

第1章 C#概述及Visual Studio.NET集成开发环境(IDE)

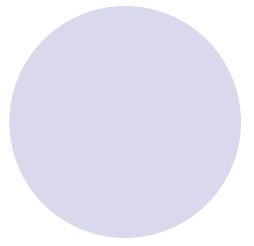
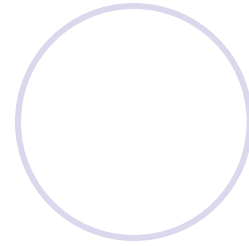
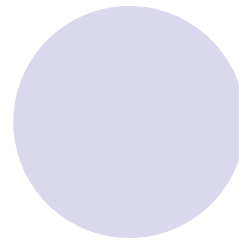
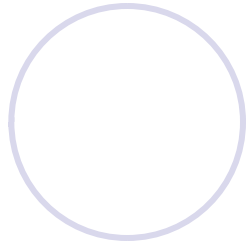
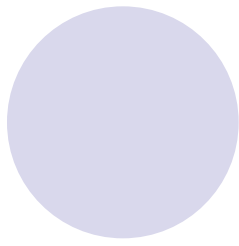
- 1.1 .NET框架概述
- 1.1.1 .NET的定义
- .NET技术是微软公司推出的一个全新概念，“它代表了一个集合、一个环境和一个可以作为平台支持下一代Internet的可编程结构”。
- 通过.NET这个平台，可以使用多种语言开发同一个项目，实现这些语言的跨平台应用。而微软默认的语言是Visual C#.NET

1.1.2 .NET开发平台

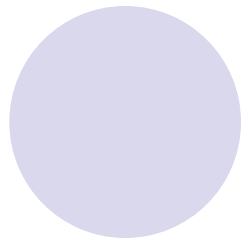
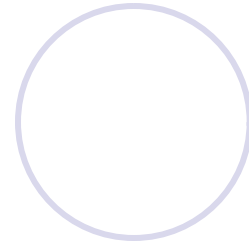
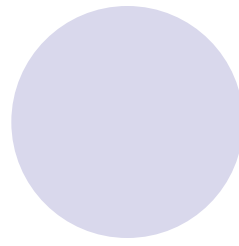
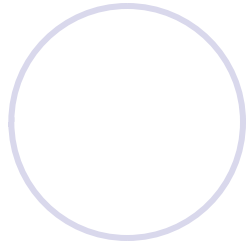
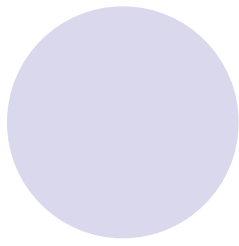
- .NET开发平台包括.NET框架和.NET开发工具等组成部分，.NET框架(Framework)是整个开发平台的基础，包括公共语言运行库和框架类库，.NET开发工具包括Visual Studio.NET集成开发环境和.NET编程语言。

1.1.3 .NET Framework 结构

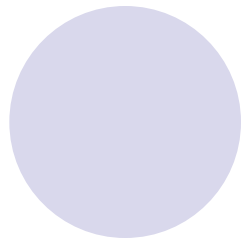
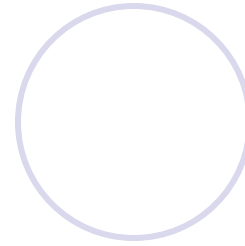
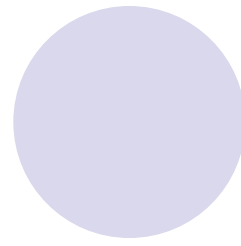
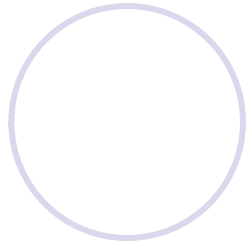
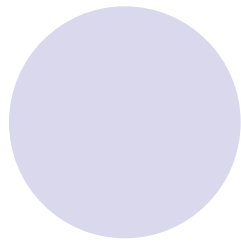
- .NET Framework(框架)的组成：公共语言运行时环境(CLR)；基类库(BCL)；基于ASP.NET编程框架的网络服务和网络表单；Windows桌面应用界面编程组件；数据库访问组件。
- 它的版本有：
 - 1.x：已用于Visual Studio 2003；
 - 2.0：已用于Visual Studio 2005；
 - 3.5：已用于Visual Studio 2008；
 - 4.0：已用于Visual Studio 2010。



