



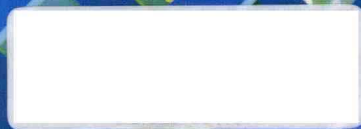
高职高专**立体化教材**计算机系列

C# 程序设计

——Windows 项目开发

(第2版)

邵顺增 李琳 主编



赠送电子课件及
其他立体化资源



清华大学出版社

高职高专立体化教材 计算机系列

C#程序设计——Windows 项目开发 (第2版)

邵顺增 李 琳 主 编

清华大学出版社
北 京

内 容 简 介

C#作为 Microsoft 公司的重要新产品,越来越受到软件开发人员的青睐,现在许多软件公司都把 C#作为自己产品的开发平台。

本书共分 8 章,每章都是以项目引领,每一项目的展开都以项目实施为逻辑顺序,划分为若干工作任务,把相关的理论知识以项目进行的顺序有机地融入阐述。通过本书的学习,读者可以初步具备开发 Windows 应用程序的能力。

本书是在全国进行教育课程教学改革大环境下,为适应高等职业院校项目化教学改革编写的,宗旨是培养学习者的实际编程能力。

本书可以作为高等职业院校信息管理及相关专业的教材,也可以作为其他学习 C#程序设计又想快速提高自己编程能力的学习者的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C#程序设计——Windows 项目开发/邵顺增,李琳主编. —2 版. —北京:清华大学出版社,2012
(高职高专立体化教材 计算机系列)

ISBN 978-7-302-29863-2

I. ①C… II. ①邵… ②李… III. ①C 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 197706 号

责任编辑:桑任松

封面设计:刘孝琼

责任校对:周剑云

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.5 字 数:421 千字

版 次:2008 年 8 月第 1 版 2012 年 10 月第 2 版 印 次:2012 年 10 月第 1 次印刷

印 数:1~4000

定 价:32.00 元

产品编号:045333-01

前 言

传统的教学组织方式主要是班级授课制,以课堂教学的形式来开展,学生学习的主要方式以听讲、讨论、阅读、背诵、理解和书面练习为主。班级授课方式符合传统职业教育课程重理论知识传授的需求,是一种在“去情景化”课堂中,传授给学生所需的理论知识的教学方法。它认为学生通过一定的实训就可以将知识运用于实践,是一种“从理论到实践”的线性演绎过程。这种学校本位的课堂教学模式,造成了职业教育实训课与理论课的划分,同时也将理论与实践割裂开来。

职业教育作为不同于普通教育的另一种类型的教育,是一种研究通过什么样的教育途径来获取合适的职业从业资格的科学。目前,我国职业教育项目课程改革的高潮已经形成,项目课程是高职课程未来的发展方向。高职项目课程的提出,正是以工学结合为主线整合了理论与实践。相对应的项目课程应该注重理论、实践的一体化,以学生活动为主建立适合学生生活的教学组织方式。遵循项目课程改革的思路,笔者及同事们在过去一年的时间里对高职“C#程序设计”这一门课程的项目教学进行了深入的探讨和研究,并进行了一段时间的实践,目前尚不能说我们的项目化教学取得了成功,但至少可以说我们有了一些认识,取得了一些经验。我们的目的在于,探索项目化课程实施的一般性步骤与方法,使得学习主体在学习中有意识地掌握三个相互依存、有机联系的本领:一是学会独立地制订计划,这是一种预测性、诊断性的工作训练;二是学会独立地实施计划,这是一种过程性、形成性的工作训练;三是学会独立地评估计划,这是一种总结性、反馈性的工作训练。

本书就是以项目课程的思路编写的。《C#程序设计》一书分为上、下两册,上册为 Windows 项目开发,下册为 Web 项目开发。本书是上册,上册共选取了 7 个项目,除第 1 章外,每章一个项目,每个项目的展开都以项目实施为逻辑顺序,划分为若干个工作任务,把相关的理论知识以项目进行的顺序有机地融入阐述,而 7 个项目基本覆盖了大纲要求的能力目标,通过本书的学习,读者可以初步具备开发 Windows 应用程序的能力。

