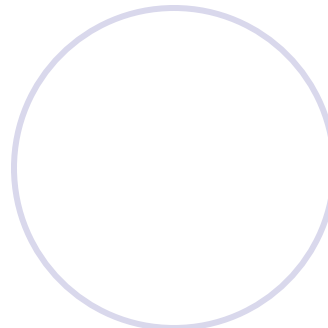
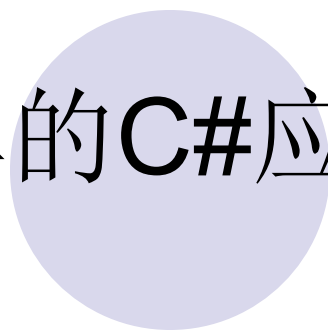
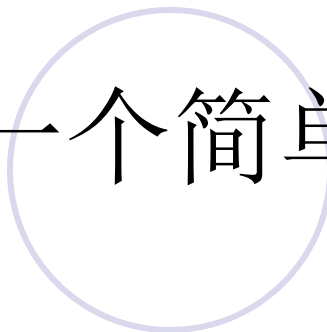
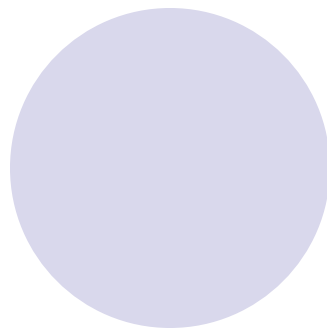
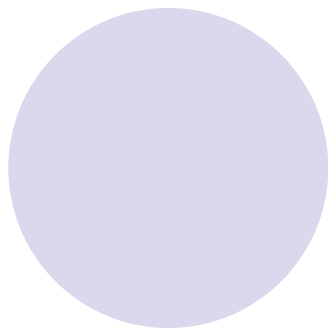
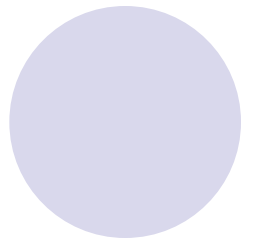
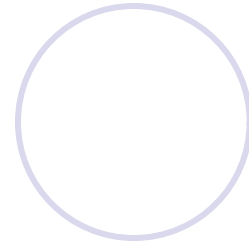
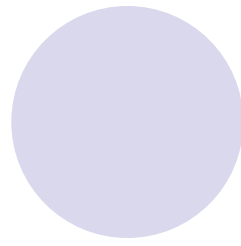
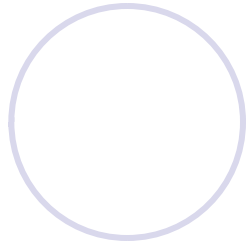
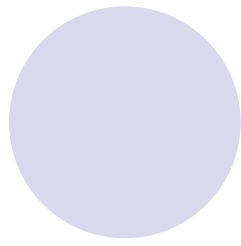


第2章 编制一个简单的C#应用程序

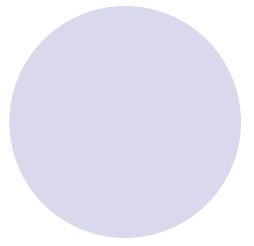
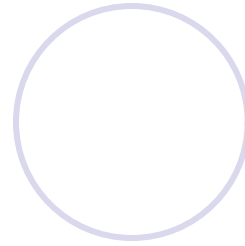
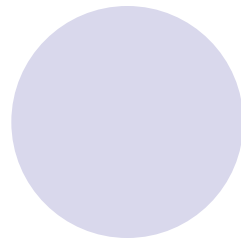
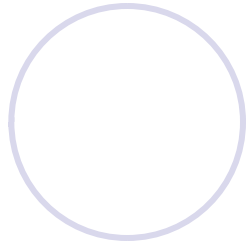
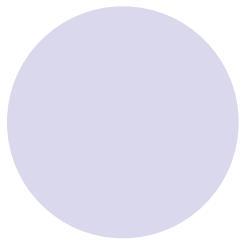


2.1 第一个控制台应用程序设计实例

- 控制台应用程序是利用**C#**语言开发的应用程序类型之一，因为不涉及**Windows**系统的组成元素，因而结构比较简单。下面介绍如何创建控制台应用程序。



- **【例2-1】** 在控制台窗口中输出“Hello world!”字样。
- 具体步骤如下：
- (1) 建立项目
- 先启动Visual Studio.NET。选择“开始”→“程序”→“Visual Studio.NET 2008”→“ Microsoft Visual Studio 2008”菜单项。
- 通常，开始编写一个**C#**应用程序，首先应该建立一个项目。在起始页面的“项目”选项卡中单击“创建项目”按钮，或者通过选择“文件”→“新建”→“项目...”菜单项打开新建项目对话框，如图2.1所示。



- 通常，开始编写一个**C#**应用程序，首先应该建立一个项目。在起始页面的“项目”选项卡中单击“创建项目”按钮，或者通过选择“文件”→“新建”→“项目...”菜单项打开新建项目对话框，如图2.1所示。