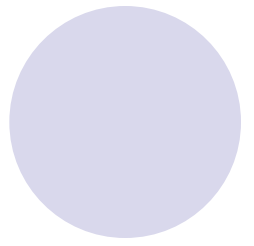
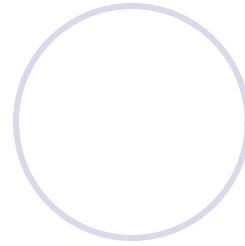
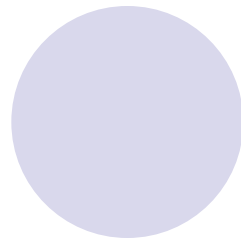
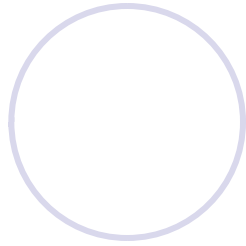
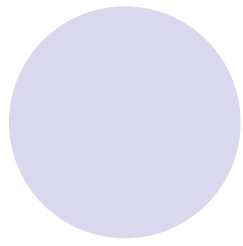
- **.NET Framework**的两个主要组件，公共语言运行时环境和**.NET Framework**类库



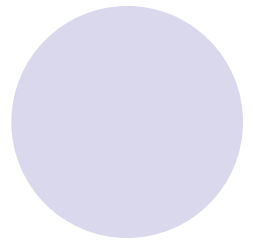
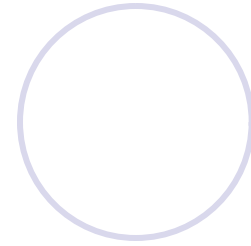
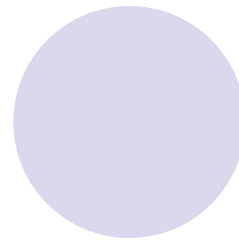
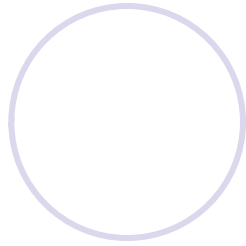
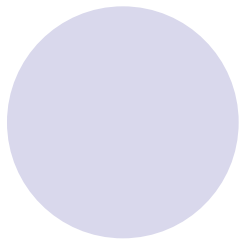
- (1) 公共语言运行时环境
- .NET框架的底层是公共语言运行时环境，提供了程序代码可以跨平台执行的机制，另外还提供了内存管理、线程管理、远程处理等核心服务，并且还强制实施严格的类型安全检查操作，从而提高代码的安全性、可靠性和准确性。它提供了系统资源统一管理和安全机制。其中，以公共语言运行库为目标的代码为“托管代码”，而不以公共语言运行库为目标的代码称为“非托管代码”



- (2) .NET Framework类库
- ADO.NET、Windows Forms和ASP.NET等组件库不是C#或Visual Basic.NET语言的一部分，而是进行相应的应用软件开发所必须掌握的。
- 所有.NET开发语言的能力和执行速度基本相同；
- 只要遵守公共语言规范(Common Language Specification, CLS)和支持公共语言运行时(Common Language Runtime, CLR)，可以采用其他语言；



- **.NET**语言生成的都是托管代码，这种代码在运行库的控制之下运行；
- **.NET**编译生成的可执行文件是一种托管代码，称为微软中间语言(Microsoft Intermediate Language, MSIL)，可以“一次编写，随处运行”，前提是有框架的支持。



- 1.1.4 Microsoft Visual Studio.NET(VS.NET)集成开发环境(IDE)
- Microsoft Visual Studio.NET(VS.NET)是一套完整的开发工具，是一个集成开发环境(IDE)，用于生成Windows应用程序、ASP.NET Web应用程序、XML Web Services和移动应用程序。其中包含的Visual Basic.NET、Visual C++.NET、Visual C#.NET等，全都使用相同的集成开发环境，该环境允许它们共享工具，并有助于创建混合语言解决方案。

1.2 C#语言概述

● 1.2.1 C#语言简介

- **C#**起源于C语言家族，因此具有**C++**的功能。**C#**采用与**C**、**C++**或**Java**一致的花括号({})语法，简单易学。此外，**C#**语法简化了**C++**的诸多复杂性，同时又提供了**Java**所不具备的很多强大的功能，例如，可为**null**的值类型、枚举、委托、**lambda**表达式和直接内存访问。
- **C#**语言是随**.NET**一起设计出来的全新的开发语言，其设计目的就是作为**VS.NET**的核心语言。经过几年的发展，**C#**已经成为**Windows**平台上软件开发的主流语言之一。

1.2.2 C#语言特点

- C#是一种现代的、面向对象的、类型安全的编程语言。C#具有下列特点：
- (1) 简单。
- (2) 面向对象
- (3) 类型安全
- (4) 现代。
- (5) 兼容性。

1.2.3 C#语言开发应用范围

- 桌面应用。包括控制台应用程序、Windows窗体应用程序等。
- Web应用。包括ASP.NET应用程序、Web服务等。
- Windows Presentation Foundation(WPF)应用程序。
- 面向服务的应用程序(Windows Communication Foundation, WCF)。
- 工作流程的应用程序(Windows Workflow Foundation, WWF)。
- 移动应用。
- Office平台应用程序。
- Windows服务。

