

# Лабораторная работа №1

## Базовые сведения о среде разработки Microsoft Visual Studio

При написании методического пособия использовалась русифицированная среда разработки Microsoft Visual Studio (MSVS) версии 10 (2010 г.).

### 1) Создание проекта

После запуска среды разработки MSVS откроется Начальная страница, снимок которой приведен на Рисунке 1. С помощью начальной страницы можно ознакомиться с последними новостями ресурса MSDN, ознакомиться со средой MSVS, открыть один из последних ранее созданных проектов, или создать новый проект. Если Вы закрыли Начальную страницу или если она у Вас не отображается, то ее всегда можно открыть с помощью команды меню «Вид-> Начальная страница».

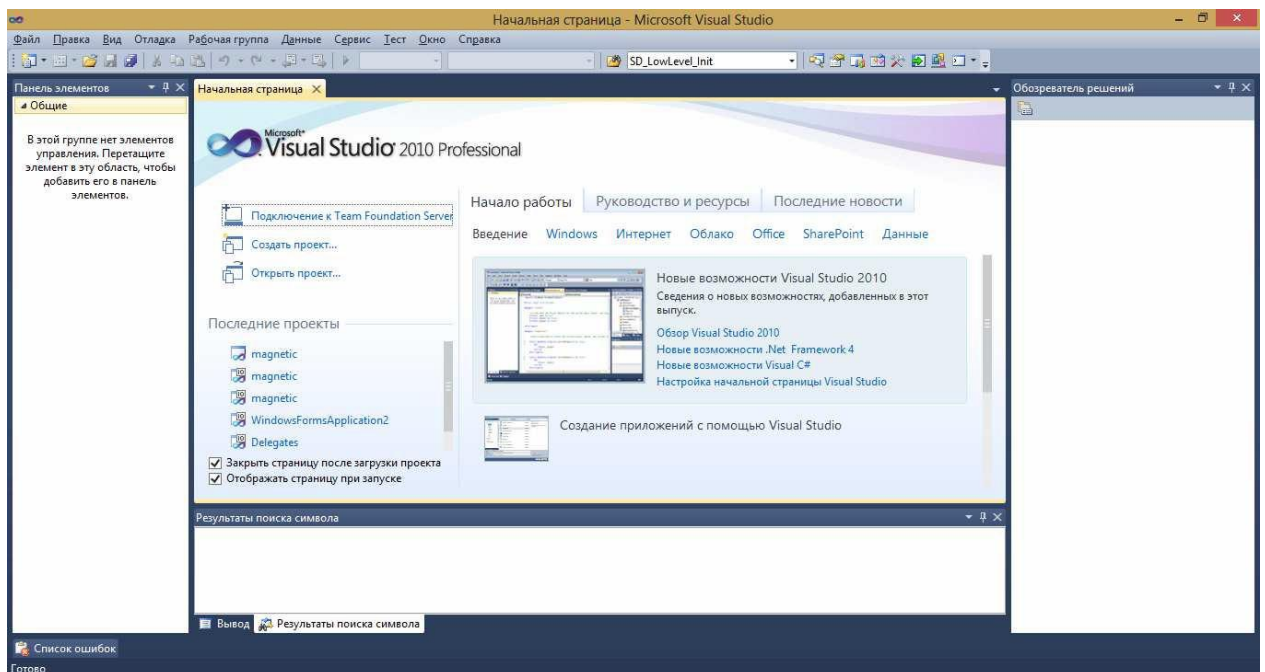


Рисунок 1 – Начальная страница

Для создания первого проекта можно выбрать соответствующий пункт на Начальной странице, либо выбрать пункт меню «Файл-> Создать-> Проект», либо воспользоваться горячими клавишами «Ctrl+Shift+N». После этого появится диалоговое окно, представленное на Рисунке 2.

