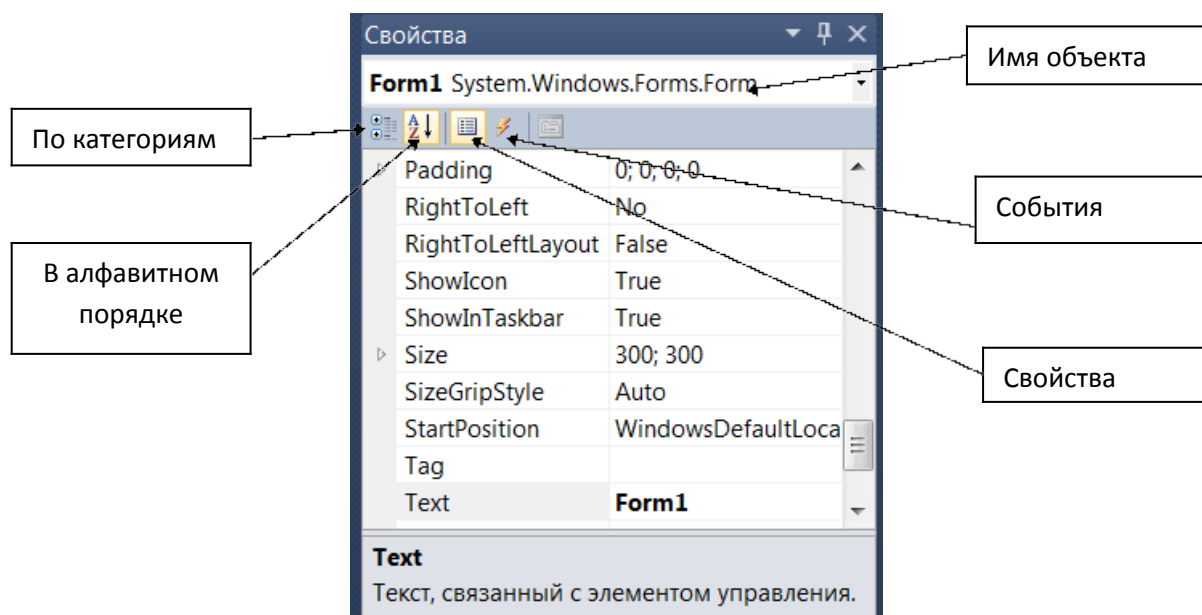


1.3.3.Окно Свойства

Это окно предназначено для просмотра и внесения изменений в заданные во время разработки свойства и события выделенных объектов, расположенных в редакторах и конструкторах.

Окно **Свойства** можно также использовать для просмотра и изменения свойств файлов, проектов и решений. Окно **Свойства** можно открыть из меню **Вид**.

В зависимости от конкретного свойства в окне **Свойства** отображаются различные типы полей редактирования. К таким полям относятся: поля ввода, раскрывающиеся списки и ссылки на пользовательские диалоговые окна редактора. Свойства, отображаемые серым цветом, доступны только для чтения.



Имя объекта	Список выделенных объектов. Отображаются только объекты активного редактора или конструктора. Если выбрано несколько объектов, отображаются общие для всех объектов свойства.
По категориям	Список всех свойств и их значений для выделенного объекта по категориям. Чтобы сократить число отображаемых свойств, можно свернуть категорию. При разворачивании или свертывании категории слева от ее имени отображается знак "плюс" (+) или "минус" (-). Категории перечислены в алфавитном порядке.
В алфавитном порядке	Сортировка всех свойств и событий времени разработки для выделенных объектов по алфавиту. Чтобы изменить значение незатененного свойства, щелкните ячейку справа от него и введите изменения.
Свойства	Отображение свойств объекта.
События	Отображение событий объекта.

1.3.4. Список ошибок;

Действия, которые можно выполняются в окне **Список ошибок**.

- Отображение ошибок, предупреждений и сообщений, созданных во время редактирования и компиляции кода.
- Отображение синтаксических ошибок, обнаруженных технологией IntelliSense.
- Открытие файла, в котором обнаружена неполадка, и перемещение к месту расположения ошибки с помощью двойного щелчка любого сообщения об ошибке.
- Фильтрация отображаемых записей и столбцов данных, которые будут показаны в каждой записи.