1.3 配置Visual C#开发环境

- 1.3.1 Visual Studio 2008 的安装



图1.2 Visual Studio 2008的安装界面



图1.3 安装向导

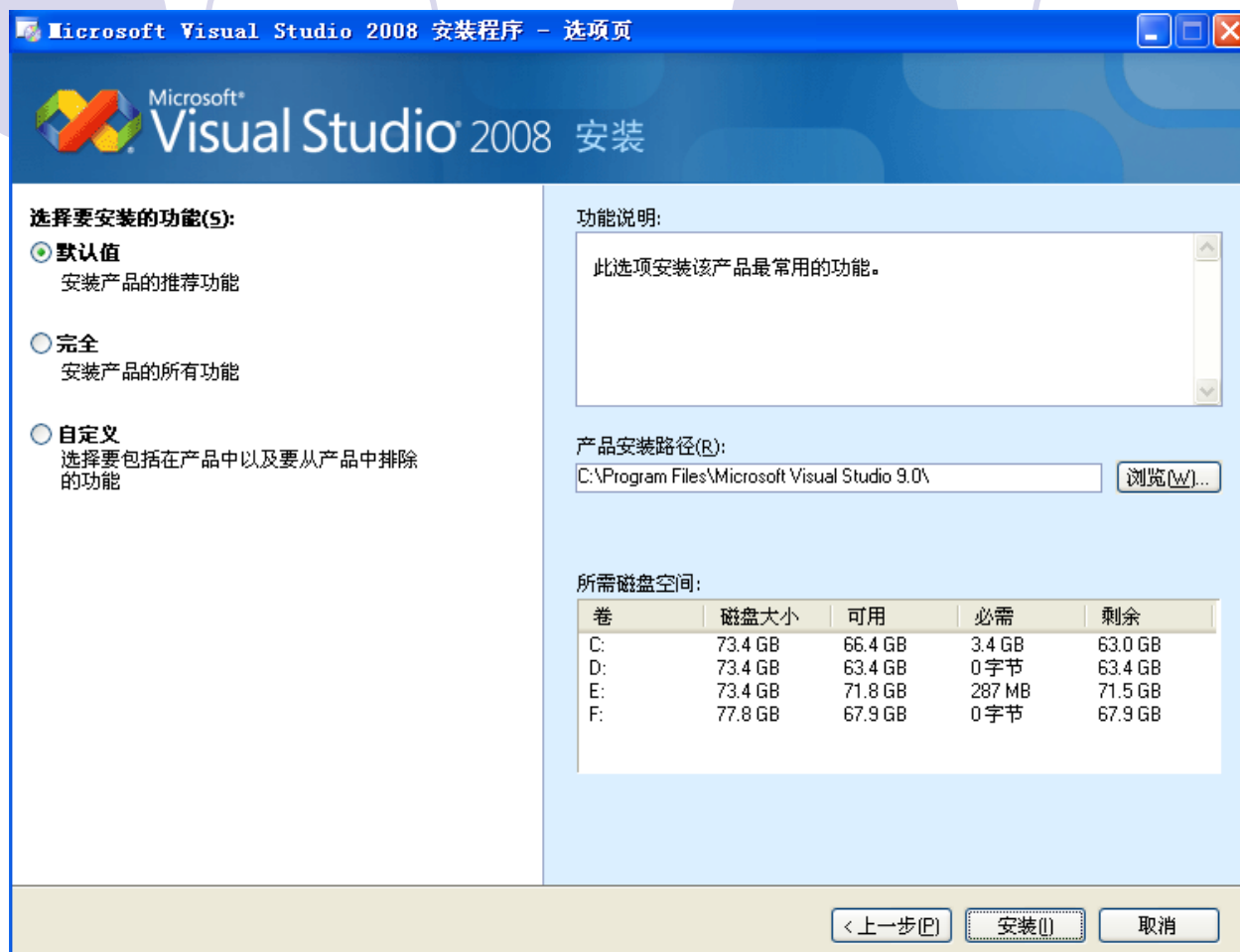


图1.4 “选择要安装的功能”及“产品安装路径”