本书的第 1 章通过创建一个控制台应用程序和 Windows 应用程序,帮助读者了解 Visual Studio .NET 2010 的开发环境;第 2 章是一个存款计算器小项目,主要帮助读者掌握基本的 C#语法、基本数据类型、变量、表达式计算等基本的编程知识;第 3 章是一个门票销售计算器,主要的教学目标是分支控制结构的运用;第 4 章和第 5 章的项目是一个学生成绩统计器和排序系统,也是比较小的项目,意在帮助读者掌握循环控制结构以及基本的排序算法;第 6 章的项目是一个标准的四则运算计算器,加入此项目主要是学习 C#中的方法,并训练编程的逻辑思维;第 7 章的贪吃蛇游戏则是一个比较复杂的项目,选取这一项目是考虑到面向对象编程语言中关键的一个内容——类的学习,同时涵盖了 GDI+ 的一些基本知识;第 8 章介绍的 C/S 架构的通讯录管理系统,是一个麻雀虽小但五脏俱全的数据库系

统开发项目,即通过 ADO.NET 技术实现对数据的查询、添加、修改和删除等操作。

作为高职“C#程序设计”课程的教材,本书作者安排了一个大概的学时分配,供广大教师或学员参考。

章	学时分配
第 1 章	4
第 2 章	6
第 3 章	8
第 4 章	10
第 5 章	12
第 6 章	18
第 7 章	30
第 8 章	42
合计	130

参加本书编写工作的有邵顺增、李琳、耿亚、冯益斌和车金庆等。在此我们还要对马永山、肖玉表示感谢,他们对本书的编写工作提出了宝贵的意见。

由于作者水平有限,书中难免存在缺点及不妥之处,欢迎广大读者批评指正,同时更欢迎从事本课程教学的同仁与我们共同探讨项目课程以及项目课程教材编写方面的问题。

编 者

目 录

第 1 章 第一个应用程序.....1	3.2 工作任务.....43
1.1 项目一：第一个控制台应用程序.....1	3.3 知识准备.....43
1.1.1 工作目标.....1	3.4 工作任务一：项目分析与算法 流程设计.....52
1.1.2 工作任务.....1	3.5 工作任务二：界面设计.....54
1.1.3 工作任务一：创建第一个空 控制台应用程序.....1	3.6 工作任务三：代码编写.....56
1.1.4 工作任务二：编写程序代码 运行调试程序.....4	3.7 工作任务四：系统运行与效果 测试.....60
1.2 项目二：第一个 Windows 应用程序6	3.8 问题探究.....60
1.2.1 工作目标.....6	3.9 项目实践.....61
1.2.2 工作任务.....7	第 4 章 学生成绩统计器66
1.2.3 工作任务一：创建第一个 Windows 应用程序7	4.1 工作目标.....66
1.2.4 工作任务二：设计程序界面 及控件属性设置.....8	4.2 工作任务.....66
1.2.5 工作任务三：编写程序 代码、运行调试程序.....10	4.3 工作任务一：项目分析与算法流程 设计.....67
1.2.6 问题探究.....12	4.4 工作任务二：代码编写.....68
1.2.7 项目实践.....13	4.4.1 知识准备68
第 2 章 存款计算器14	4.4.2 代码编写与分析80
2.1 工作目标.....14	4.5 工作任务三：系统运行与效果测试.....83
2.2 工作任务.....15	4.6 问题探究.....83
2.3 工作任务一：项目分析与算法 流程设计.....15	4.7 项目实践.....84
2.4 工作任务二：界面设计.....15	第 5 章 成绩排序系统89
2.5 工作任务三：代码编写.....17	5.1 工作目标.....89
2.5.1 知识准备.....17	5.2 工作任务.....90
2.5.2 代码编写与分析.....29	5.3 工作任务一：项目分析与算法 流程设计.....90
2.6 工作任务四：系统运行与效果 测试.....36	5.4 工作任务二：界面设计.....94
2.7 问题探究.....36	5.5 工作任务三：代码编写.....94
2.8 项目实践.....37	5.5.1 知识准备94
第 3 章 门票销售计算器.....40	5.5.2 代码编写与分析106
3.1 工作目标.....40	5.6 工作任务四：系统运行与效果 测试.....111
	5.7 问题探究.....111
	5.8 项目实践.....111

第6章 四则运算计算器.....114

6.1 项目一: 整数四则运算计算器

设计.....114

6.1.1 工作目标.....114

6.1.2 工作任务.....114

6.1.3 工作任务一: 构建计算器
的状态图.....1156.1.4 工作任务二: 设计程序模块
的流程图.....1176.1.5 工作任务三: 设计界面,
完成窗体及控件的属性
设置.....1216.1.6 工作任务四: 编写程序
代码.....1226.1.7 工作任务五: 测试软件,
对错误进行调试修改.....132