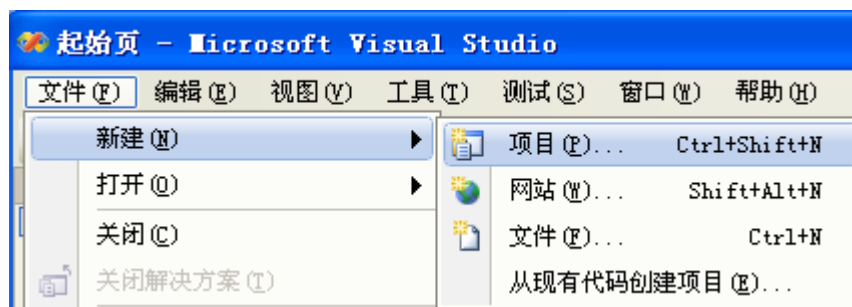
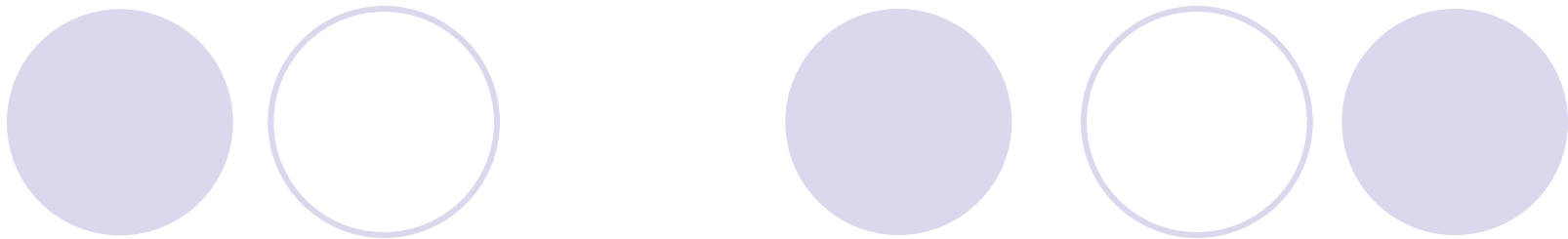


图2.1 用菜单创建项目

- 
- (2) 选择**C#**项目类型
 - 在新建项目对话框的左侧项目类型选项组中选择**Visual C#**。右侧为模板选项组，**Visual Studio**为程序员提供了多种类型的模板，如**Windows**应用程序、类库等。这里选择“控制台应用程序”选项，如图2.2所示。
 - (3) 项目名称
 - 在图2.2所示的新建项目对话框的下方输入项目名称、文件夹位置和解决方案名称。该项目名称为“**welcome**”。
 - 设置完成后，单击“确定”按钮，**Visual Studio**将在指定的文件夹中创建解决方案文件，并在该文件夹下创建以项目名称命名的子文件夹，存储项目文件。

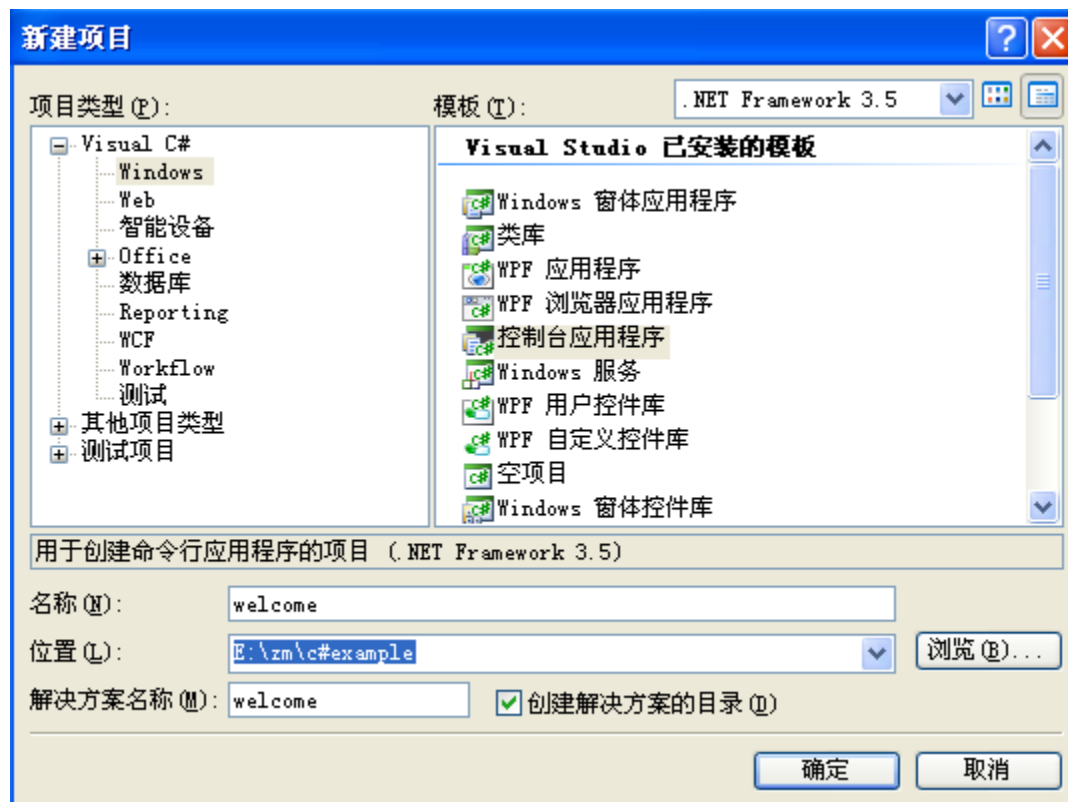
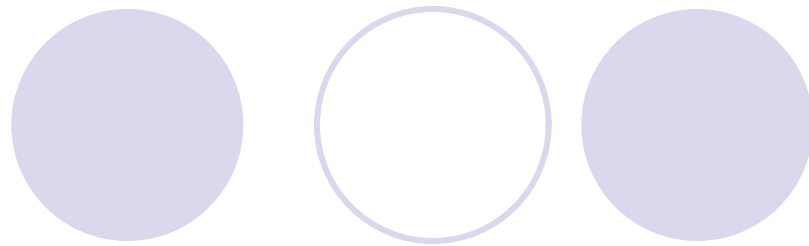
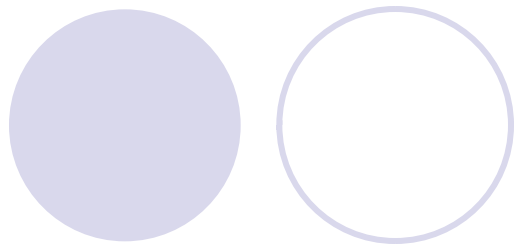