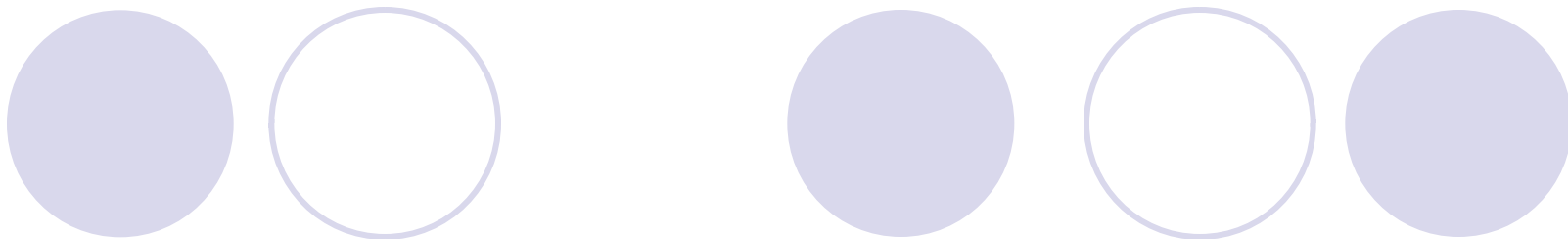


图1.5 选择要安装的功能

1.3.2 C#的启动

- 安装完Microsoft Visual Studio 2008后，选择“开始”→“程序”→“Microsoft Visual Studio 2008”→“ Microsoft Visual Studio 2008”，Microsoft Visual Studio 2008就启动了，如图1.6所示

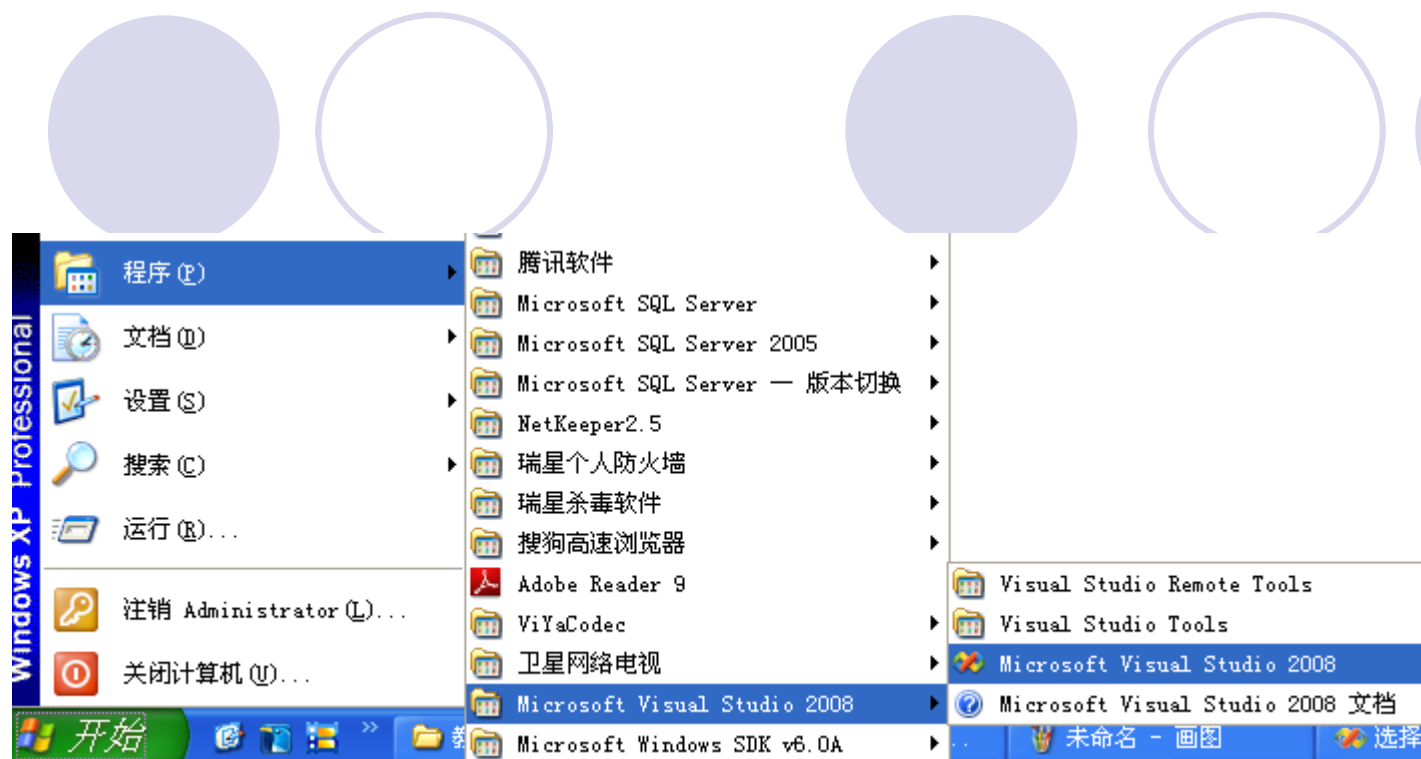
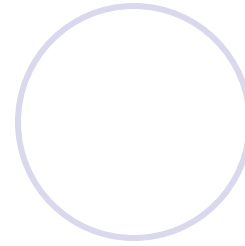
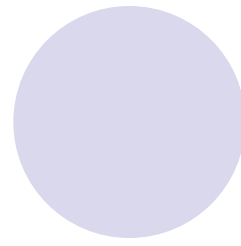
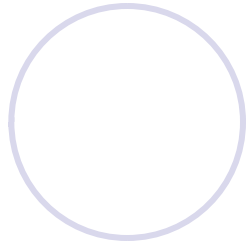
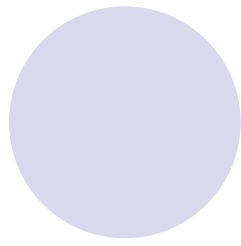