Для открытия окна **Список ошибок**, в меню **Вид** щелкнуть **Список ошибок**.

С помощью кнопок **Ошибки**, **Предупреждения** и **Сообщения** можно выбирать отображаемые записи.

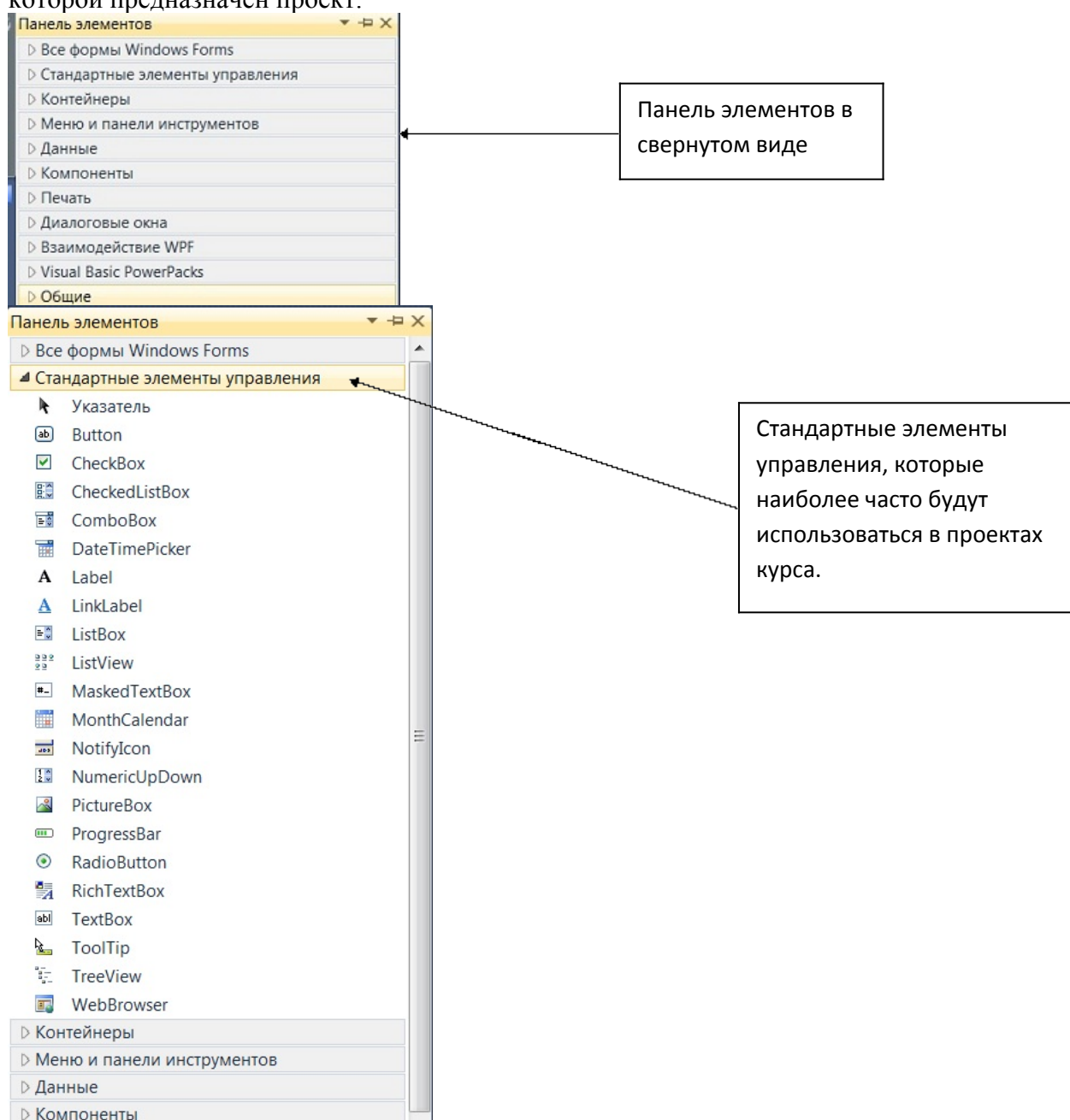
1.3.5. Панель элементов.