图2.2 新建项目对话框



- (4) 打开项目
- 创建**welcome**项目后，可在起始页上的“最近的项目”列表中看到该项目的名称，如图2.3所示。再次打开**Visual Studio**时，直接单击该链接即可打开此项目。
- 或在起始页中单击打开项目，在打开项目对话框中的查找范围内选择项目所在目录，然后选中要打开的项目文件，如图2.4所示。

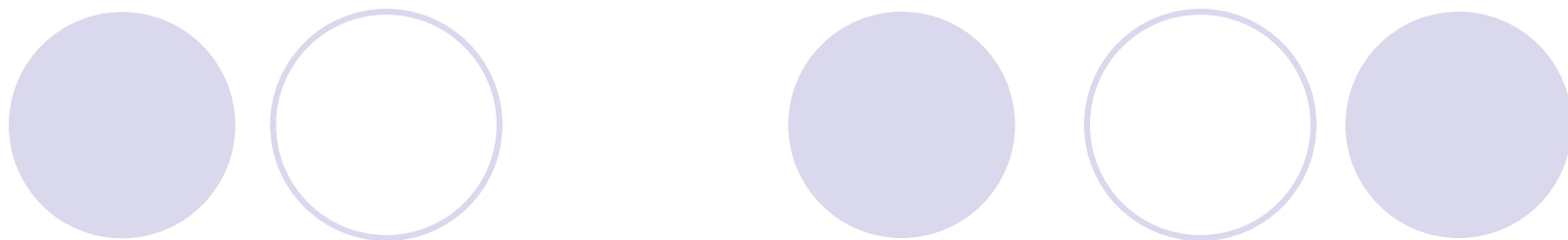


图2.3 起始页窗口

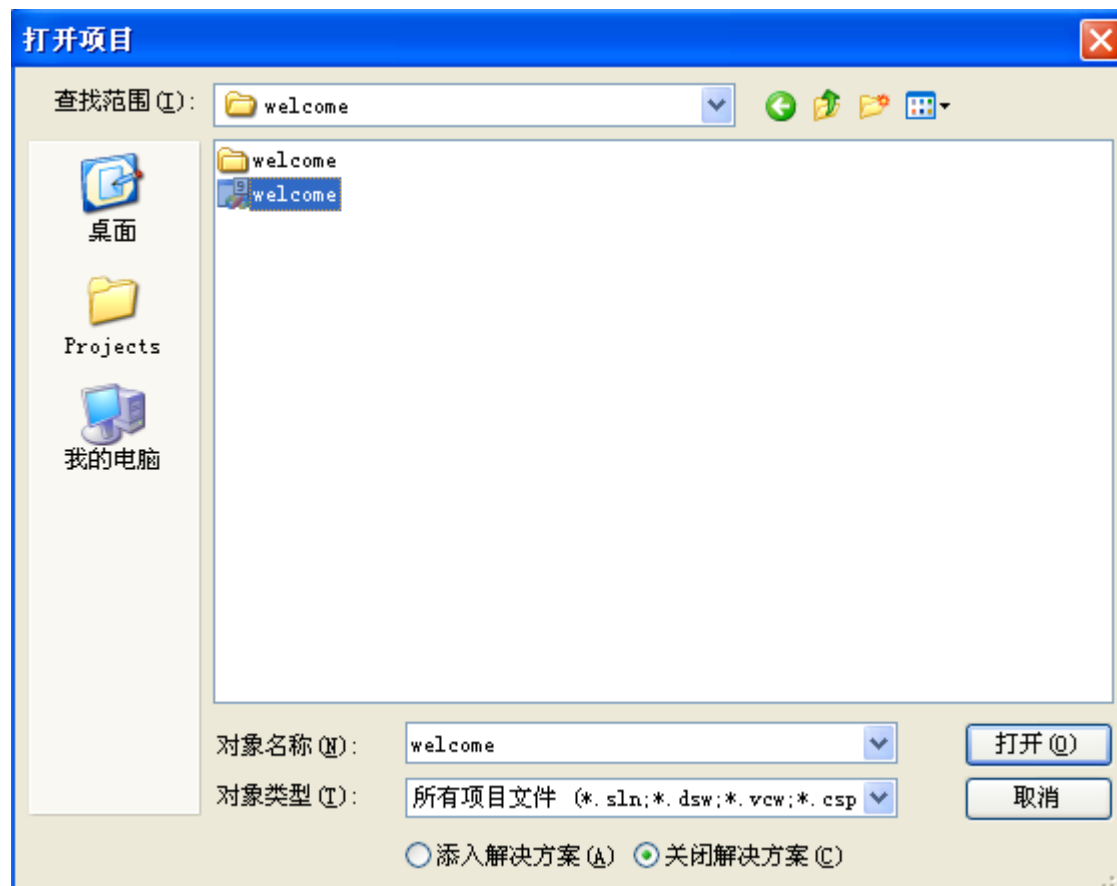