图1.6 开始菜单下Microsoft Visual Studio 2008启动界面



- 第一次启动Microsoft Visual Studio 2008会出现如图1.7所示“选择默认环境设置”界面。在该页面中，选择“Visual C#开发设置”选项，并单击“启动Visual Studio”按钮，进入起始页，如图1.8所示。

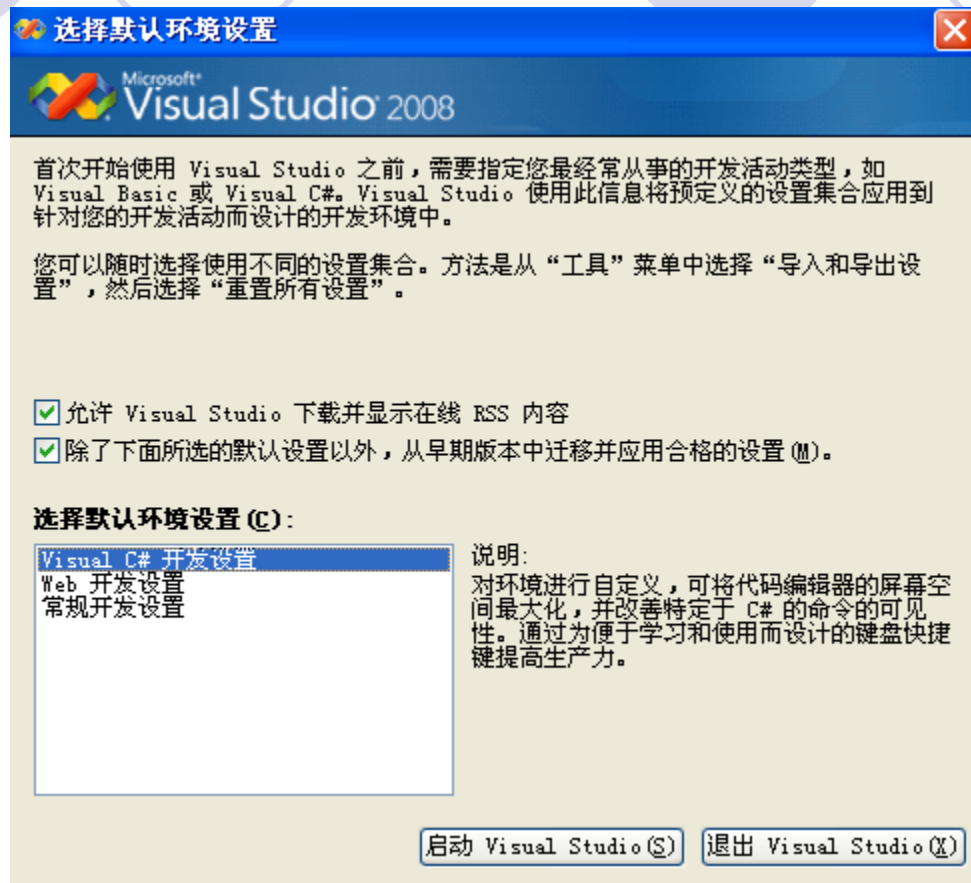


图1.7 选择默认环境设置界面

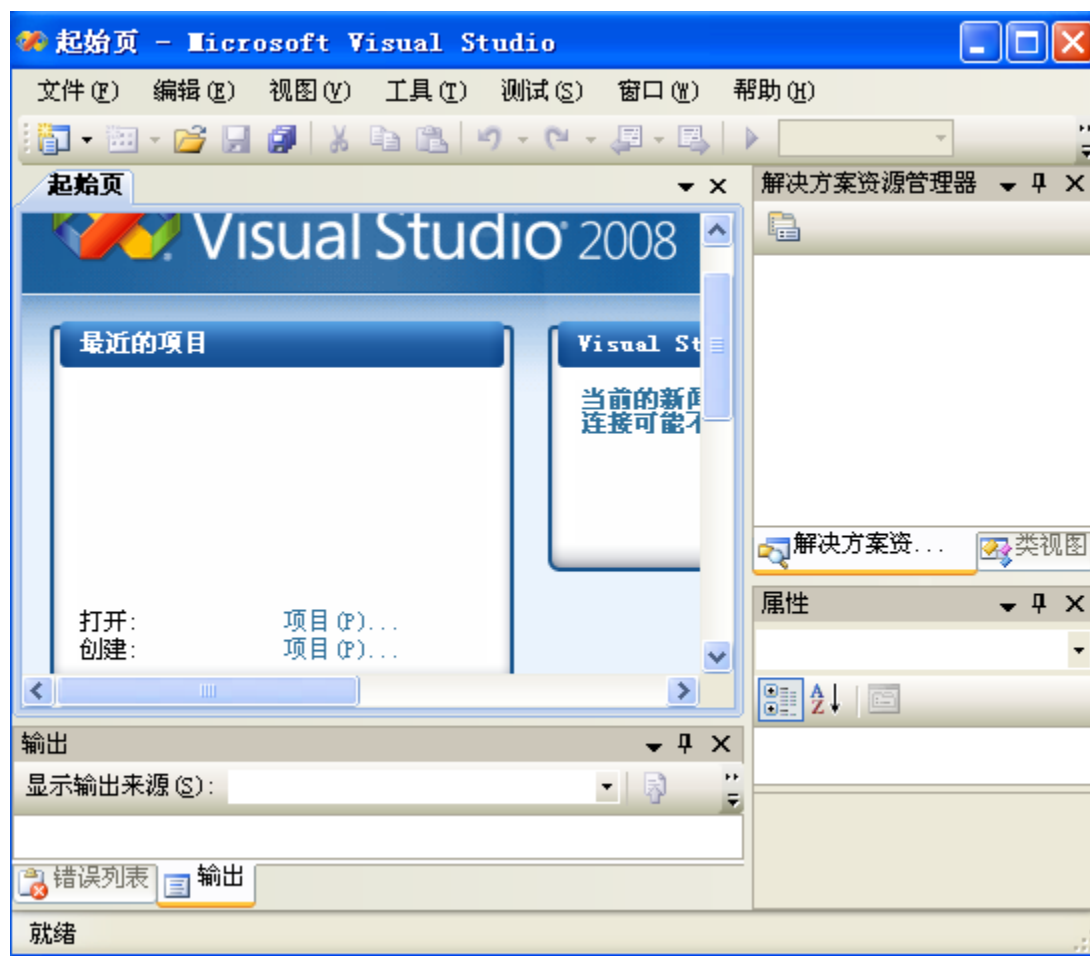


图1.8 起始页界面

1.3.3 起始页

- 起始页是打开**IDE**的第一个窗口。每次启动**IDE**时，该页作为默认页显示。
- 若想打开某个已经存在的项目，单击图1.8左下侧的打开项目。打开项目的另一方法：选择“文件”→“打开”→“项目/解决方案(P)...”菜单项，进入“打开项目”对话框。选中某项目后，单击“打开”按钮，就打开了某一项目。
- 若要创建一个新项目，则单击“创建项目”选项，进入“新建项目”对话框。新建项目另一方法：选择“文件”→“新建”→“项目”菜单项，如图1.9所示，也可以进入“新建项目”对话框，如图1.10所示。

