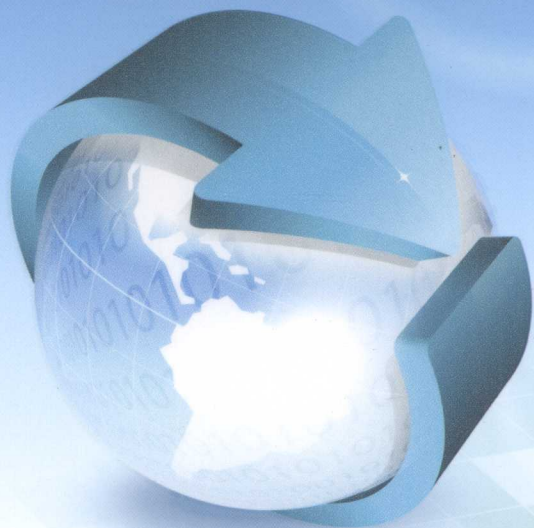




高职高专工作过程·立体化创新规划教材

——计算机系列



C# WEB 数据库编程

刘晓刚 吴俊峰 主 编
王 琼 梅 冰 刘 芳 副主编

赠送
电子课件



清华大学出版社

高职高专工作过程·立体化创新规划教材——计算机系列

C# WEB 数据库编程

刘晓刚 吴俊峰 主 编

王 琼 梅 冰 刘 芳 副主编

清华大学出版社

北 京

内 容 简 介

这是一本严格采用“任务驱动、项目导向”模式规范编写的 C#数据库及 Web 应用设计编程教材。本书内容可分为三个部分: C#程序设计基础、ASP.NET 网站开发和 ADO.NET 数据库开发。三部分内容在“精学够用”的基础上侧重于知识综合, 目的是培养学生动态网站的综合开发能力。本书以 Visual Studio 2005 为开发平台, 主要介绍了 ASP.NET 应用程序开发、C#基本语法知识、面向对象程序设计知识、Web 页面设计基础、Web 控件、ASP.NET 内置对象等, 在 ASP.NET 中使用 XML, 以 SQL Server 2005 为基础进行 ADO.NET 数据库编程, 使用用户控件、自定义控件, 创建 Web 服务等。本书内容丰富, 结构精炼, 操作方便, 实例典型、实用。

本书作为动态网站开发的入门类教材, 既可以作为高等职业教育计算机及相关专业的教材, 也可作为 C#、ASP.NET、ADO.NET、Web 数据库的各种培训班、职业资格等级考试或认证考试的培训教材, 还可用于读者自学。

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# WEB 数据库编程/刘晓刚, 吴俊峰主编; 王琼, 梅冰, 刘芳副主编. —北京: 清华大学出版社, 2010.8
(高职高专工作过程·立体化创新规划教材——计算机系列)
ISBN 978-7-302-23018-2

I. ①C… II. ①刘… ②吴… ③王… ④梅… ⑤刘… III. ①C 语言—程序设计—高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 107719 号

责任编辑: 章忆文 张丽娜

装帧设计: 山鹰工作室

责任校对: 王 晖

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 刷 者: 北京密云胶印厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 24.25 字 数: 584 千字

版 次: 2010 年 8 月第 1 版 印 次: 2010 年 8 月第 1 次印刷

印 数: 1~4000

定 价: 38.00 元

产品编号: 038089-01

前 言

随着高等职业技术教育的发展,“任务驱动、项目导向”成了主流的教学模式。本书以 Microsoft Visual Studio 2005 为集成开发环境,通过多个典型、实用的项目,引导学生完成基于 C#语言的 ASP.NET 并连接后台数据库的交互式网站开发,除了学习 C#、ASP.NET 和 ADO.NET 三方面的技术外,更侧重于培养学生对三种技术的综合运用能力。

本书特色

综合 C#、Web 和数据库三方面的技术,重点培养动态网站的综合开发技能,有效避免学生所学知识分离,不懂综合应用的弊病。运用“任务驱动、项目导向”模式开发,每一章都有引导项目进行任务驱动学习,章后均配有项目实践进行加强,同时配以上机测试考核,效果更好;理论知识“精学够用”,在工作任务开发的过程中穿插讲解相关知识;专业软件公司全程指导并参与编写,同时参考了微软教材和其他多本教材再加以创新,所选项目和任务丰富实用,课例简短而且说明性强。