6.2 项目二: 实数四则运算计算器

设计.....132

6.2.1 工作目标.....132

6.2.2 工作任务.....132

6.2.3 工作任务一: 设计程序模块
的流程图.....1336.2.4 工作任务二: 编写程序
代码.....136

6.3 项目三: 带记忆功能的计算器

设计.....139

6.4 问题探究.....141

6.5 项目实践.....141

第7章 贪吃蛇游戏.....142

7.1 工作目标.....142

7.2 工作任务.....143

7.3 知识准备——类、对象、字段、
属性和方法.....143

7.3.1 类的基本概念.....143

7.3.2 属性.....146

7.3.3 实例构造函数.....150

7.4 工作任务一: 项目总体设计.....155

7.4.1 项目功能描述.....155

7.4.2 主窗体布局.....155

7.4.3 项目的总体结构.....156

7.5 工作任务二: Bean 类.....157

7.5.1 豆(Been)类分析.....157

7.5.2 豆(Been)类实现.....157

7.5.3 豆(Been)类代码.....160

7.6 工作任务三: Block 类.....161

7.6.1 块(Block)类分析.....161

7.6.2 块(Block)类实现.....162

7.6.3 块(Block)类代码.....163

7.7 工作任务四: Snake 类.....164

7.7.1 蛇(Snake)类分析.....164

7.7.2 蛇(Snake)类实现.....166

7.7.3 蛇(Snake)类代码.....175

7.8 工作任务五: Floor 类.....180

7.8.1 场地(Floor)类分析.....180

7.8.2 场地(Floor)类实现.....181

7.8.3 场地(Floor)类代码.....185

7.9 工作任务六: PubClass 类.....187

7.10 工作任务七: Speed 类.....188

7.10.1 自定义速度(Speed)类分析...188

7.10.2 自定义速度(Speed)类代码...190

7.11 工作任务八: 主界面(Start 类).....190

7.11.1 主界面(Start 类)分析.....190

7.11.2 主界面(Start 类)实现.....193

7.11.3 主界面(Start 类)代码.....198

7.12 知识扩展——派生类.....201

7.13 问题探究.....206

7.14 项目实践.....207

第8章 通讯录管理系统.....208

8.1 工作目标.....208

8.2 工作任务.....209

8.3 知识准备.....209

8.3.1 管理信息系统.....209

8.3.2 C/S 结构.....210

8.3.3 ADO.NET 技术框架.....210

8.4 工作任务一: 系统概要设计.....213

8.4.1 功能模块分析.....213

8.4.2 数据库设计.....213

8.5 工作任务二：设计“用户管理”窗体.....	215	8.8.2 任务实现	247
8.6 工作任务三：设计“用户登录”窗体.....	220	8.9 工作任务六：设计“修改联系人”窗体.....	255
8.6.1 知识准备.....	220	8.9.1 知识准备	255
8.6.2 任务实现.....	223	8.9.2 任务实现	257
8.7 工作任务四：设计“主操作”窗体.....	230	8.10 工作任务七：设计“查找联系人”窗体.....	261
8.7.1 知识准备.....	230	8.10.1 知识准备	261
8.7.2 任务实现.....	237	8.10.2 任务实现	264
8.8 工作任务五：设计“新增联系人”窗体.....	246	8.11 问题探究.....	267
8.8.1 知识准备.....	246	8.12 项目实践.....	268
		参考文献.....	269

第 1 章 第一个应用程序

本章建立一个控制台应用程序及一个 Windows 应用程序，通过两个简单程序的编写及运行，大致了解 Microsoft Visual Studio 2010 的编程环境，同时认识 C#程序的框架和一些基础知识。

1.1 项目一：第一个控制台应用程序

1.1.1 工作目标

1. 终极目标

完成第一个控制台应用程序。

2. 促成目标

- (1) 完成控制台应用程序的创建。
- (2) 完成程序代码的编写和运行调试。

1.1.2 工作任务

工作任务一：创建第一个空控制台应用程序。

工作任务二：编写程序代码、运行调试程序。

1.1.3 工作任务一：创建第一个空控制台应用程序

(1) 选择“开始”|“程序”|Microsoft Visual Studio 2010|Microsoft Visual Studio 2010 命令，出现程序主界面，如图 1.1 所示。