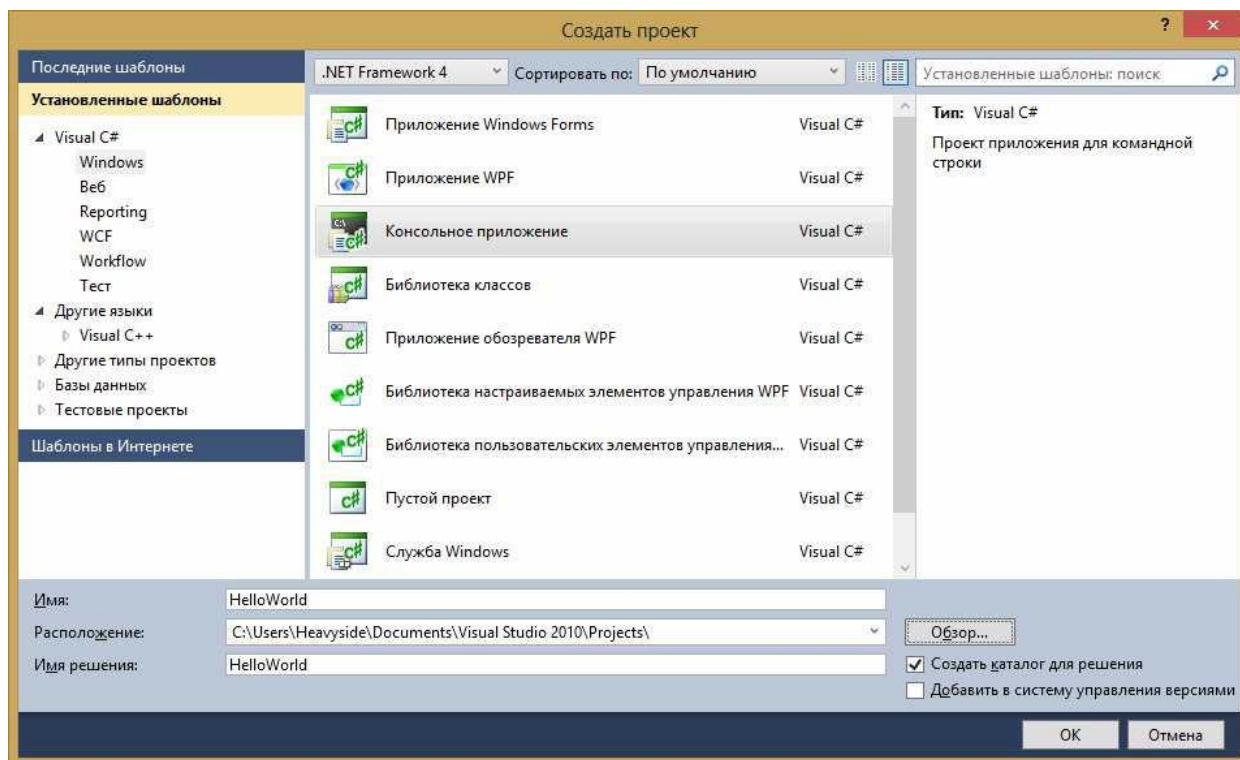


Рисунок 2 – Диалоговое окно «Создать проект»

Проекты в среде Microsoft Visual Studio создаются на базе специальных шаблонов, которые организованы в древовидную структуру, которая отображается слева в окне «Создать проект». Компьютер, на котором производилась подготовка методического материала, был настроен по умолчанию на работу с языком C#, поэтому первым элементов в дереве шаблонов является пункт «Visual C#». Если среда разработки на Вашем компьютере настроена на работу с другим языком программирования, то шаблон для C# можно поискать в дочерних элементах пункта «Другие языки». Если Вам не удалось найти пункт «Visual C#», то Вам необходимо установить дополнительные пакеты для среды разработки.

Для языка C# подготовлено несколько наиболее используемых шаблонов, которые выводятся в списке в центральной части окна «Создать проект»:

- 1) Консольное приложение, предназначенное для работы с командной строкой.
- 2) Приложение Windows Forms, создается оконное Windows-приложение.
- 3) Приложение WPF – создается приложение типа Windows Presentation Foundation.
- 4) Пустой проект.

При выборе шаблона в поле справа выводится его краткое описание. На данном этапе нас интересует консольное приложение, поэтому необходимо выбрать соответствующий шаблон.

В поле «Имя:» вводится название проекта. В поле «Расположение:» указывается каталог, в котором будет создана папка с проектом. Расположение можно указать вручную, либо воспользоваться кнопкой «Обзор». Отметим, что в среде MSVS проекты можно объединять в решения. Мы будем создавать отдельное решение для каждого проекта, поэтому значение поля «Имя решения:» совпадает с значением поля «Имя:». Введите имя проекта, например «HelloWorld» и укажите удобное для Вас расположение. После необходимо нажать кнопку «ОК». Отметим, что инженеры ВолГУ имеют право удалять любые данные с жесткого диска рабочей станции, поэтому, чтобы не потерять свою работу, используйте специально отведенную для студентов папку.

После создания консольного проекта главное окно среды разработки MSVS примет вид, представленный на Рисунке 3. В центре окна располагается главный файл проекта: «Program.cs» (в языке C# не производится разделение файлов на заголовочный и файл реализации, а используется единый файл, имеющий расширение «.cs»). В окне «Обозреватель решений» будет отображаться структура Вашего проекта.