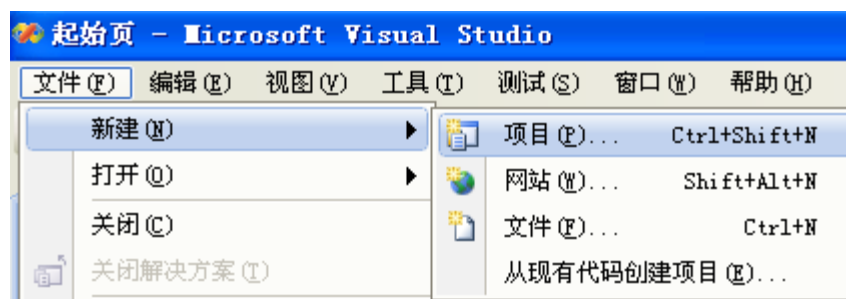


图1.9 用菜单方法新建项目

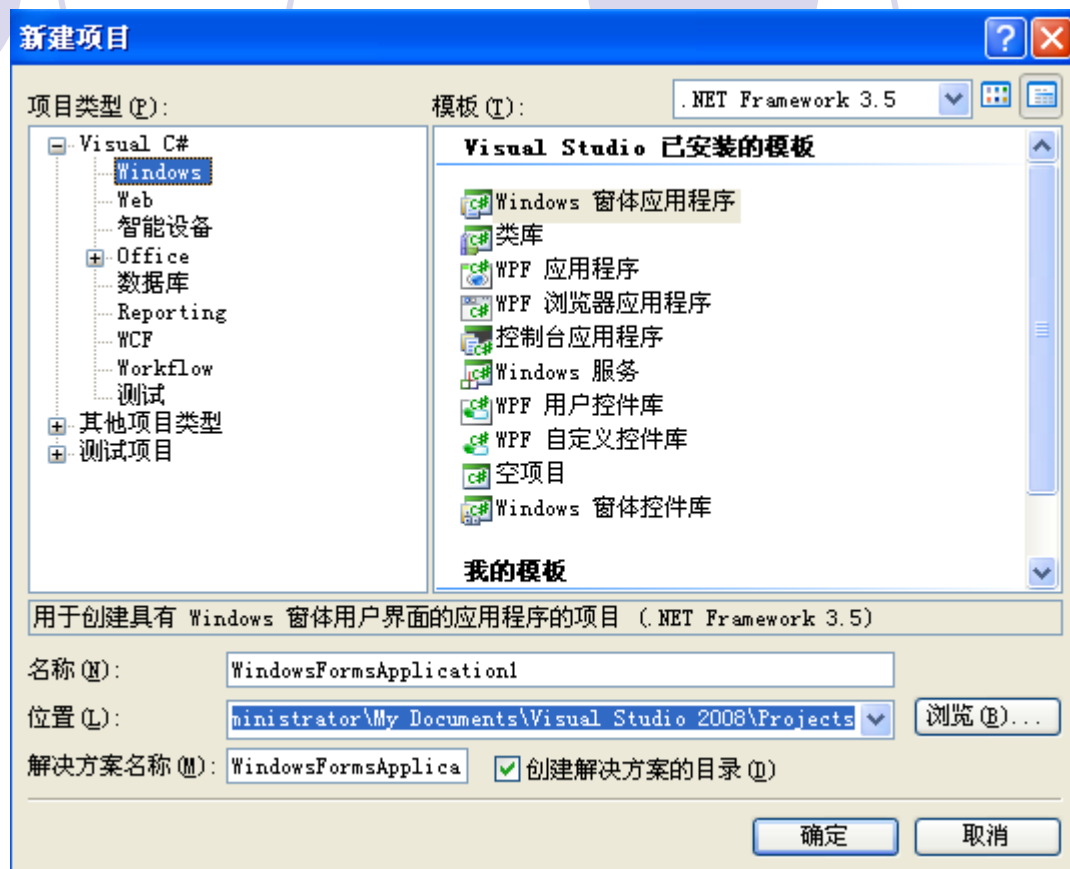
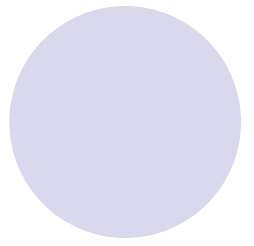
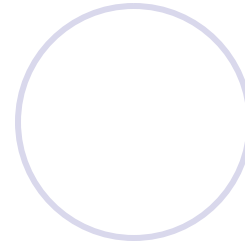
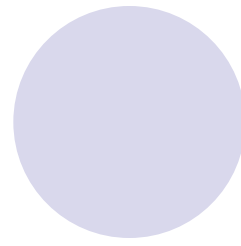
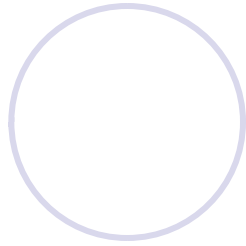