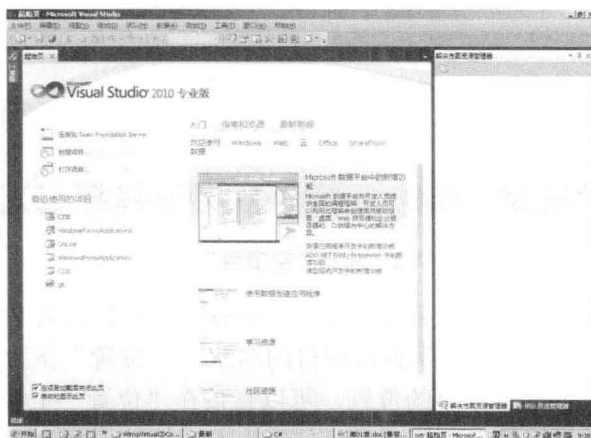


图 1.1 Visual Studio 2010 “起始页”主界面

(2) 在界面的左上角单击“新建项目”链接，打开“新建项目”对话框，如图 1.2 所示。

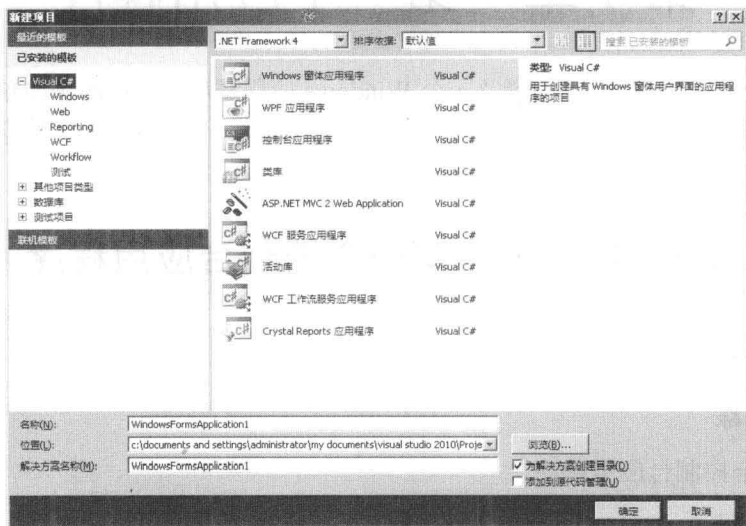


图 1.2 “新建项目”对话框

(3) 在“新建项目”对话框左侧“已安装的模板”中选择 Visual C# 下的 Windows，然后在中间区域选择“空项目”，如图 1.3 所示。

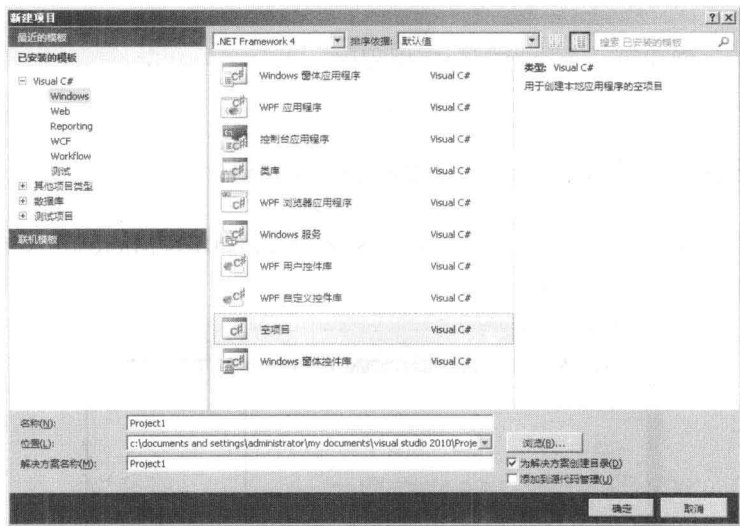


图 1.3 选择“空项目”界面

(4) “新建项目”对话框的“名称”文本框中有系统默认的名字“Projectx”，这里在“名称”文本框中输入“Helloc”作为该项目的名字。“位置”下拉列表框中也有系统默认的位置，若想改成其他存放项目的位置，可以直接在“位置”下拉列表框中输入位置，也可以通过单击“浏览”按钮，打开“项目位置”对话框选择位置。

提示：“名称”和“位置”这两个框中虽都有默认的名字，但最好根据实际情况进行修改，特别是“位置”下拉列表框，系统默认的是 VS .NET 系统文件夹，文件夹层数多，不好记，创建的文件不好找，所以最好改成自己定义的文件夹。