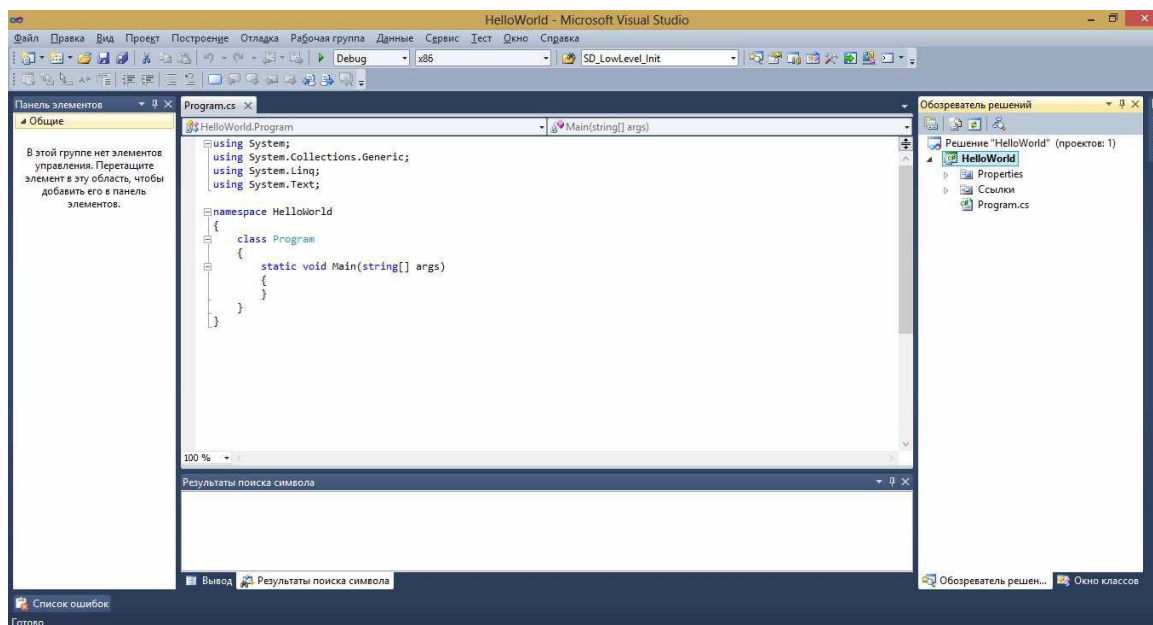


Рисунок 3 – Консольный проект


На данном этапе проект состоит из одного файла. Наберите ниже приведенный листинг в файл «Program.cs»:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!"); Console.ReadKey();
        }
    }
}
```

## 2) Запуск и отладка

Запустить программу на выполнение можно следующими способами:

- 1) Воспользоваться кнопкой  на панели управления
- 2) Выбрав пункт меню «Отладка-> Начать отладку» (F5)
- 3) Выбрав пункт меню «Отладка-> Запуск без отладки» (Ctrl+F5)

После запуска проекта в режиме отладки можно воспользоваться следующими командами:

- 1) Точка останова (F9). Устанавливает или убирает точку останова

отладки в текущей позиции в листинге текущего файла. Точку останова можно установить с помощью мыши, нажав на серую область слева от листинга напротив необходимой строки (рисунок 4).

2) Шаг с заходом (F11). Выполняет следующую команду с заходом в пользовательский метод (методы изложены в Лабораторной работе №2).

3) Шаг с обходом (F10). Выполняет следующую команду без захода в пользовательский метод.

4) Шаг с выходом (Shift + F11). Выходит из пользовательского метода к следующей строки после вызова.

5) Продолжить (F5). Продолжает выполнение программы до следующей точки останова, либо до завершения программы, либо до приостановки программы.

6) Остановить отладку (Shift + F5). Завершает выполнение программы и выходит из режима отладки.

7) Прервать все (Ctrl+Alt+Break). Прерывает выполнение программы и устанавливает курсор в текущую позицию. Помогает определить место «зависания» программы.

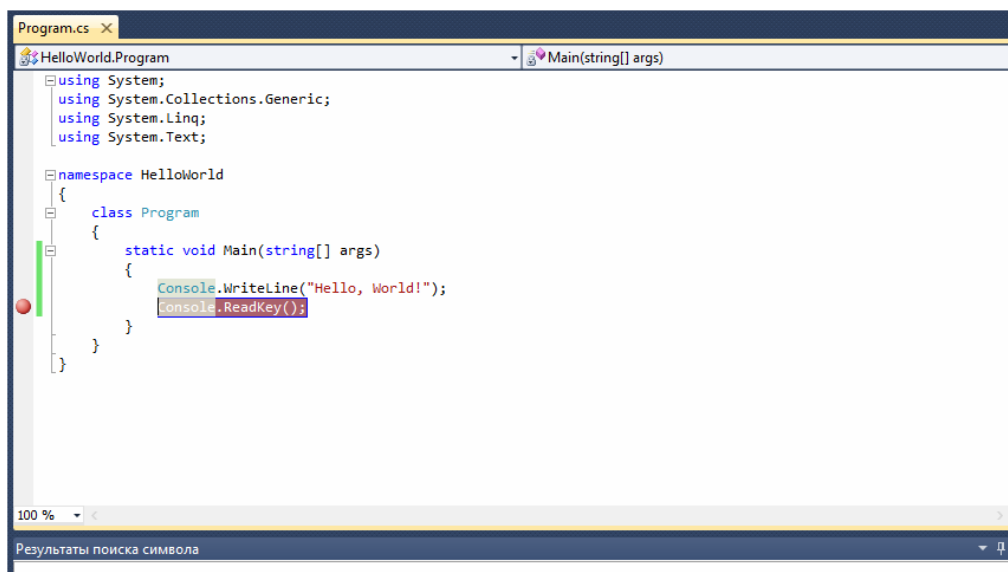


Рисунок 4 – Точка останова

### 3) Приложение Windows Forms

Перейдем к знакомству с основными возможностями среды MSVS для создания оконных Windows приложений. Вызовите диалоговое окно «Создать проект» (Ctrl+Shift+N) и выберете шаблон «Приложение Windows Forms». Назовите проект, например WindowAPP (рисунок 5).