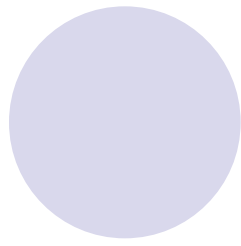


图1.10 “新建项目”对话框



- 在“新建项目”对话框中，选中所创建项目的模板，输入项目名称及存储项目的位置等，再单击“确定”按钮，就创建好了一个空的项目。

1.3.4 Visual C#的退出

- 选择“文件”→“退出”菜单项，就可退出Visual C#环境，如图1.11所示。

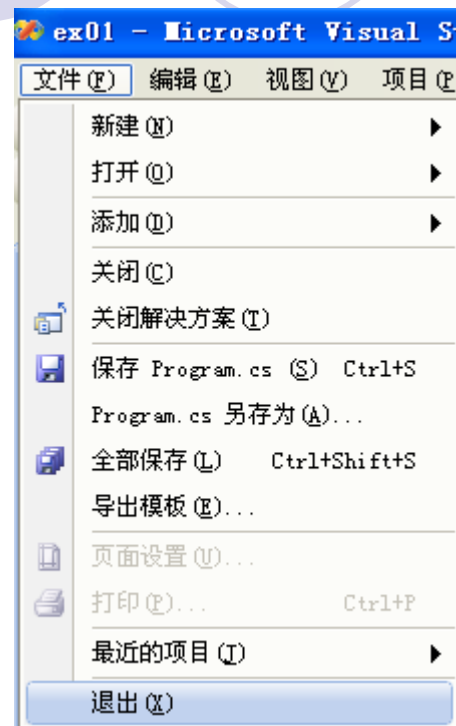
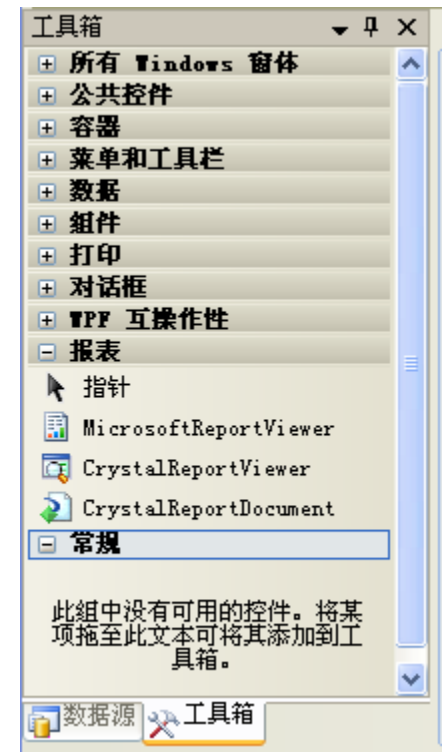