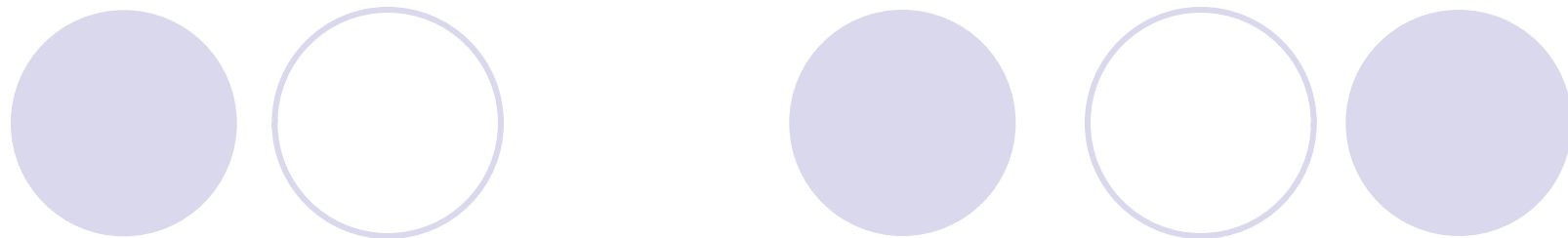
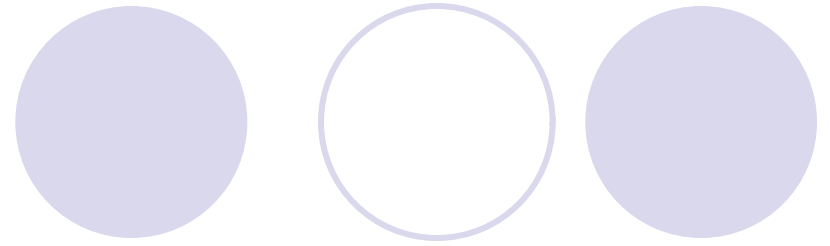
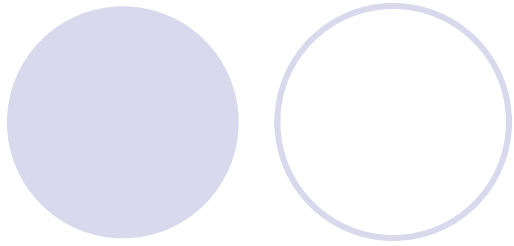


图2.4 打开项目对话框



- (5) 代码编辑器
- 打开项目文件，并进入代码编辑器窗口，如图2.5所示。

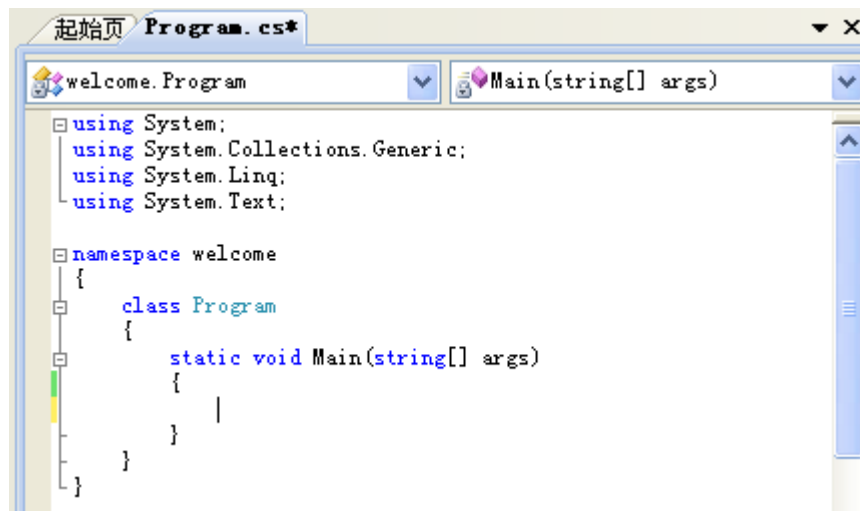
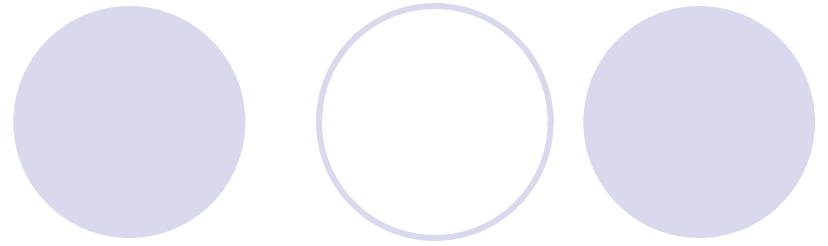
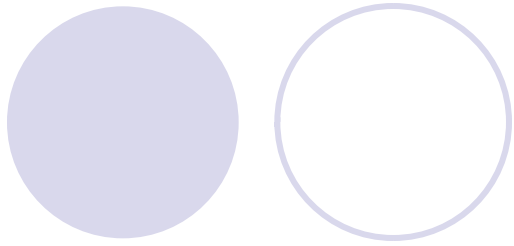