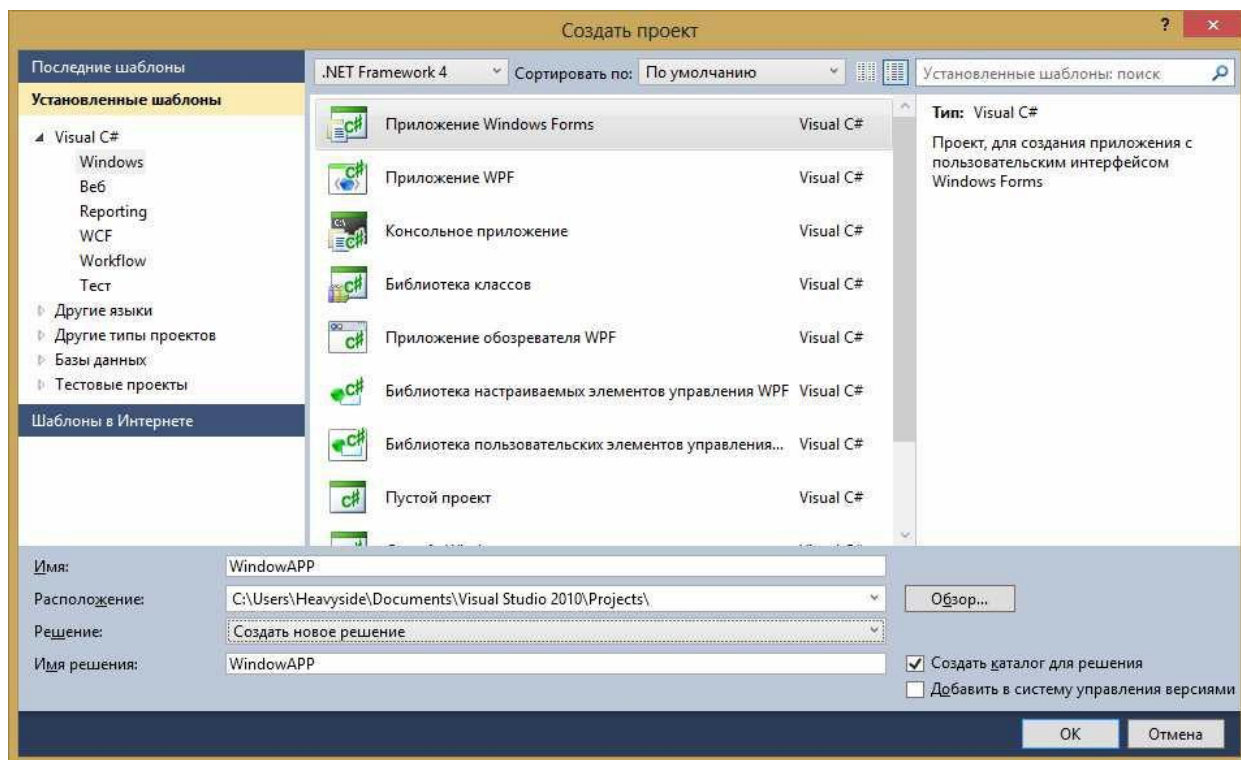


Рисунок 5 – Создание оконного приложения

После создания проекта главное окно среды разработки примет вид, представленный на Рисунке 6. По центру отображается главная форма приложения, которая будет появляться при запуске приложения (Ctrl+F5). В списке справа приведены элементы управления, доступные для размещения на форме. Если у Вас они не отображаются, выполните команду «Вид-> Панель элементов». Для удобства можно вывести все элементы управления, либо определенную группу:

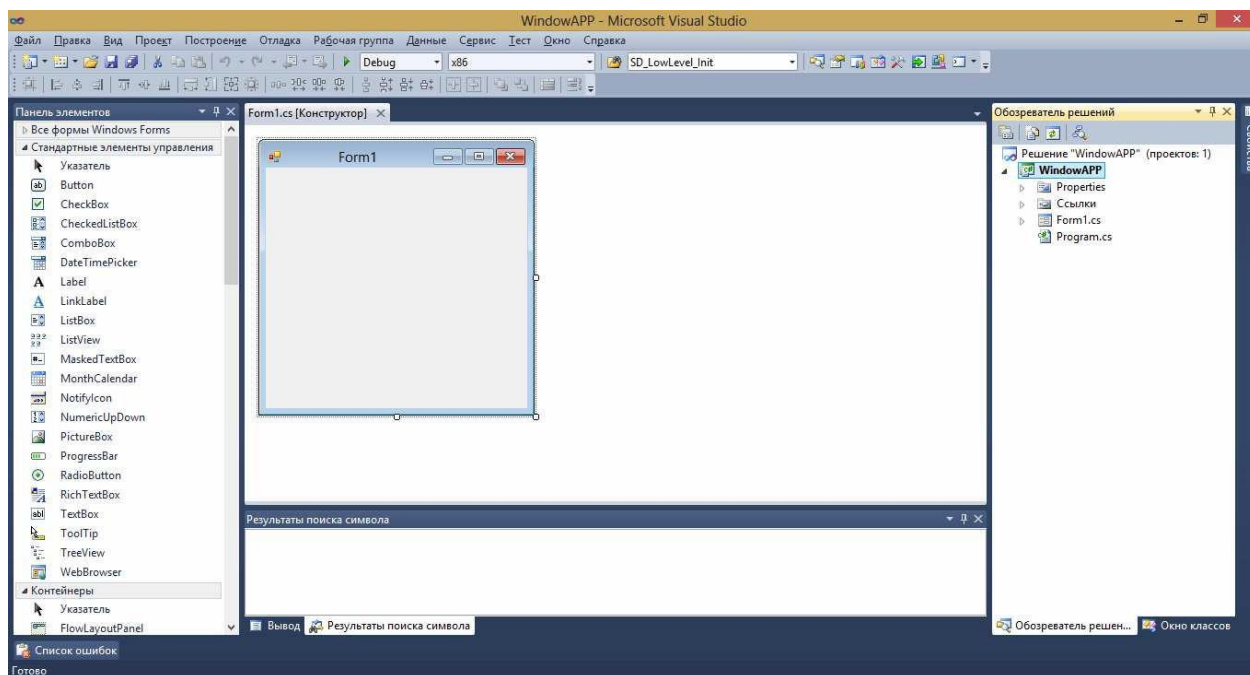


Рисунок 6 – Пустое приложение Windows Forms

- 1) Стандартные элементы управления. Наиболее часто используемые элементы управления, которые можно встретить во всех оконных приложениях.
- 2) Контейнеры. Позволяют объединять различные элементы управления в группы.
- 3) Меню и панели инструментов. Позволяют добавлять на форму главное (располагается в верхней части окна) и контекстное (меню при щелчке правой кнопкой мыши) меню.
- 4) Данные. Содержит компоненты, необходимые для визуализации данных и для работы с БД.
- 5) Компоненты. Здесь располагаются продвинутые компоненты, которые не отображаются на форме, но, в некоторых случаях незаменимы.
- 6) Печать. Содержатся компоненты, необходимые для вывода информации на принтер.
- 7) Диалоговые окна. Располагаются основные окна Windows, такие как открыть файл, сохранить файл, выбор шрифта и т.д.
- 8) Отчеты.

Для помещения какого-либо компонента на форму его достаточно просто

перетащить с помощью мыши либо щелкнуть по нему 2 раза мышью.

## **Button**

Поместите на форму элемент Button (кнопка). Для изменения вида элемента можно воспользоваться окном редактирования свойств элемента управления. Оно обычно отображается справа. Если у Вас оно отсутствует, то выберите пункт меню «Вид-> Окно свойств». Свойства можно либо сгруппировать по функциональному назначению, либо отсортировать в алфавитном порядке. На примере кнопки покажем основные свойства всех элементов управления.