本书由几所院校的优秀教师和专业软件公司联合编写,是在各家院校的成熟教案及原有自编教材基础上整合编写而成的,作者拥有丰富的开发案例和教学经验。本书共分 15 章,计划需要 190 个课时,建议分为 C#(第 1~8 章)和 ASP.NET(第 9~15 章)两部分,各用一学期进行学习。为了给教师授课提供方便,我们还提供了与本书对应的教学课件和工作实训营项目源程序,读者可以与作者(电子邮箱: sylxg@126.com)联系索取。

本书主要内容

第 1 章介绍如何建立一个输出“Hello,world”的项目,通过控制台应用程序和 Windows 应用程序两种方式实现,目的在于让学生熟悉 Microsoft Visual Studio 2005 编程环境,同时了解 Microsoft .NET Framework 和 C#的相关知识,为第一部分 C#编程及面向对象知识打好基础。

第 2 章介绍如何实现一个简单的加减法测试器项目。通过该项目的实现,主要学习 C#语法的一些基础知识,比如标识符和关键字、书写规则以及数据类型等,为编写复杂应用程序奠定基础。

第 3 章讲解如何实现一个电力公司的电费计算项目。通过该项目的实现,主要学习 C#语言的运算符与表达式,以及标签控件、按钮控件和文本框控件的用法以及一些简单的编程知识。

第 4 章介绍如何设计一个公司员工的薪水计算项目,用于实现对普通公司员工每月薪水的计算。通过该项目的实现,主要学习如何编写分支结构的应用程序。

第 5 章讨论如何设计一个对某组学生成绩进行分析的项目,可以分别找出其中的最高分和最低分并求出平均分。通过该项目的实现,主要学习如何编写循环结构和 C#数组的应用程序。

第 6 章先通过建立一个能吃能唱的燕子类项目来学习类的基本创建方法;然后进行知识的加深拓展,通过一个能升级的士兵项目来学习类的重载知识。

第 7 章介绍如何设计一个学生类项目和面积计算的项目,第一个项目学习类的继承,

第二个项目学习接口和多态。

第 8 章讲解如何设计一个能交互窗体的项目,主要学习 C#委托和事件以及 C#异常处理的相关知识。掌握委托的使用方法,事件的声明和定义方法以及 C#异常处理技术。

第 9 章介绍如何设计一个用户注册网页,学习各种 HTML 控件的作用与用法,掌握各种 HTML 控件的作用与用法,以及 ASP.NET 开发与调试、发布的方法。

第 10 章通过设计密码修改网页,学习常用 Web 控件的使用方法和属性,正则表达式及其语法格式,服务器端验证控件。掌握常用 Web 控件、标准选择控件、正则表达式的语法格式和用法以及服务器端验证控件的使用方法。

第 11 章通过设计开发电子商务网站一个简单的购物车,学习 ASP.NET 的 Response、Request、Server、Application、Session 以及 Cookie 等对象。目标是掌握这些对象的用法和实现网页计数器、购物车以及用户访问网页计数的方法。

第 12 章通过设计开发在线即时贴,主要学习 ADO.NET 技术,包括面向连接的数据访问技术,面向无连接的数据访问技术,使用数据源控件连接数据库,数据绑定,使用数据控件显示查询到的数据,使用数据控件实现数据的更新、插入和删除。

第 13 章通过开发计数器用户控件,主要学习用户控件的属性和事件以及自定义控件的知识。掌握用户控件的创建,如何使用用户控件的属性和事件,以及自定义控件的创建和使用方法。

第 14 章通过一个引导项目主要学习网站的统一布局与导航的知识,主要内容有母版页、站点地图控件、主题与外观等。目的是掌握母版页的创建和可编辑区域的定义,如何进行网页与母版页的关联,掌握网站结构的创建和定义,以及导航控件的使用。