图2.5 代码编辑器窗口

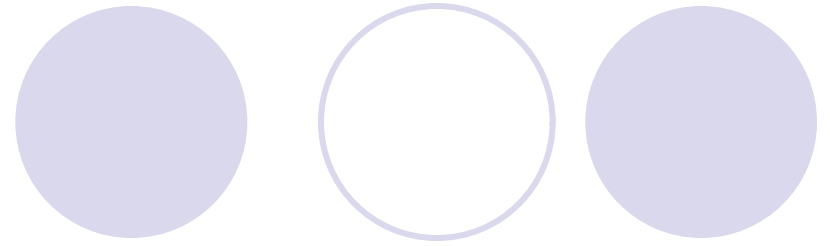
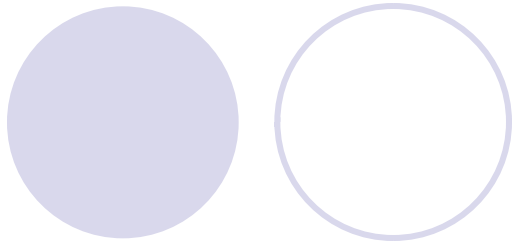
(6) 编写代码

打开**Program .cs**文件，系统自动添加如下代码：

- using System;
- using System.Collections.Generic;
- using System.Linq;
- using System.Text;
- namespace welcome
- {
- class Program
- {
- static void Main(string[] args)
- {
- Console.WriteLine("Hello world");
- }
- }
- }



- (7) 保存项目
- 在“文件”菜单中，单击“全部保存”选项，或在工具栏中单击“保存”按钮。
- (8) 运行程序
- 选择“调试”→“开始执行”菜单项



- 运行结果

```
C:\ file:///E:/zm/c#example  
Hello world
```

2.2 第一个Windows应用程序实例

- **【例2-2】** 创建一个Windows应用程序，在窗口中输出“大家好，欢迎使用C#”字样
- 具体步骤如下：
- (1) 运行Visual Studio 2008，创建一个Windows应用程序项目，在新建项目的模板中选中Windows窗体应用程序，在“名称”文本框中将项目改名为exwin-01，在“位置”文本框中输入项目保存的目录位置，本例保存在本地硬盘的“E:\ZM\c#example”文件夹，选中“创建解决方案的目录”复选框，然后单击“确定”按钮，如图2.9所示。
- (2) 在“解决方案资源管理器”中，单击Form1.cs，重命名Form1.cs为FormMain.cs (见图2.10)



图2.9 创建Windows应用程序的新建项目界面

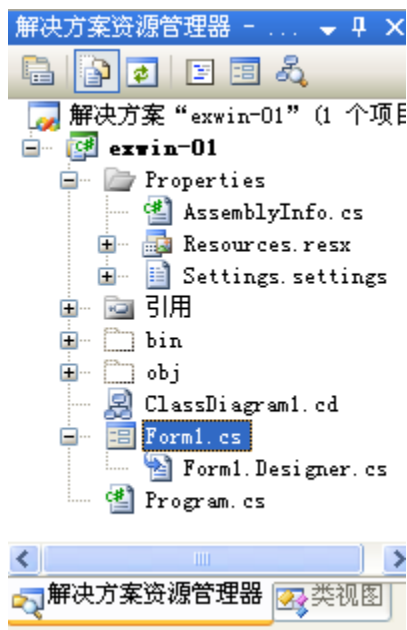
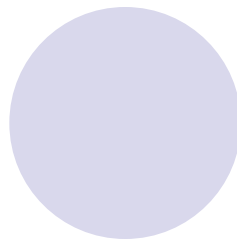
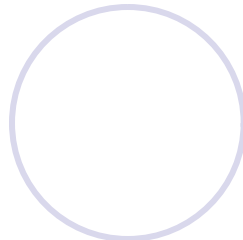
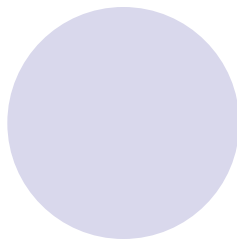
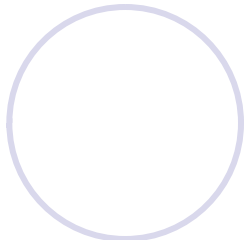
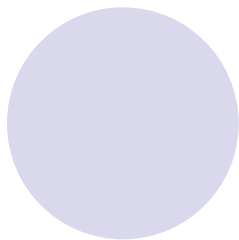
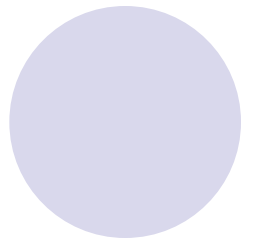
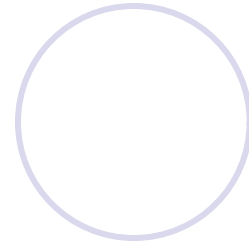
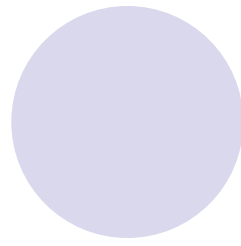
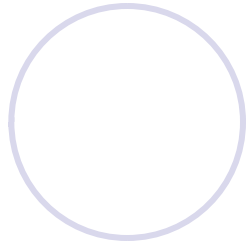
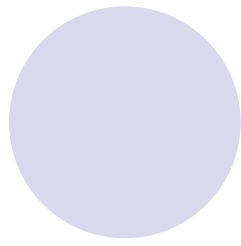