(5) 然后单击“确定”按钮，系统建立一个空项目“Helloc”，并进入 Visual Studio .NET 系统，如图 1.4 所示。

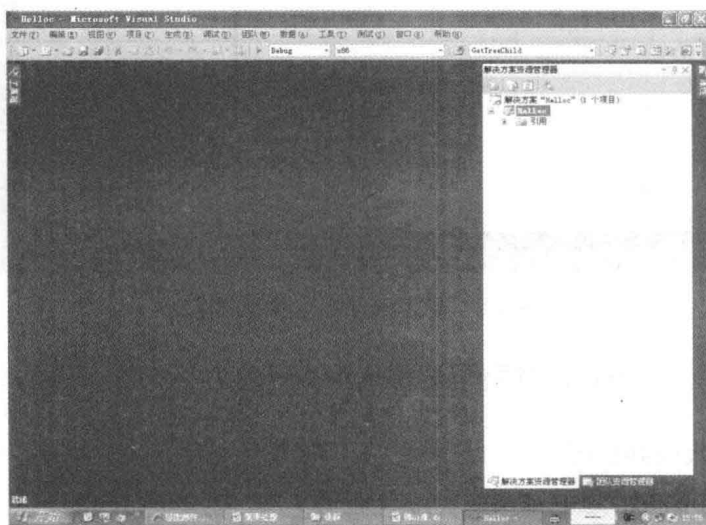


图 1.4 空项目 Helloc 界面

(6) 在右边“解决方案资源管理器”中右击 Helloc 项目，在弹出的快捷菜单中选择“添加”|“添加新项”命令，打开添加新项对话框，如图 1.5 所示。



图 1.5 添加新项对话框

(7) 在“模板”中选择“代码”文件，在“名称”文本框中输入 Helloc(扩展名可省略)，单击“添加”按钮，系统创建一个空的 Helloc.cs 文件，并在左边打开，如图 1.6 所示。

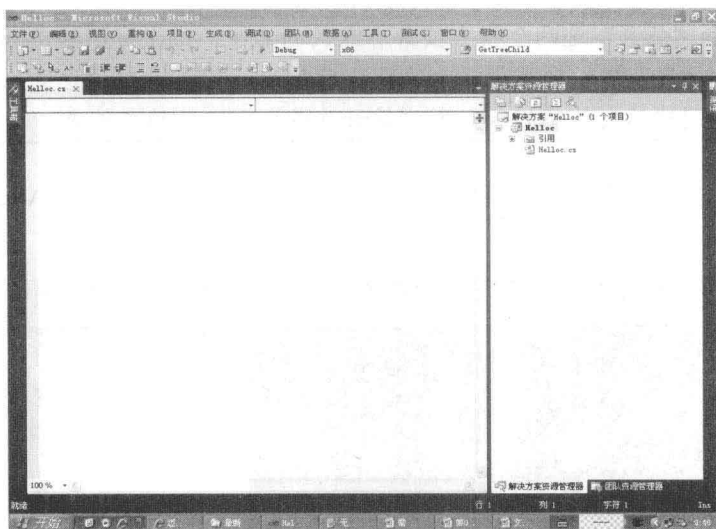


图 1.6 创建 Hello.cs 文件界面

1.1.4 工作任务二：编写程序代码运行调试程序

1. 编写程序代码并执行

(1) 在打开的 Hello.cs 文件中输入以下代码。

```
//引用命名空间
using System;
//自定义类
class Hello
{
//主方法
    static void Main()
    {
        //在控制台上输出“我的第一个应用程序!”
        Console.WriteLine ("我的第一个应用程序!");
    }
}
```

(2) 在菜单栏中选择“调试”|“开始执行(不调试)”命令，也可以按 Ctrl+F5 组合键。运行结果如图 1.7 所示。



图 1.7 运行结果

2. 程序解读

1) C#程序的基本框架

所有 C# 程序都有一个基本框架，它的基本结构如下。

```
using 命名空间
class 类名
{
    ...
    static void Main()
    {
        方法体
    }
}
```

2) 命名空间

`using System` 表示导入命名空间 `System`。高级语言为了提高编程的效率，总是在系统中加入许多系统预定义的元素，即编写了许多完成常用功能的程序放在系统中，编程时只要把系统中的程序导入即可使用。