第 15 章重点介绍了一个新闻发布系统实例,主要内容有注册页面的编写、后台数据库的设计、数据查询、数据添加、数据删除、数据修改和调试运行。目的是通过一个后台功能较为完备的“新闻发布系统”网站的制作,使学生了解 ASP.NET 项目开发的完整过程,提高学生対前面章节知识的综合运用能力,进一步加深对所学知识的理解。

读者对象

本书作为动态网站开发的入门类教材,既可以作为高等职业教育计算机及相关专业的教材,也可作为 C#、ASP.NET 和 ADO.NET 的各种培训班、职业资格等级考试或认证考试的培训教材,还适用于读者自学。

参与本书编写的主要是浙江经贸职业技术学院的一线教师和杭州思伟信息科技有限公司的工程师。本书由刘晓刚、吴俊峰担任主编,王琼(十堰教育学院教师)、梅冰(杭州思伟信息科技有限公司总经理)、刘芳担任副主编。此外,参与本书编写的还有张斌(桂林电子科技大学教师)、赵强(嘉兴职业技术学院教师)和彭小泉(杭州思伟信息科技有限公司技术总监)。

本书的编写得到了浙江经贸职业技术学院、十堰教育学院、桂林电子科技大学、嘉兴职业技术学院等高校和杭州思伟信息科技有限公司的支持,在此表示衷心的感谢。

由于作者水平有限,编写过程中难免存在不足之处,欢迎读者与我们联系,帮助我们提高。

编 者

目 录

第 1 章 概述——导入“Hello, world”程序..... 1

- 1.1 工作场景导入..... 2
- 1.2 .NET 框架与 C#语言 3
- 1.3 控制台应用程序的结构..... 4
 - 1.3.1 控制台应用程序的基本结构 4
 - 1.3.2 输出..... 5
 - 1.3.3 编译并运行程序..... 6
- 1.4 Windows 应用程序结构..... 6
- 1.5 回到工作场景 1..... 7
- 1.6 回到工作场景 2..... 9
- 1.7 工作实训营..... 10
- 1.8 习题..... 11

第 2 章 加减法测试器 13

- 2.1 工作场景导入..... 14
- 2.2 C#标识符和关键字 14
- 2.3 代码书写规则..... 15
- 2.4 数据类型..... 15
 - 2.4.1 C#语言的数据类型划分 15
 - 2.4.2 值类型..... 17
 - 2.4.3 引用类型..... 20
 - 2.4.4 DateTime 和 TimeSpan 类..... 23
 - 2.4.5 Math 类 24
 - 2.4.6 数据类型的转换..... 25
- 2.5 回到工作场景..... 28
- 2.6 工作实训营..... 32
- 2.7 习题..... 32

第 3 章 电费计算程序 35

- 3.1 工作场景导入..... 36
- 3.2 C#运算符和表达式 36
 - 3.2.1 运算符的分类..... 36
 - 3.2.2 运算符的优先级..... 37
 - 3.2.3 算术运算符和算术表达式 38

- 3.2.4 赋值运算符和赋值表达式..... 38
- 3.2.5 关系运算符和关系表达式..... 39
- 3.2.6 逻辑运算符和逻辑表达式..... 40
- 3.2.7 位运算 41
- 3.2.8 其他特殊运算符 43
- 3.3 回到工作场景 46
- 3.4 工作实训营 48
- 3.5 习题 49

第 4 章 薪水计算程序 51

- 4.1 工作场景导入 52
- 4.2 选择结构程序设计 52
 - 4.2.1 if 语句..... 53
 - 4.2.2 if...else 语句..... 54
 - 4.2.3 if...else 语句嵌套..... 56
 - 4.2.4 switch 语句..... 58
- 4.3 回到工作场景 61
- 4.4 工作实训营 65
- 4.5 习题 65