1) Name. Задаёт имя элемента управления. На функциональность элемента имя не влияет, но позволяет отличить элемент управления от других. Имя элемента используется для связи программного кода с элементом.

2) Text. Задаёт текст, отображаемым элементом.

3) TextAlign. Задаёт выравнивание текста в кнопке.

4) Location. Задаёт координаты кнопки относительно родительского элемента управления (формы или контейнера).

5) Size. Задаёт размер элемента управления. Width ширина элемента, Height – высота.

6) Enabled. Задаёт активность элемента (способность генерировать события и реагировать на действия пользователя). True – элемент активен, False – элемент не реагирует.

7) Visible. Задаёт видимость элемента. При значении True элемент отображается, при значении False – невидим.

8) Locked. Фиксирует элемент в контейнере на этапе разработки. Заблокированный элемент нельзя передвигать с помощью мыши и изменять его размер. При этом можно изменять свойства элемента в окне свойств.

## **Label**

Компонент Label позволяет размещать на форме текстовые метки, подсказки для пользователя. Как самостоятельный элемент управления используется для вывода коротких сообщений. Основное применение –



пояснение входных данных для других элементов. Отображаемый текст редактировать нельзя.

### **TextBox**

Компонент TextBox используется для ввода пользовательских текстовых данных. Дополнительные свойства компонента:

1) Lines. Хранит набор строк, отображаемых компонентом. Это свойство полезно, если включен режим Multiline.

2) RightToLeft. Позволяет отображать текст справа налево. Свойство полезно при поддержке мультиязычности.

3) ScrollBars. Позволяет отобразить полосу прокрутки для просмотра больших текстовых сообщений. Можно установить только горизонтальную полосу (Horizontal), только вертикальную (Vertical), обе (Both), либо ни одной (None).