在 C# 中，把系统中包含的程序按功能分成多个部分，每部分放在一个命名空间中，需要时用如下形式导入命名空间。

```
using 命名空间
```

而另一方面，必须得到与之相匹配的动态链接库的支持，即必须首先添加“命名空间”的引用，否则编译环境就会无法识别。

3) 定义类

C# 的每一个程序至少包括一个自定义类，程序的所有内容都必须属于一个类。类的定义格式如下。

```
Class Hello
{
    类体
}
```

类中的所有代码都包含在一对大括号({})中，左右大括号必须全部配套，如果出现大括号不匹配的现象，那么程序就是错误的。

4) Main 方法

C# 中必须且只能包含一个 `Main` 方法，其格式必须如下。

```
static void Main()
{
    方法体
}
```

此方法是程序的入口点和出口点，系统从 `Main` 方法开始执行，`Main` 方法结束，程序就结束。

5) 代码注释

注释是非执行语句，即在程序运行时不执行此类语句，它的作用是对语句或程序段做说明，帮助理解程序，提高程序的可读性。对现代编程来说，程序的可读性是程序性能的

重要标志之一,所以一定要从一开始就养成写注释的编程习惯。

注释语句的格式如下。

```
//: //后的为注释内容
/*...*/: 符号内的是注释内容
```

例如: //自定义类。

6) 输出

控制台应用程序通常使用 .NET 框架的运行时库提供的输入/输出服务。前面第一个控制台应用程序所完成的输出功能都是通过 System 命名空间中的控制台(Console)类中的 WriteLine 方法完成的。

格式如下。

```
System.Console.WriteLine(输出项);
```

功能: 把输出项在控制台上输出。

例如:

```
System.Console.WriteLine("Hello World!"); //输出字符串 Hello World!
```

System 命名空间中的控制台(Console)类中还有与之对应的输入方法 ReadLine。

格式如下。

```
System.Console.ReadLine();
```

功能: 从键盘上读取一行以 Enter 键为结束符的数据,并把读取的数据返回。

由于它要返回一个值,所以通常要把它返回的值存放起来,以备后面使用。通过如下方式来使用它。

```
string str;
str=System.Console.ReadLine();
```

如果在程序的开头导入了命名空间 System,则在程序中可直接使用 Console 类的方法,无需使用完全限定名。例如:

```
Console.WriteLine("Hello World!");
Str=Console.ReadLine();
```

7) 语句结束符

分号(;)是 C# 程序中的语句结束符,除控制语句外,每条语句都以“;”结束。

8) 编译并运行程序

从 IDE 编译并运行程序。按 F5 键或 Ctrl+F5 组合键运行程序。

1.2 项目二: 第一个 Windows 应用程序

1.2.1 工作目标

1. 终极目标

完成第一个 Windows 应用程序。

2. 促成目标

- (1) 完成 Windows 应用程序的创建。
- (2) 完成界面及控件属性的设置。
- (3) 完成程序代码并运行调试。

1.2.2 工作任务

工作任务一：创建第一个空 Windows 应用程序。

工作任务二：设计程序界面及控件属性。

工作任务三：编写程序代码、运行调试程序。

1.2.3 工作任务一：创建第一个 Windows 应用程序

(1) 选择“开始”|“程序”|Microsoft Visual Studio 2010|Microsoft Visual Studio 2010 命令，出现程序主界面，如图 1.8 所示。