第 5 章 成绩分析程序 67

- 5.1 工作场景导入 68
- 5.2 C#数组..... 68
 - 5.2.1 C#数组的声明..... 68
 - 5.2.2 数组的初始化 69
 - 5.2.3 访问数组成员 70
 - 5.2.4 数组对象 70
- 5.3 循环结构程序设计 70
 - 5.3.1 while 语句 71
 - 5.3.2 do...while 语句..... 72
 - 5.3.3 for 语句..... 73
 - 5.3.4 foreach 语句 75
 - 5.3.5 跳转语句 76
- 5.4 回到工作场景 77
- 5.5 工作实训营 80

5.6 习题.....	81	7.8 习题.....	133
第 6 章 会歌唱的燕子和能升级的士兵.....	83	第 8 章 能交互的窗体.....	139
6.1 工作场景导入.....	84	8.1 工作场景导入.....	140
6.2 类的创建.....	84	8.2 委托、事件与异常.....	140
6.2.1 创建类和对象.....	85	8.2.1 委托.....	140
6.2.2 访问修饰符.....	86	8.2.2 事件.....	142
6.3 编写方法.....	87	8.2.3 异常.....	146
6.3.1 没有参数的方法.....	88	8.3 回到工作场景.....	150
6.3.2 带有参数的方法.....	88	8.4 工作实训营.....	152
6.3.3 静态方法.....	89	8.5 习题.....	153
6.4 构造函数与析构函数.....	90	第 9 章 用户注册网页.....	155
6.4.1 实例构造函数.....	90	9.1 工作场景导入.....	156
6.4.2 静态构造函数.....	92	9.2 HTML 控件与 HTML 标记.....	156
6.4.3 析构函数.....	93	9.2.1 HTML 控件与 HTML 标记的 区别.....	157
6.5 属性.....	94	9.2.2 HTML 控件的常用属性.....	157
6.6 重载.....	96	9.3 HtmlInput 控件.....	159
6.6.1 运算符重载.....	96	9.3.1 Input(Button).....	159
6.6.2 方法重载.....	97	9.3.2 Input(CheckBox).....	160
6.6.3 构造函数重载.....	98	9.3.3 Input(Radio).....	161
6.7 回到工作场景 1.....	99	9.3.4 Input(Text)和 Input(Password).....	162
6.8 回到工作场景 2.....	102	9.3.5 Input(File).....	164
6.9 工作实训营.....	104	9.3.6 Input(Hidden).....	165
6.10 习题.....	105	9.4 HtmlSelect 控件.....	167
第 7 章 学生类和面积计算项目.....	109	9.5 HtmlTextArea 控件.....	168
7.1 工作场景导入.....	110	9.6 HtmlImage 控件.....	170
7.2 类的继承.....	110	9.7 HtmlTable 控件.....	171
7.2.1 实现继承.....	111	9.8 HtmlDiv 控件.....	173
7.2.2 构造函数.....	112	9.9 回到工作场景.....	175
7.2.3 重写成员.....	116	9.10 工作实训营.....	178
7.3 接口.....	119	9.11 习题.....	178
7.3.1 创建接口.....	119	第 10 章 密码修改网页.....	181
7.3.2 接口成员.....	121	10.1 工作场景导入.....	182
7.3.3 接口与继承.....	123	10.2 Web 服务器控件.....	182
7.4 多态.....	125	10.2.1 Web 服务器控件概述.....	182
7.5 回到工作场景 1.....	128	10.2.2 Web 标准控件.....	183
7.6 回到工作场景 2.....	130		
7.7 工作实训营.....	133		