图2.10 Form1.cs重命名



- 在窗体设计器中选中窗体，然后在属性栏中分别修改窗体的**Text**属性、**Size**属性和**FormBorderStyle**属性，如图2.11所示为修改窗体的**Text**属性值，观察修改后窗体显示的效果。

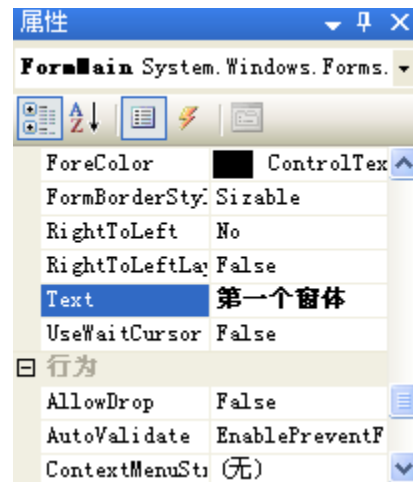
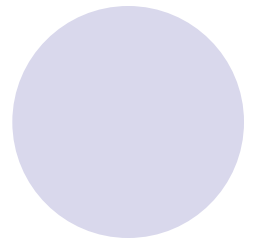
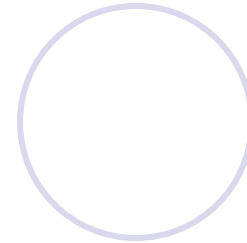
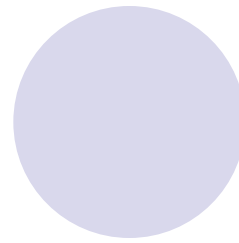
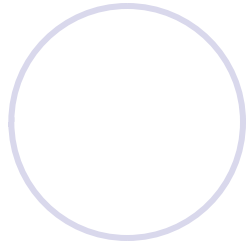
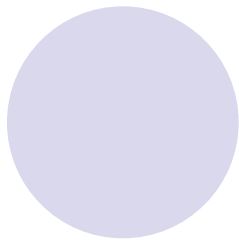
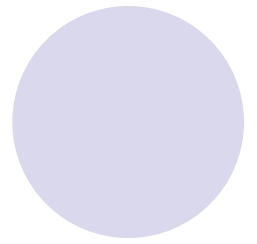
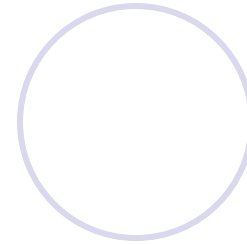
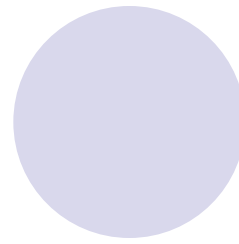
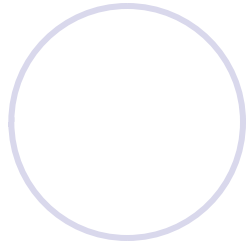
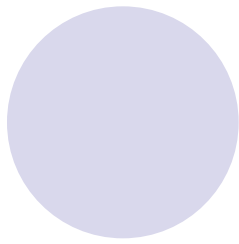
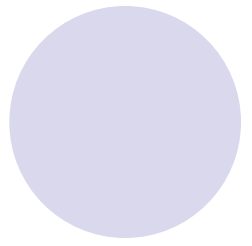
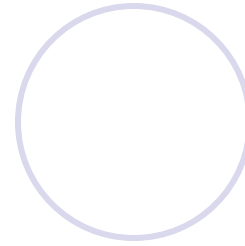
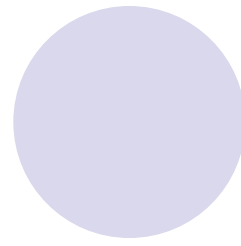
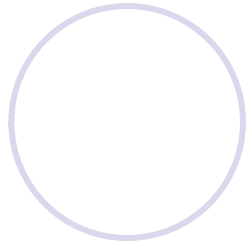
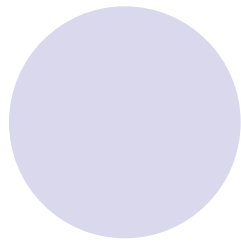