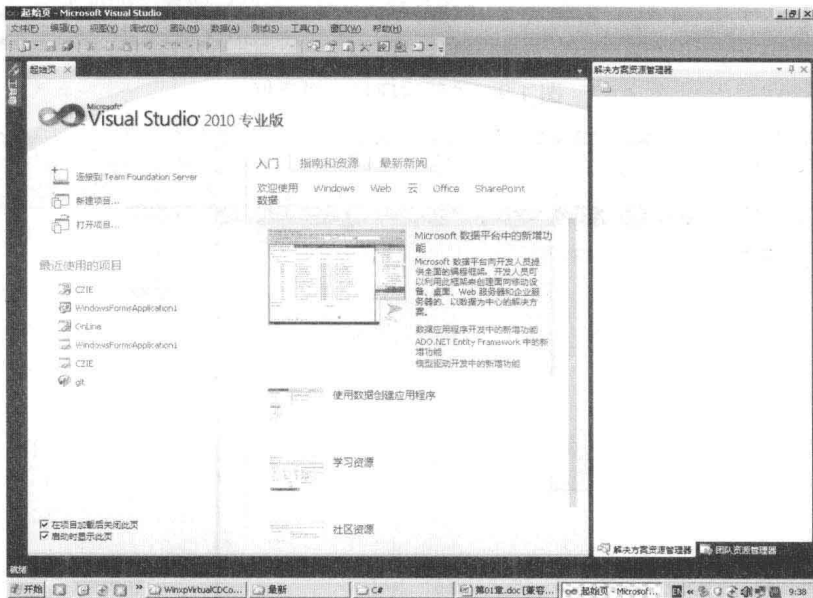


图 1.8 Visual Studio .NET 2010 起始页

(2) 在界面的左上角单击“新建项目”链接，打开“新建项目”对话框，如图 1.9 所示。

(3) 在“新建项目”对话框左侧下“已安装的模板”中选择 Visual C# 下的 Windows，在中间区域选择“Windows 窗体应用程序”项，在“名称”文本框中会出现系统默认的名字“WindowsApplicationx”，这是以“Hellow”作为该项目的名字。“位置”下拉列表框中也有系统默认的位置，也改成想要存放项目的位置，可以直接在“位置”下拉列表框中输入位置，也可以通过单击“浏览”按钮，打开“项目位置”对话框选择位置。