10.3 标准选择控件	196	12.7.3 数据绑定控件	283
10.4 Web 服务器端验证控件	201	12.7.4 数据绑定的语法和步骤	284
10.5 回到工作场景	212	12.8 GridView 控件	285
10.6 工作实训营	215	12.9 DataList 控件	295
10.7 习题	215	12.10 DetailsView 控件	301
第 11 章 购物车	219	12.11 FormView 控件	303
11.1 工作场景导入	220	12.12 Repeater 控件	306
11.2 Response 对象	220	12.13 回到工作场景	309
11.3 Request 对象	223	12.14 工作实训营	318
11.4 Server 对象	230	12.15 习题	320
11.5 Application 对象及网页计数器	232	第 13 章 计数器用户控件	323
11.6 Session 对象及购物车	236	13.1 工作场景导入	324
11.7 Cookie 对象及用户访问计数	241	13.2 用户控件	324
11.8 回到工作场景	246	13.2.1 用户控件概述	324
11.9 工作实训营	249	13.2.2 用户控件的创建和使用	325
11.10 习题	250	13.3 自定义控件	333
第 12 章 在线即时贴	251	13.3.1 自定义控件概述	333
12.1 工作场景导入	252	13.3.2 创建一个自定义控件	334
12.2 创建数据库	253	13.4 回到工作场景	337
12.2.1 数据库的创建	254	13.5 工作实训营	339
12.2.2 创建数据表	255	13.6 习题	340
12.2.3 ASP.NET 的两种数据访问 模式	256	第 14 章 网站的统一布局与导航	343
12.3 了解 ADO.NET	256	14.1 工作场景导入	344
12.3.1 ADO.NET 的结构	257	14.2 应用程序统一布局的实现	344
12.3.2 ADO.NET 的对象	257	14.3 网站地图与导航	351
12.4 导入命名空间与连接数据库	258	14.4 回到工作场景	360
12.5 使用 SQL 命令执行数据库操作 和查询	262	14.5 工作实训营	363
12.6 使用 DataSet 对象实现数据库 操作	270	14.6 习题	364
12.7 数据源控件与数据绑定	273	第 15 章 新闻发布系统综合实训	367
12.7.1 数据源控件概述	273	15.1 工作场景导入	368
12.7.2 使用 SqlDataSource 控件	274	15.2 回到工作场景暨实训过程指导	371
		15.3 工作实训营	376
		参考文献	377

第 1 章

概述——导入“Hello,world” 程序



本章要点

- .NET 框架。
- C#程序的基本结构。
- 控制台应用程序和 Windows 应用程序的编写方法。



技能目标

- 掌握 C#应用程序的基本结构。
- 掌握 C#控制台应用程序和 Windows 应用程序的编写方法。



1.1 工作场景导入

建立一个输出字符串“Hello,world”的程序,要求通过两种方式实现:控制台应用程序和 Windows 应用程序。

【工作场景 1】简单的控制台应用程序。

如图 1.1 所示,在 Microsoft Visual Studio 2005(简称 VS2005)编程环境中,利用 C#语言编写一个能够输出字符串“Hello,world”的控制台应用程序。

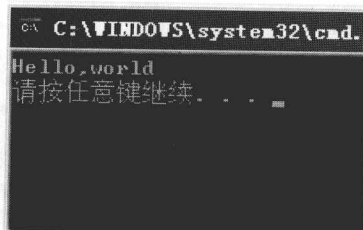


图 1.1 控制台应用程序

【工作场景 2】简单的 Windows 应用程序。

如图 1.2 所示,创建一个窗体,上面有一个按钮和一个文本框。单击按钮,将在文本框中显示“Hello,world”字符串。

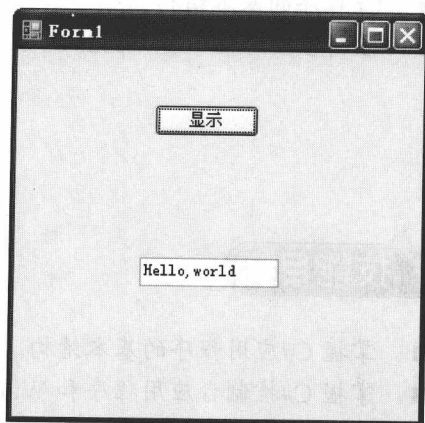


图 1.2 Windows 应用程序

【引导问题】

- (1) 什么是 .NET 框架? Microsoft Visual C# 2005 是什么?
- (2) 如何使用 VS2005 开发 C# 控制台应用程序?
- (3) 如何使用 VS2005 开发 Windows 应用程序?