Панель элементов служит для отображения значков элементов управления и других элементов, которые можно добавлять в проекты Visual Studio.

Чтобы открыть **Панель элементов**, в меню **Вид** надо выбрать **Панель элементов**. По умолчанию **Панель элементов** находится в скрытом состоянии, зафиксировать ее в открытом состоянии можно с помощью кнопки **Автоматически скрывать** в строке окна.

Любой значок с **Панели элементов** можно перетащить в область представления "Конструктор" или вставить в редактор кода интегрированной среды разработки Visual Studio. В обоих случаях в активный файл проекта добавляется базовый код для создания экземпляра элемента, представленного на **панели элементов**.

На **Панели элементов** отображаются только элементы, относящиеся к текущему типу файла. Коллекция доступных элементов управления также зависит от версии .NET Framework, для которой предназначен проект.



Подробнее с особенностями работы в среде Visual Studio 2010 можно ознакомиться через локальную документацию или через Интернет на сайте MSDN.

2. Основные свойства и методы Windows Form.

Form используется для представления любого окна, отображаемого в приложении. Класс **Form** используется для создания стандартных окон, окон инструментов, необрамленных и перемещаемых окон. Класс **Form** также используется для создания модальных окон, например, диалогового окна. Особым видом формы является форма интерфейса MDI, содержащая другие формы, называемые дочерними MDI-формами. MDI-форма создается с

посредством задания свойству **IsMdiContainer** значения **true**. Дочерние MDI-формы создаются с помощью установки свойства **MdiParent** для родительской MDI-формы, которая будет содержать дочернюю форму.

С помощью свойств, доступных в классе **Form**, имеется возможность определять внешний вид, размер, цвет и компоненты управления создаваемого окна или диалогового окна. Многие свойства характерные для класса **Form**, имеются и у многих других элементов управления, так как все элементы, включая класс **Form** наследуются от класса **Control**.

Свойство	Значение	
Text	Задаёт имя приложения, отображаемого в строке заголовка.	
Size	Определяет размер окна при его отображении. В общем случае определяет размер любого элемента управления	
ForeColor	Основной цвет для отображения текста в данном элементе.	
BackColor	Фоновый цвет окна элемента	
Name	Возвращает или задаёт имя элемента управления.	
FormBorderStyle	Возвращает или задаёт стиль границы формы. Ниже приводятся возможные значения	
	None	Нет границы.
	FixedSingle	Фиксированная граница из одной линии.
	Fixed3D	Фиксированная трёхмерная граница.
	FixedDialog	Толстая фиксированная граница стиля диалогового окна.
	Sizable	Граница с изменяемыми размерами. По умолчанию.
	SizableToolWin- dow	Изменяемая граница окна как у панели инструментов.
	FixedToolWindow	Неизменяемая граница окна как у панели инструментов.
MinimizeBox и MaximizeBox	Позволяют управлять возможностью свёртывания, развёртывания или изменения размера формы во время выполнения.	

Подробно со свойствами любого элемента можно ознакомиться через окно **Свойства**, выбрав в нём соответствующий элемент.

Задание 1.

Создать простейший проект, как описано в разделе 1.3.1. Запустить его на исполнение – **F5** или меню **Отладка - Начать отладку**. Ещё не написано ни одной строчки кода, а приложение работает. Исследовать какие изменения будут происходить в приложении при изменении свойств **FormBorderStyle**, **MinimizeBox** и **MaximizeBox**. Воспользоваться окном **Свойства** устанавливая для вышеперечисленных свойства все возможные значения. Описать в отчёте, что будет происходить. Для каждого возможного набора комбинаций свойств, представить скрины выполнения. Скрин получается нажатием комбинации клавиш **Alt+PrtSc**. Далее он помещается в отчёт с помощью команды **Специальная вставка – Аппаратно-независимый рисунок**.