4) Multiline. Позволяет включить режим многострочного ввода.

5) PasswordChar. Позволяет установить символ, отображаемый вместо вводимых символов. При пустом значении поля отображается исходный текст. Используется при вводе паролей.

6) ReadOnly. При установленном значении True запрещает редактирование отображаемого текста.

### **CheckBox**

Разновидность элемента Button. Используется в окнах настройки для выбора нескольких параметров. Свойства:

1) Checked. Указывает, установлен ли флажок.

2) CheckState. Указывает состояние флажка в режиме ThreeState. Unchecked – флажок не установлен, Checked – флажок установлен, Indeterminate – неопределенное состояние.

3) ThreeState. Указывает режим работы компонента. При значении True значение флага обрабатывается свойством CheckState, при значении False значение флага обрабатывается свойством Checked.

## **RadioButton**

Разновидность элемента Button. Используется в окнах настройки для выбора одного из группы параметров. При выборе одного из элементов флаг на другом элементе сбрасывается. Элементы можно группировать, размещая в разных контейнерах. За значение флага отвечает свойство Checked.

## **PictureBox**

Элемент PictureBox используется для отображения изображения из ресурсов приложения, которое задается в свойстве Image.

## **ListBox**

Этот элемент используется для отображения не редактируемого списка данных. Основные свойства:

- 1) Items. Список элементов для отображения.
- 2) Sorted. Позволяет включить режим сортировки отображаемых данных.
- 3) SelectionMode. Позволяет включить режим выбора нескольких элементов списка.

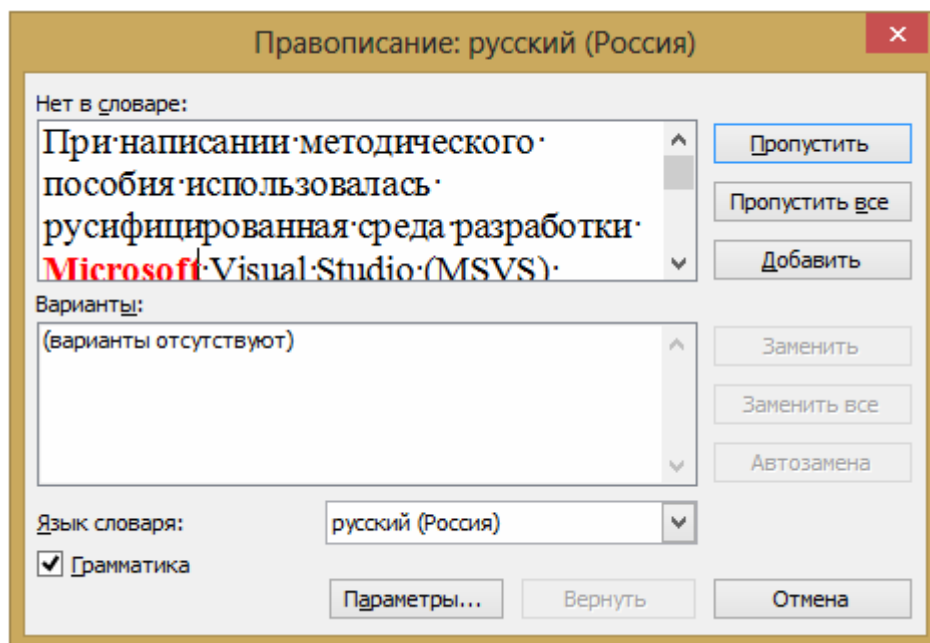
## **ComboBox**

Ниспадающий список. Этот элемент также как и ListBox хранит коллекцию элементов, однако отображает только один из них.

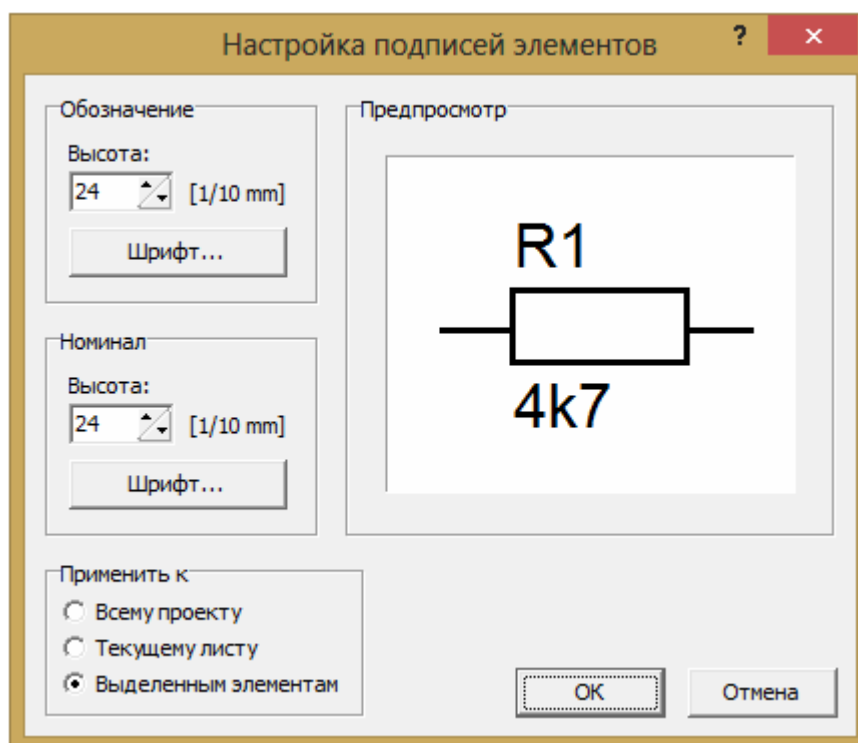
## Задания к лабораторной работе №1

Разработать разметку формы следующего вида:

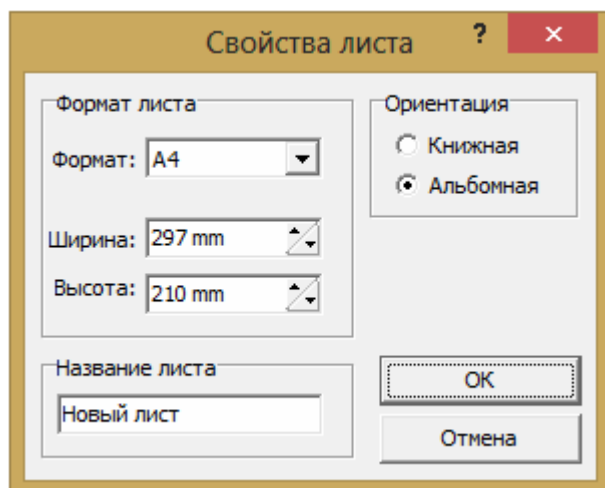