1.2 .NET 框架与 C#语言

【学习目标】 了解.NET 框架和 C#语言的相关知识。

1. 什么是.NET

Microsoft 公司对.NET 所做的定义为：.NET 是 Microsoft 面向 XML Web 服务的平台，它使用一种统一的、个性化的方式将信息、设备和人员紧密地联系在一起。

要澄清.NET 的概念，需要将.NET 分成 3 个主要部分。

- .NET 战略：该战略基于这样一种想法，即所有的设备将来都可以通过 Internet 连接在一起，同时所有的软件都将成为在该网络上提供的一种服务。
- .NET Framework：是指像 C#、ASP.NET 这样可使.NET 更加具体的技术。该架构提供了具体的服务和技术，以便开发人员创建相应的应用程序来满足连接到 Internet 的用户的需要。
- .NET 企业服务器：是指像 SQL Server 2005 这样的由.NET Framework 应用程序使用的服务器产品。

2. 什么是.NET 框架

.NET 框架(即.NET Framework)是支持生成、运行下一代应用程序和 XML Web 服务的内部 Windows 组件，是 Visual Studio .NET 应用程序开发环境的核心。它定义了语言之间互操作的规则，以及如何把应用程序编辑为可执行代码，它还负责管理任何 Visual Studio .NET 语言创建的应用程序的执行。.NET Framework 内容丰富，这里仅介绍最重要的几个部分。

在该框架上可以编写识别 Internet 并支持 Internet 的应用程序，而且这些应用程序接受诸如 XML、HTTP 和 SOAP 这样的开放标准。

该框架提供了一些非常重要且强大的应用程序开发技术。例如，可以用来创建典型的 GUI 应用程序的 Windows Form。当然，也包括用于创建网络应用程序的 ASP.NET。

.NET Framework 具有两个组件：公共语言运行库(Common Language Runtime, CLR)和.NET Framework 类库(Framework Class Library, FCL)。

3. Visual Studio.NET 及 Visual C#

Visual Studio .NET(后面简称 VS.NET)是.NET 平台下最为强大的开发工具，无论是软件服务商，还是企业应用程序的部署与发布，Visual Studio .NET 都可以提供近乎完美的解决方案。

Visual Studio .NET 是唯一一个从一开始就是为 XML Web 服务面创建的开发环境。

Microsoft Visual C# 2005 是一种编程语言，它是为生成在 .NET Framework 上运行的多种应用程序而设计的。C#(读作 C Sharp)语言简单，功能强大，类型安全，而且是一种面向对象的语言。

Visual Studio 支持 Visual C#，这是通过功能齐全的代码编辑器、项目模板、设计器、代码向导和功能强大且易于使用的调试器以及其他工具来实现的。



1.3 控制台应用程序的结构

【学习目标】 掌握 VS2005 控制台应用程序的基本结构，学会用 C# 语言编写和运行控制台应用程序。

1.3.1 控制台应用程序的基本结构

1. 基本结构

控制台应用程序的基本结构为：

```
//结构开始
using 命名空间;
class 类名
{
    ...
    static void Main()
    {
        方法体
    }
}
//结构结束
```

2. 命名空间

上述控制台应用程序的基本结构中的第一行语句“using 命名空间;”的作用是导入命名空间，该语句类似于 C 和 C++ 中的 #include 命令。导入命名空间之后，就可以自由使用其中的元素了。System 是 .NET 框架提供的最基本的命名空间之一，Console 是 System 命名空间包含的系统类库中已定义的一个类。利用该类的 ReadLine() 和 WriteLine() 方法，可以进行输入/输出操作。

3. 定义类

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

```
Class 类名
{
    类体
}
```

类中的所有代码都被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，那么这就是一个错误的程序。

4. Main()方法

在一个 C# 程序中，必须且只能包含一个 Main() 方法，其格式如下：

```
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() 方法一旦结束，程序随即结束。

5. 语句结束符

分号(;)是 C# 程序中语句的结束符，除控制语句外，每一条语句都以分号结束。

6. 代码注释

代码注释是非执行语句，它的作用是对语句或程序段作说明，帮助程序员理解程序，提高程序的可读性。

注释语句的格式如下：

```
//: //后的内容为注释内容，如：//自定义类。
/*...*/: 符号内的是注释内容。
///: XML 注释方式。///符号是一种特殊的注释方式，可以利用 Visual Studio .NET 开发工具
将///注释转换为 XML 文件。
```

1.3.2 输出

控制台应用程序通常使用 .NET 框架运行时库所提供的输入/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 命名空间的控制台(Console)类中的 WriteLine() 方法来完成的。

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回车为结束符的数据，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。例如：