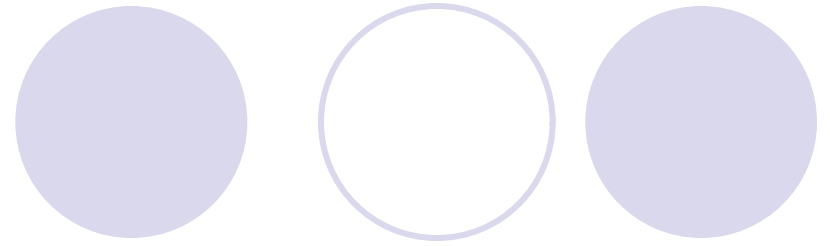
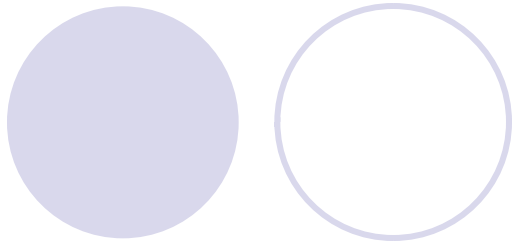
图2.11 修改窗体属性的值



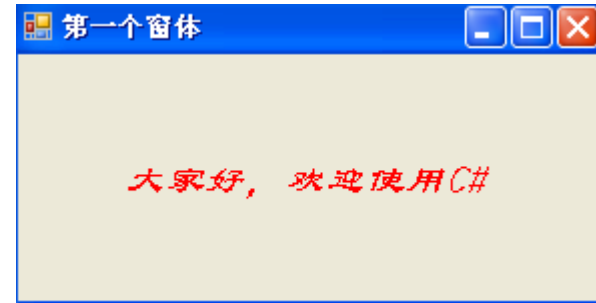
- (3) 从工具箱中向设计窗体拖放一个**Label**控件。选中该控件，然后修改其**Text**属性为“大家好，欢迎使用**C#**”。



- 选择**Label1**对象，然后设置其**Font**属性为“隶书，5号字，斜体”，选择**ForeColor**属性为红色，再选择**TextAlign**属性值，观察显示效果，并思考如何使某行的信息居中显示。修改**AutoSize**属性为“**False**”，拖动该控件右下角改变其大小，观察变化。

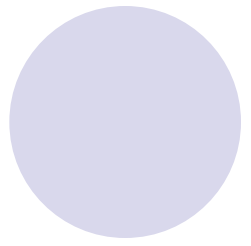
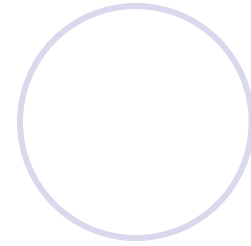
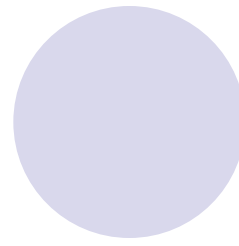
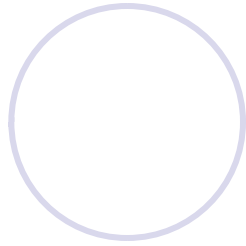
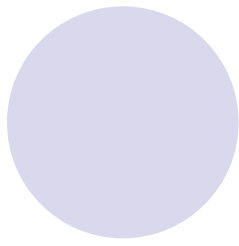


- 按**F5**运行程序，运行界面如图所示。可以看到，没有编写一句代码，就可以完成这样一个功能。



2.3 C#项目和解决方案

- 项目是一组要编译到单个程序集(在某些情况下是单个模块)中的源文件和资源。例如，项目可以是类库，或一个**Windows GUI**应用程序。解决方案是构成某个软件包(应用程序)的所有项目集。
- 一个解决方案中可以有多多个项目，但一个解决方案中最多只有一个项目是含有**Main**方法的，而其他项目中可以有一些类、方法、接口等的定义，是为其他项目服务的。

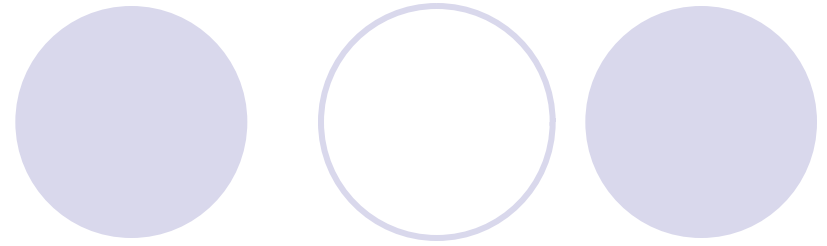
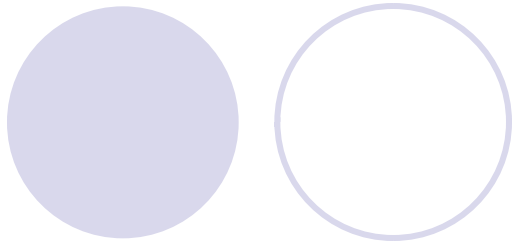


- 例如，其中可能有一个用户界面，有某些定制控件和其他组件，它们都作为应用程序的库文件一起发布。不同的管理员甚至有不同的用户界面。每个应用程序的不同部分都包含在单独的程序集中，因此，在**Visual Studio.NET**看来，它们都是独立的项目。可以同时编写这些项目，使它们彼此连接起来。可以把它们当做一个单元来编辑。**Visual Studio.NET**把所有的项目看做一个解决方案，把该解决方案当做是可以读入的单元，并允许用户在其上工作