2.1. Работа с элементами управления в Windows Forms.

Windows Forms — это основной компонент пользовательского интерфейса. Каждая форма является контейнером, содержащим элементы управления и меню. Формы могут реагировать на события мыши и клавиатуры, поступающие от пользователя, и выводить на экран данные для пользователя с помощью элементов управления, которые содержатся в форме.

Если приложение взаимодействует с пользователем, оно должно содержать хотя бы одну форму. Более сложные приложения часто требуют нескольких форм для того, чтобы программа выполнялась последовательно и логично.

Большинство форм разрабатываются путем добавления элементов управления на поверхность формы с целью создания пользовательского интерфейса.

Элемент управления — это компонент на форме, использующийся для отображения сведений или ввода пользовательских данных.

Используются три способа для переноса элементов управления на форму

1. Последовать действий для отображения элемента управления заданного размера в форме.

1. Открыть форму в режиме конструктора.
2. В **Панели элементов** щелкнуть элемент управления, который требуется добавить в форму.
3. Щелкнуть место в форме, где должен располагаться левый верхний угол элемента управления, а затем перетащить указатель мыши на место, в котором должен располагаться правый нижний угол элемента управления. Элемент управления добавляется на форму в указанное место с указанными размерами.

2. Отображение элемента в левом верхнем углу формы.

Дважды щелкнуть по элементу управления в Панели элементов. Этот элемент с размером по умолчанию отобразится в левом верхнем угле формы.

3. Перетаскивание элемента управления на форму.

1. Открыть форму в режиме конструктора.
 2. В **Панели элементов** щелкнуть требуемый элемент управления и перетащить его в форму.
- Элемент добавляется в форму в указанное место с размером по умолчанию.

Элементы управления в Панели элементов могут быть двух типов: отображаемые и неотображаемые. Отображаемые элементы используются для создания пользовательского интерфейса приложения. Неотображаемые элементы управления используются для задач не связанных с интерфейсом приложения. Такие элементы при их выборе на Панели элементов, располагаются не на форме, а в окне конструктора под формой. Пример такого элемента управления является элемент Timer, формирующий в приложении повторяющиеся события.

Задание 2.

Разработать приложение, которое содержит элемент управления PictureBox. В этот элемент поместить любую картинку, по центру формы. Данная форма не должна иметь кнопок MinimizeBox и MaximizeBox в строке заголовка. Приложение должно загружаться и оканчивать свою работу по таймеру. Элемент управления Timer позволяет запустить процесс выполнения некоторого кода по окончании определенного промежутка времени. Длительность промежутка времени в миллисекундах задается в свойстве Interval. Исследовать какие еще свойства надо изменить, чтобы таймер работал. С элементом управления Timer необходимо связать событие. Это можно сделать, дважды щелкнув на этом элементе. В полученном методе с именем Timer1_Tick надо поместить метод this.Close().