```
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，而无需使用完全限定名。例如：

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

1.3.3 编译并运行程序

从 IDE 编译并运行程序，可以按 `F5` 或 `Ctrl+F5` 键。



1.4 Windows 应用程序结构

【学习目标】掌握 VS2005 下 Windows 应用程序的基本结构，学会用 C# 语言编写和运行 Windows 应用程序。

Windows 应用程序框架跟控制台应用程序一样，并且创建了一个供编程人员设计界面的窗体，用户可以向窗体增加其他控件，并修改控件的属性。下面介绍编写 Windows 应用程序的过程。

1. 添加控件

控件就是对象，可以用它显示信息，通过它向系统输入信息或者响应用户的操作。它们被放在 `Form` 对象中。控件具有自己的一些属性、方法和事件。向窗体添加控件的方法如下。

- (1) 在工具箱中单击所要添加的控件，然后把鼠标移到窗体上，按住鼠标左键并拖动鼠标，画出所需的控件。
- (2) 从工具箱中拖动控件到窗体上。
- (3) 在工具箱中双击所要添加的控件，即可把控件添加到窗体上。

2. 设置控件的属性

单击窗体中的控件，该控件被选中，控件边上出现带有 8 个小方块的虚线矩形选框，然后通过属性窗口设置该控件的属性。例如：设置按钮控件的 `Text` 属性为“确定”。设置文本框的 `Name` 属性为 `Display`。

3. 编写 Windows 程序代码

进入控件事件(按钮的单击事件)方法编辑窗口的的方法有如下 3 种。

- (1) 双击要编写事件方法的按钮。
- (2) 单击属性窗口中具有闪电图标的事件按钮，出现事件窗口，然后查找到事件名并双击它。
- (3) 在事件窗口中找到所需事件名后，在它的右边输入名字，然后双击它。

4. 编译运行

从 IDE 编译并运行程序，可以按 F5 或 Ctrl+F5 键，方法与控制台应用程序相同。



1.5 回到工作场景 1

通过 1.2~1.4 节内容的学习，我们掌握了 .NET 框架的基本知识和 C# 程序的基本结构，应该可以模仿范例编写最简单的 C# 程序了。下面我们回到第 1.1 节的工作场景 1 中，完成工作任务。

【工作过程 1】 创建第一个空的控制台应用程序。

(1) 选择“开始”→“程序”→Microsoft Visual Studio.Net 2005→Microsoft Visual Studio.Net 2005 命令，出现“起始页”界面，如图 1.3 所示。

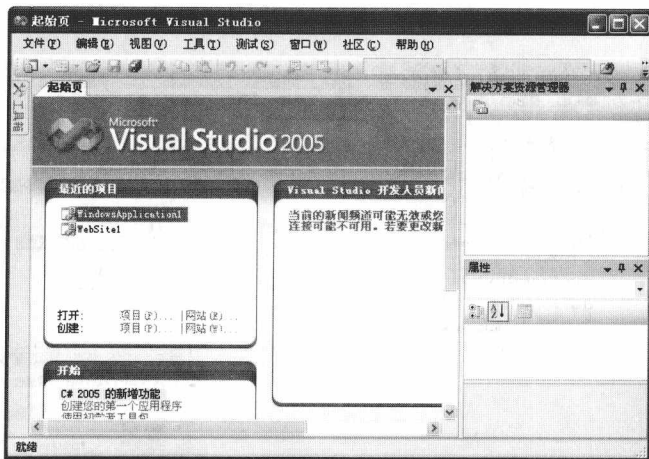


图 1.3 “起始页”界面

(2) 在“最近的项目”中单击“创建：项目”，打开“新建项目”对话框。

(3) 从“新建项目”对话框左侧的“项目类型”中选择 Visual C#，然后从右侧的“模板”中选择“控制台应用程序”。

(4) 在“新建项目”对话框的“名称”文本框中有系统默认的名字 ConsoleApplication1，修改“名称”文本框中的内容为 FirstApplication1，作为该项目的名字，如图 1.4 所示。