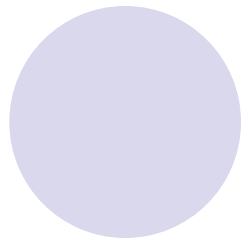
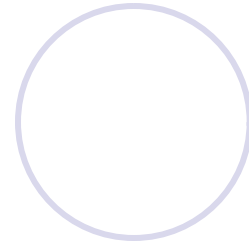
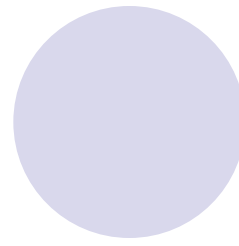
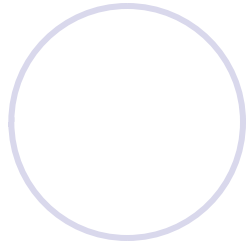
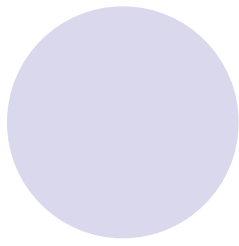
2.4 应用程序起始点

- 前面简单介绍了一个创建**Windows**应用程序实例。利用这个程序了解一下应用程序的起始点。
- 在“解决方案资源管理器”窗口中双击**Program.cs**节点，打开**Program.cs**文件。该文件是“**exwin-01**”应用程序的入口点，因为它包含了**Main()**方法，内容如下：

- 1.using System;
空间 //引用System命名
- 2.using System.Collections.Generic;
- 3.using System.Linq;
- 4.using System.Windows.Forms;
- 5.
- 6.namespace exwin_01 //创建exwin_01命名空间
- 7.{
- 8. static class Program //创建静态类
- 9. {
- 10. /// <summary>
- 11. /// 应用程序的主入口点。
- 12. /// </summary>



- 13. [STAThread]
- 14. static void Main() //创建Main静态方法，即应用程序的入口点
- 15. {
- 16. Application.EnableVisualStyles(); //启用可视化样式
- 17. Application.SetCompatibleTextRenderingDefault(false);
- //设置呈现格式
- 18. Application.Run(new FormMain ()); //运行FormMain窗体
- 19. }
- 20. }
- 21. }

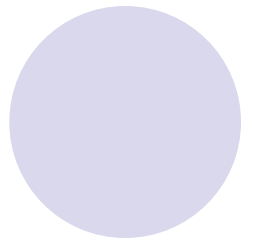
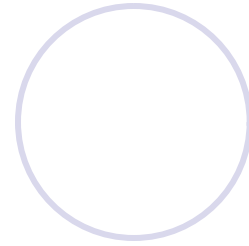
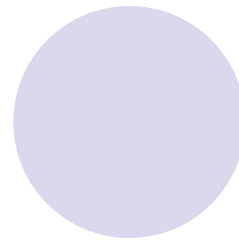
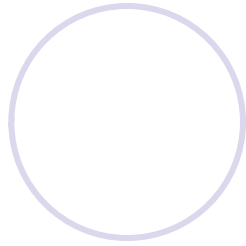
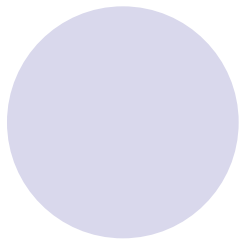


- 对程序的进一步说明：
- 第1~4行代码分别使用**using**关键字引用了4个命名空间(分别是**System**、**System.Collections.Generic**、**System.Linq**、**System.Windows.Forms**)。
- 第14行代码在类**Program**中定义了**Main**方法，该方法为应用程序的入口点，即应用程序运行后，调用的第一个方法。
- 第16行代码调用**EnableVisualStyles()**方法，启用可视化样式。
- 第17行代码调用**SetCompatibleTextRenderingDefault(false)**方法，设置呈现格式。
- 第18行代码调用**Run()**方法，运行**FormMain**窗体。

2.5 配置程序集信息

- 在.NET中使用的程序集包括完全自我说明的描述程序集的数据，这样当程序执行时不需要到其他目录或注册表中查询包含在程序集中对象的信息。
- 在“解决方案资源管理器”窗口中的单击**Properties**节点，然后单击**AssemblyInfo.cs**节点，打开**AssemblyInfo.cs**文件。该文件用来配置exwin-01应用程序的程序集信息，内容如下：

- 1.using System.Reflection;
- 2.using System.Runtime.CompilerServices;
- 3.using System.Runtime.InteropServices;
- 4.// 有关程序集的常规信息通过下列属性集
- 5.// 控制。更改这些属性值可修改
- 6.// 与程序集关联的信息
- 7.[assembly: AssemblyTitle("exwin-01")] //设置程序集的标题
- 8.[assembly: AssemblyDescription("")] //设置程序集的简介
- 9.[assembly: AssemblyConfiguration("")] //设置程序集的配置
- 10.[assembly: AssemblyCompany("AAA")] //设置程序集
的公司
- 11.[assembly: AssemblyProduct("exwin-01")] //设置程序集
的产品
- 12.[assembly: AssemblyCopyright("Copyright © AAA 2011")]//设置程
序集的版权
- 13.[assembly: AssemblyTrademark("")] //设置程序集的商标
- 14.[assembly: AssemblyCulture("")] //设置程序集的文化
-



- 说明：
- 第1～3行使用**using**关键字引用了3个命名空间。
- 第7～14行分别配置程序集的信息，如标题、公司名称、文化环境等