1)



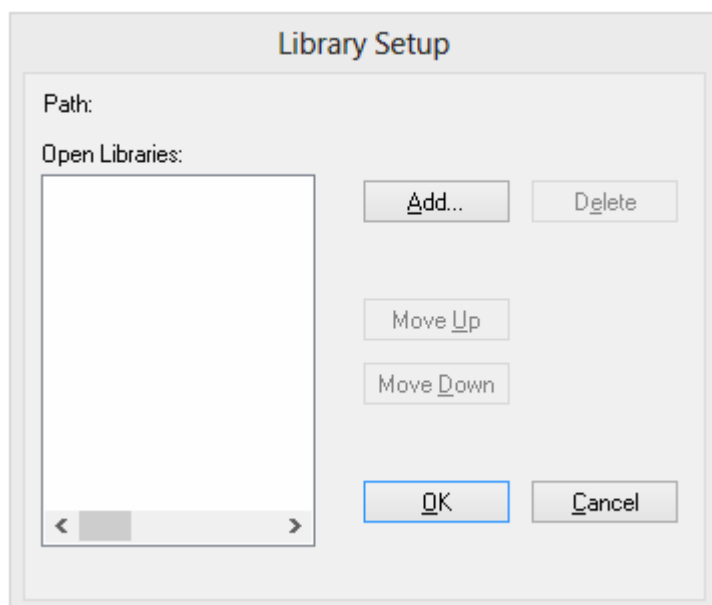
2)



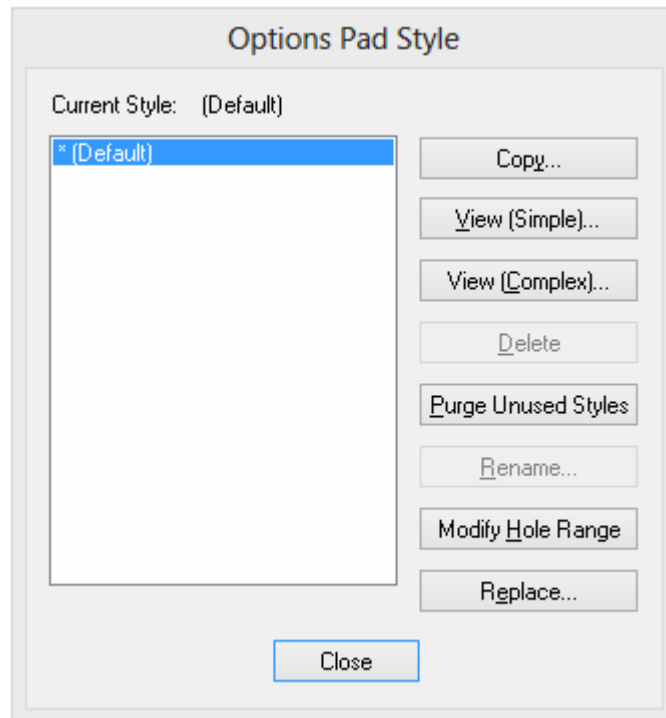
3)



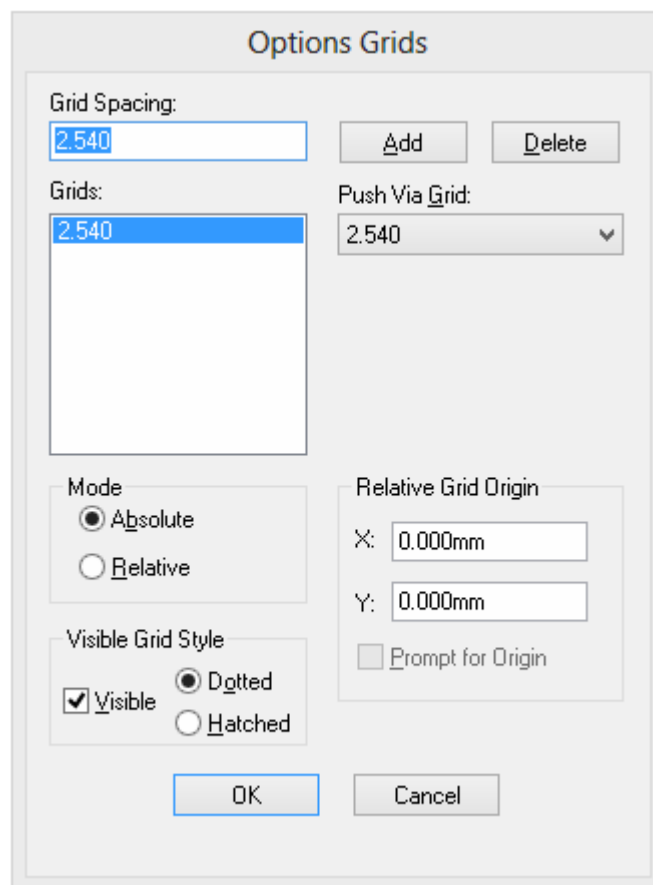
4)



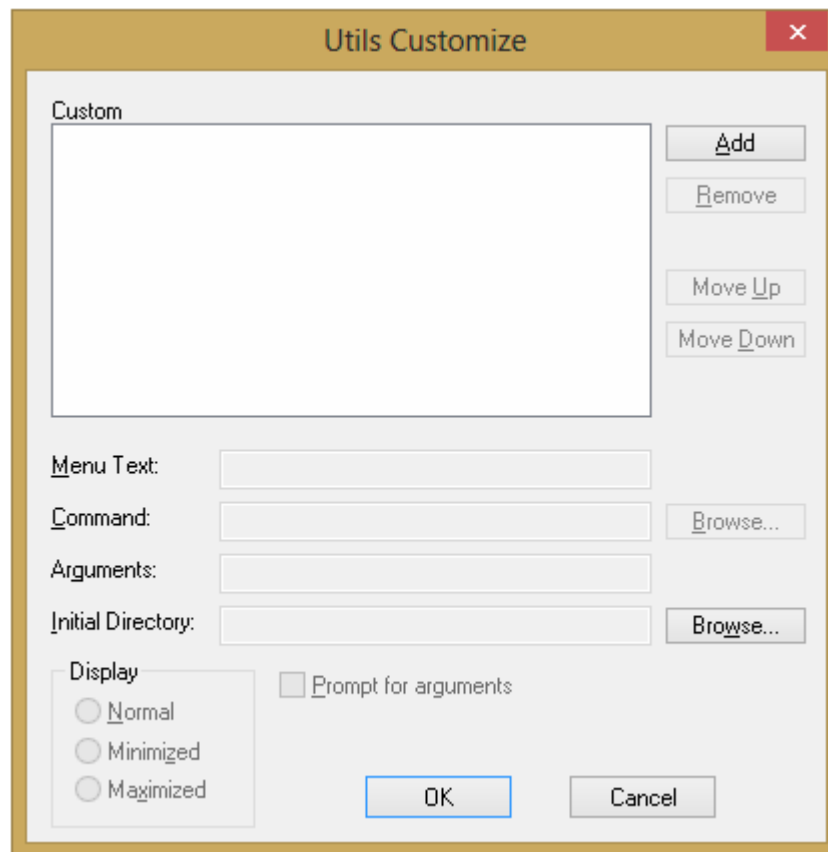
5)



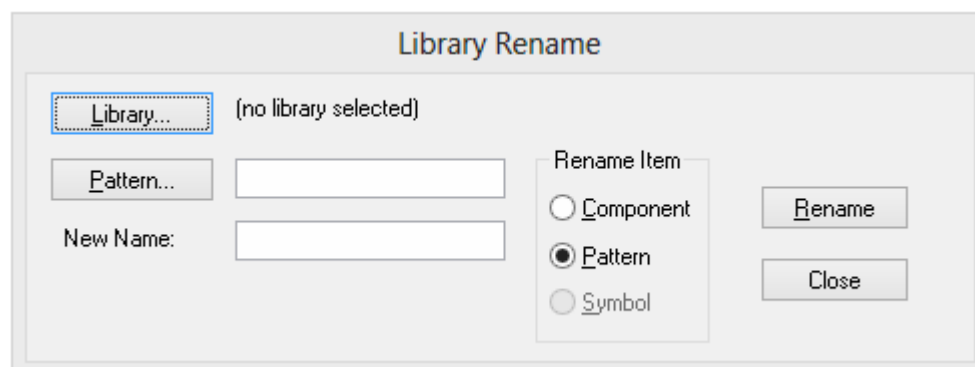
6)



7)



8)



9)

**Place Component**

Component Name:

RefDes:

Value:

Library: 

▼

Pattern:

Pattern Graphics: 

▼

☐ Auto Swap Graphics

10)

**Place Attribute**

Attribute Category:

- All Attributes
- Component
- Net
- Clearance
- Physical
- Electrical
- Placement
- Manufacturing
- Router
- Simulation
- SPECCTRA Route
- SPECCTRA Placement

Name:

[user-defined]

- ANTIPAD\_GAP
- AREA\_AREA\_GAP
- AutoRouteWide
- BoardEdgeClearance
- BURIED\_VIA\_GAP
- Clearance
- COMPONENT\_HEIGHT
- ComponentHeight
- ComponentRotation
- ComponentSpacing
- Description
- DiffPairGap
- DRILL\_GAP
- EXPOSE\_PIN
- HoleToHoleClearance
- InsertionGripLength
- InsertionGripWidth
- InspectionAngle
- JUNCTION\_TYPE

Name:

Value:

☒ Visible

Location

X:  Y:

Text Style: 

(Default) ▼

Rotation:

☐ Flipped

☐ Right Reading

Justification

☐

☐

☐

☐

☐

☐

☒

☐

☐

11)

**Place Dimension**

<b>Styles:</b> Point to Point	<b>Orientation</b> <input checked="" type="radio"/> Horizontal <input type="radio"/> Vertical	<b>Text Style</b> (Default) Text Style...
<b>Units:</b> cm	<b>Text Orientation</b> <input checked="" type="radio"/> Horizontal <input type="radio"/> Vertical	<b>Line Width</b> 0.800mm Line Width...
<b>Layer:</b> Board	<b>Symbol</b> <input checked="" type="radio"/> Text <input type="radio"/> Circle <input type="radio"/> Square <input type="radio"/> Triangle	<b>Arrowhead</b> Length: 4.750mm Width: 1.575mm
<b>Precision:</b> 1 <b>Symbol Size:</b> 500.0mil <b>Center Size:</b> 200.0mil	<b>Tolerance</b> Positive: 0.0000cm Negative: 0.0000cm <input type="checkbox"/> Display Tolerance	<input type="checkbox"/> Display Units <input checked="" type="checkbox"/> Display Diameter Symbol <input type="checkbox"/> Suppress Leading Zeros <input type="checkbox"/> Suppress Trailing Zeros <input checked="" type="checkbox"/> Associated Dimension