图1.11 退出Visual C#环境

1.3.5 Visual C#菜单栏、工具栏、工具箱



1.3.6 解决方案资源管理器

- 当创建项目后，会在“解决方案资源管理器”中显示自动生成的项目，如图1.14所示。解决方案中包含一个或多个项目，每个项目都对应于软件中的一个模块。在解决方案资源管理器中，将同类的文件放在一个目录下，当单击这个目录后，会将对应的目录下的文件全部显示出来。例如，单击“引用”文件夹前面的“+”，就会将引用的程序集全部显示出来，同时“+”变成“-”。
- 用鼠标右键单击“解决方案资源管理器”中的节点，将弹出一个上下文菜单，通过其中的菜单命令可以对节点对象进行操作。

1.3.7 类视图

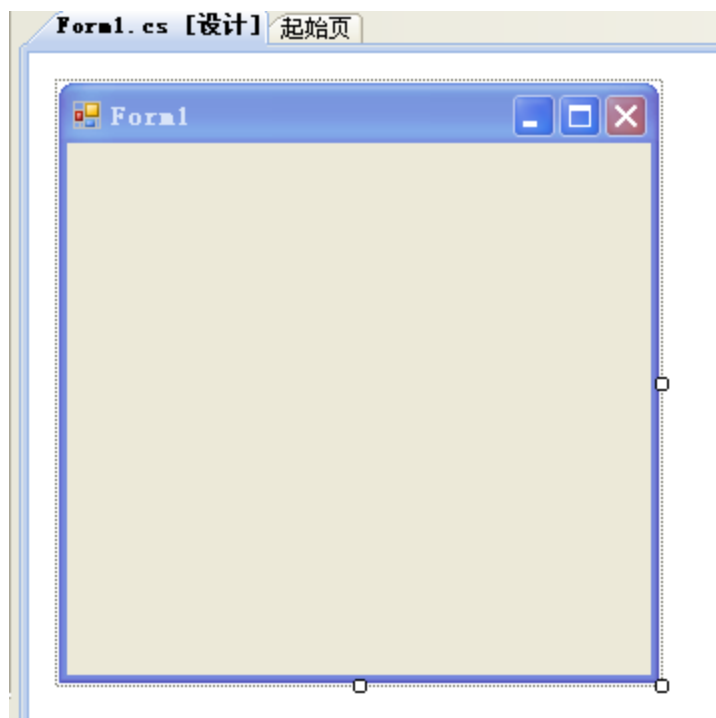


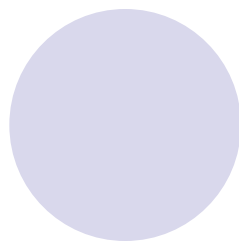
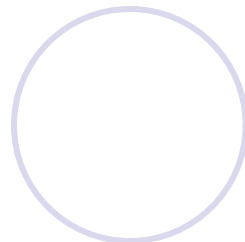
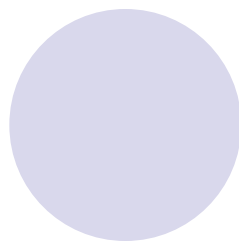
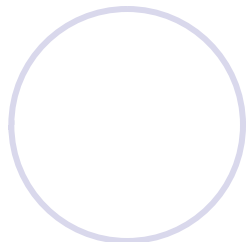
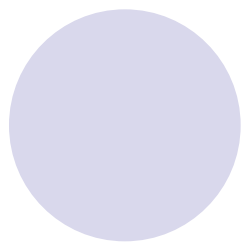
1.3.8 属性窗口



1.3.9 窗体设计器、代码编辑窗口

1. 窗体设计器





2. 代码编辑窗口

```
Program.cs  Form1.cs [设计]  起始页
exwin_01.Program  Main()
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace exwin_01
{
    static class Program
    {
        /// <summary>
        /// 应用程序的主入口点。
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```