(4) 然后单击“确定”按钮，系统建立一个 Windows 项目“Hellow”，并进入 Visual

Studio .NET 系统，如图 1.10 所示。

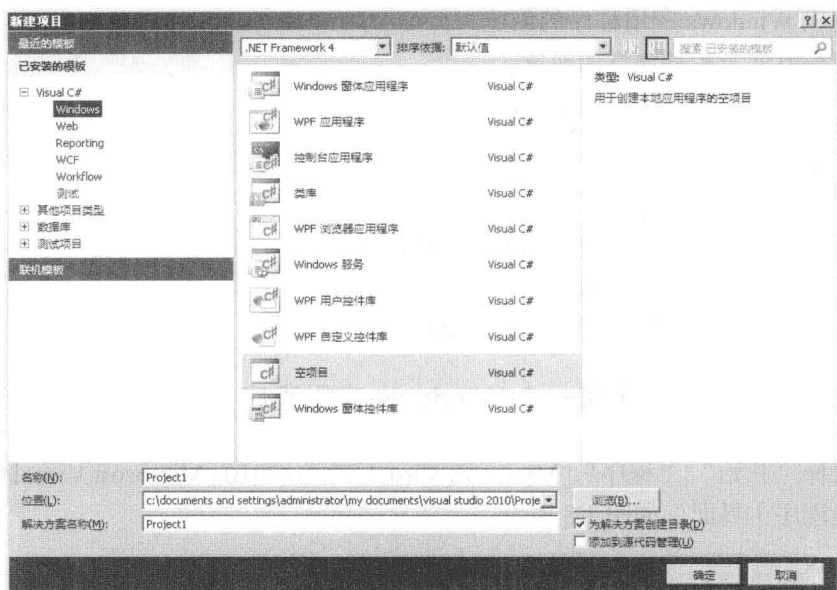


图 1.9 “新建项目”对话框

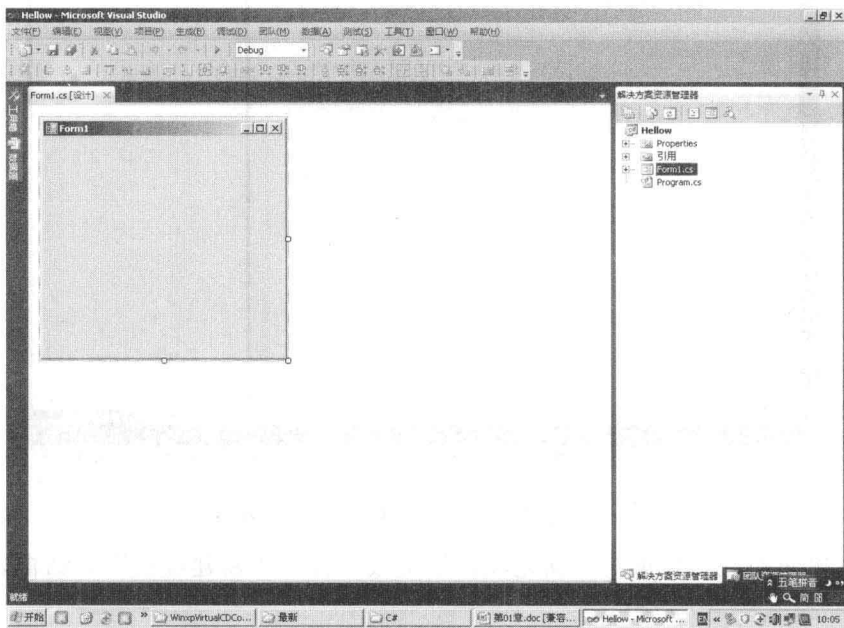


图 1.10 新建 Windows 项目 Hello 界面

1.2.4 工作任务二：设计程序界面及控件属性设置

(1) 将鼠标指针移到左边(有的在右边)“工具箱”标签上，出现“工具箱”窗口，如图 1.11 所示。

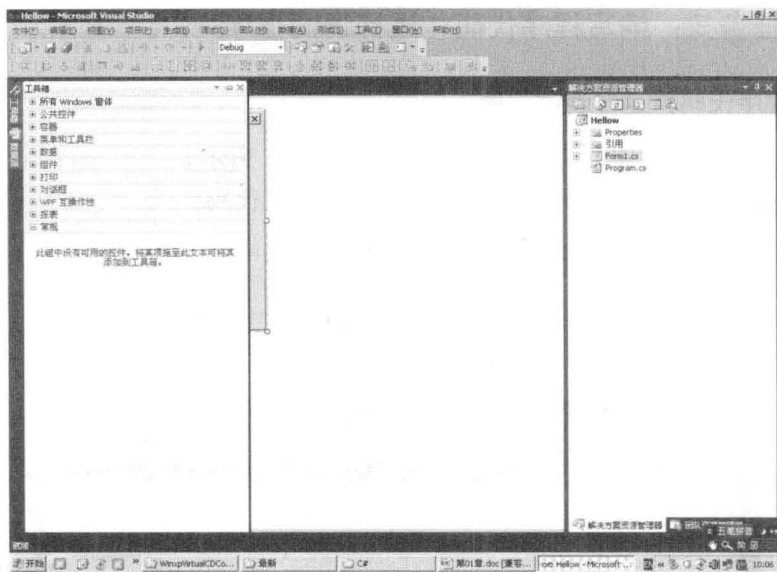


图 1.11 “工具箱”窗口

(2) 单击“工具箱”窗口上的“所有 Windows 窗体”，出现“所有 Windows 窗体”下拉菜单，单击 Button 控件并将其拖到窗体上，窗体上出现一个 Button 控件，上面显示 button1，再拖动一个 TextBox 控件到窗体上，摆放好它的位置，如图 1.12 所示。

(3) 右键单击 button1 按钮，从快捷菜单中选择“属性”命令，在窗口右边出现“属性”窗口，在“属性”窗口中单击“属性”按钮，从中找到 Text 属性，它的值默认为 button1，此处修改为“确定”，使按钮上显示的文字为“确定”；找到 Name 属性，它的值默认也为 button1，此处修改为 btnOK。单击 textBox1(同样它的四面八方有小方块)，单击右下角的“属性”窗口，找到(Name)属性，它的默认值为 textBox1，此处修改为 txtDisplay。

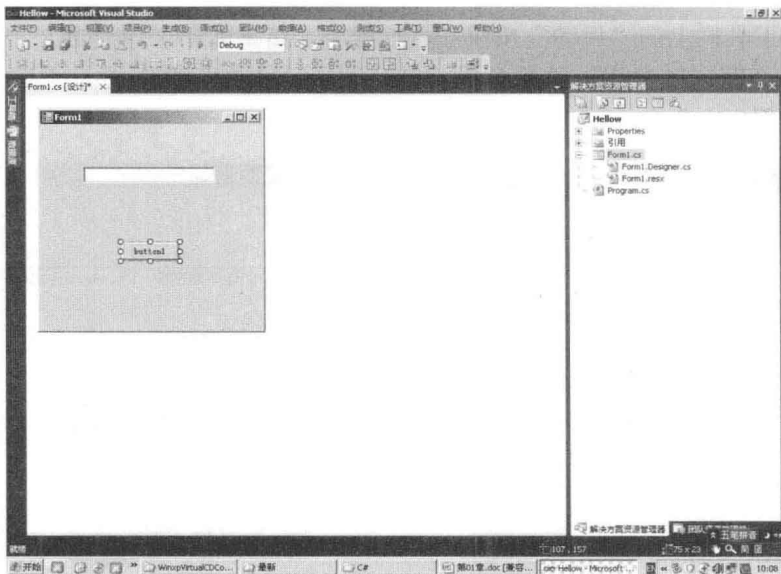


图 1.12 窗体设计界面