OK		Cancel

12)

**Modify Pad Style (Simple)**

Name: 1

<b>Type</b> <input checked="" type="radio"/> Thru <input type="radio"/> Top <input type="radio"/> Bottom	Width: 1.524mm Height: 1.524mm Shape: Ellipse	<b>Hole</b> Diameter: 0.965mm <input checked="" type="checkbox"/> Plated
<b>Plane Connection</b> <input checked="" type="radio"/> Thermal <input type="radio"/> Direct	<b>Plane Swell</b> <input checked="" type="checkbox"/> Use Global Swell Local swell: 0.000mm	OK Cancel



13)

**Drill Symbol Display Configurations**

Drill Symbol Configurations:

Drill Symbol Configuration

Name: DrillSymConfig01

Size: 2.032mm

☒ All Holes In All Ranges  
☐ Specific Hole Range

☐ Plated Holes  
☐ Non-plated Holes  
☒ All Holes

Color:  

☐ Pads Only  
☐ Vias Only  
☒ Pads And Vias

Top  
Bottom

<< Add <<   Modify   Delete

Assign Global Drill Symbols...   OK   Cancel

14)

**Net Classes**

Class Name:

Add   Delete   Rename

Classes:

Select  
Unhighlight  
Highlight

☐ Differential Pair

Nets in this Class:

Net Class Attributes

Edit...

Net Class Layer Attributes

Edit...

Unassigned Nets:

<- Add   Remove ->

Set Nets From Design Selection   Close   Set Nets From Design Selection

## **7) Контрольные вопросы**

- 1) Какие основные типы проектов существуют в Microsoft Visual Studio для языка C#?
- 2) Как установить точку останова?
- 3) Чем отличается шаг с заходом от шага с обходом при отладке?
- 4) Опишите основные свойства элемента управления Button.
- 5) Опишите основные свойства элемента управления TextBox.
- 6) Опишите основные свойства элемента управления CheckBox. Чем отличается CheckBox от RadioButton?
- 7) Опишите предназначение и основные свойства PictureBox.
- 8) Опишите списки ListBox и ComboBox.
- 9) Опишите элемент управления NumericUpDown.