2.1. События и элементы управления

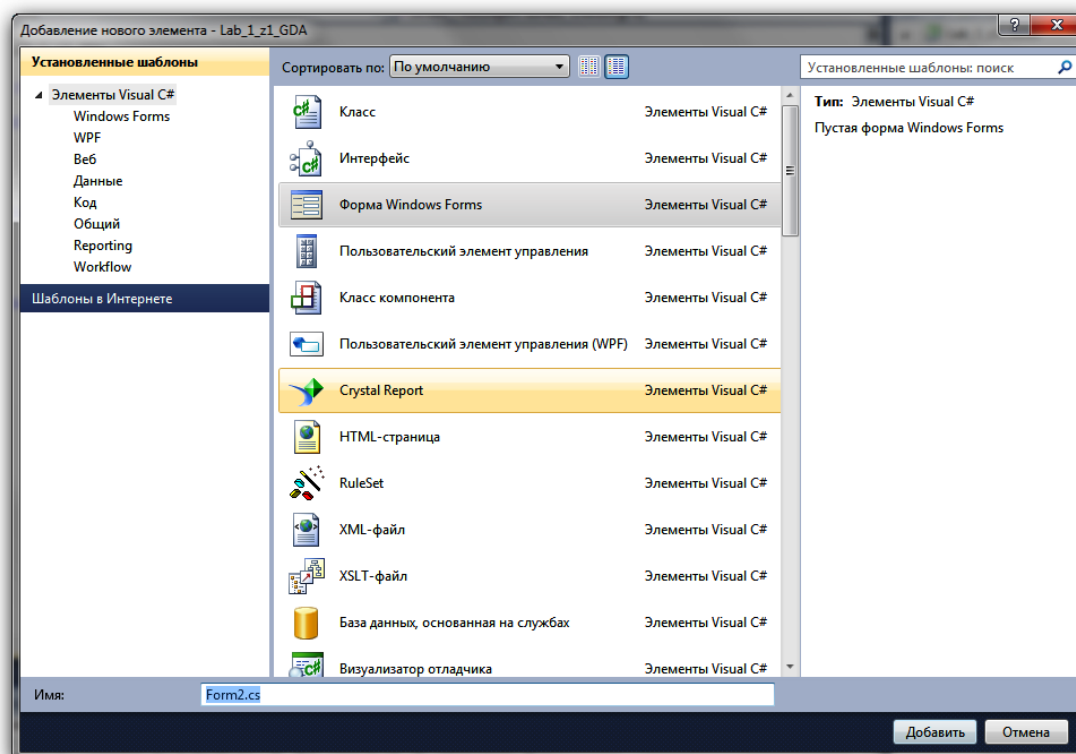
По соглашению события в .NET Framework имеют сигнатуру *Имя_события(sender, e)*, где *sender* — это объект Object, задающий ссылку на класс или структуру, породившую событие, а *e* — это объект EventArgs или класса, производного от EventArgs, предоставляющий данные события.

2.2. Добавление форм в проект

Большинство проектов будут требовать более одной формы. Дополнительные формы можно добавить и настроить во время разработки или же создать в коде во время выполнения экземпляры предварительно разработанных форм.

Добавление новой формы в проект во время разработки

Выбрать Добавить форму Windows в меню Проект. Появится диалоговое окно Добавление нового элемента.



Выбрать **Windows Form** и ввести имя новой формы в поле **Имя**. Щелкнуть **Добавить**, чтобы добавить форму в среду разработки.

Во время разработки можно добавлять и настраивать столько форм, сколько потребуется разрабатываемому приложению. Кроме того, существует возможность создавать в коде новые экземпляры форм. Этот метод чаще всего применяется, когда необходимо отобразить форму, которая уже была разработана

Чтобы получить доступ к свойствам и методам формы, используется особое ключевое слово *this* (C#).

Получение доступа к методам и свойствам формы из ее кода.

Используется ключевое слово *this* (C#). Например:

```
this.Text = "Это вторичная форма";
```

Кроме того, можно создать новые экземпляры форм во время выполнения, описав переменную, представляющую тип формы, и создав экземпляр данной формы.

Добавление формы в приложение во время выполнения

Объявить и создать экземпляр переменной, представляющей форму, и задать ей значение. Данный пример предполагает, что в проекте разработана форма с именем *Form2*:

```
Form2 myForm;
```

```
myForm = new Form2();
```

```
Отображение новой формы myForm.Show();
```

Задание 3.

Создать приложение с кнопкой, добавить в приложение вторую форму и связать появление второй формы с щелчком по кнопке.

Задание 4. Создать приложение с кнопкой. Щелчок по кнопке должен приводить к появлению окна сообщения **MessageBox**. Для показа сообщения использовать класс **MessageBox**, позволяющий отображать окно сообщения, содержащее текст, кнопки и символы, с помощью которых информируется и инструктируется пользователь. Должно быть спроектировано несколько вариантов приложения с различными вариантами отображения окна сообщения.

Задание 5.

Разработать графический интерфейс приложения, позволяющего проверить знание логических операций. Результат надо представить в виде рисунка в графическом редакторе, рисунка в Word или в Visual Studio. Должны быть показаны элементы управления, которые необходимо использовать в данной программе.