(5) 单击“确定”按钮，建立一个项目 FirstApplication1，并进入 Visual Studio .NET 系统，如图 1.5 所示。

【工作过程 2】 编写程序代码。

打开 Program.cs 文件，在 Main() 方法的大括号里输入代码。

```
static void Main()
{
    //在控制台输出“Hello,world”
```

```
Console.WriteLine("Hello,world");  
}
```

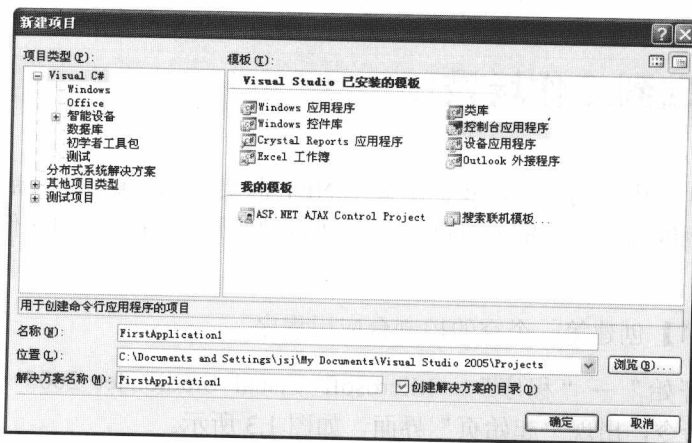


图 1.4 “新建项目”对话框

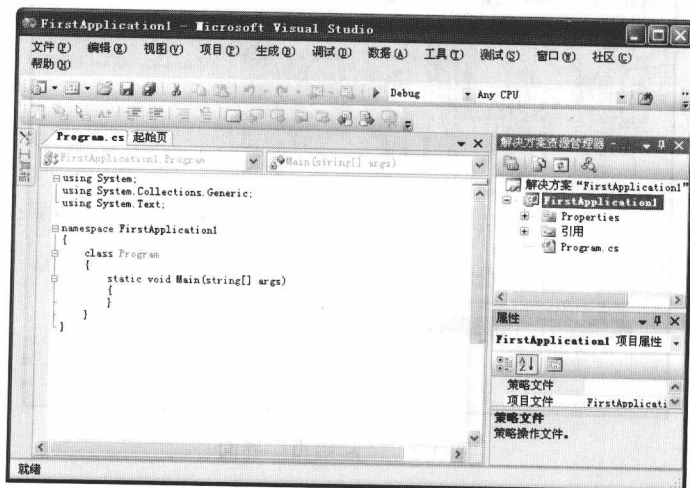


图 1.5 Visual Studio .NET 界面

【工作过程 3】调试执行。

选择“调试”→“开始执行(不调试)”命令，也可以按快捷键 Ctrl+F5，程序调试执行结果如图 1.6 所示。

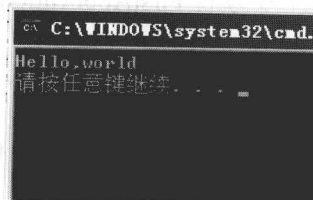


图 1.6 程序调试执行结果



1.6 回到工作场景 2

下面我们再回到第 1.1 节的工作场景 2 中，完成 Windows 应用程序的工作任务。

【工作过程 1】 创建第一个 Windows 应用程序。

- (1) 选择“开始”→“程序”→Microsoft Visual Studio .Net 2005→Microsoft Visual Studio .Net 2005 命令，出现“起始页”界面。
- (2) 在“最近的项目”中单击“新建项目”，打开“新建项目”对话框。
- (3) 从模板中选择“Windows 应用程序”选项，在“名称”文本框中有系统默认的名字 WindowsApplicationx。现在在“名称”文本框中输入 FirstApplicationx，作为该项目的名字。
- (4) 单击“确定”按钮，建立一个 Windows 项目 FirstApplicationx，并进入 Visual Studio .NET 系统，如图 1.7 所示。

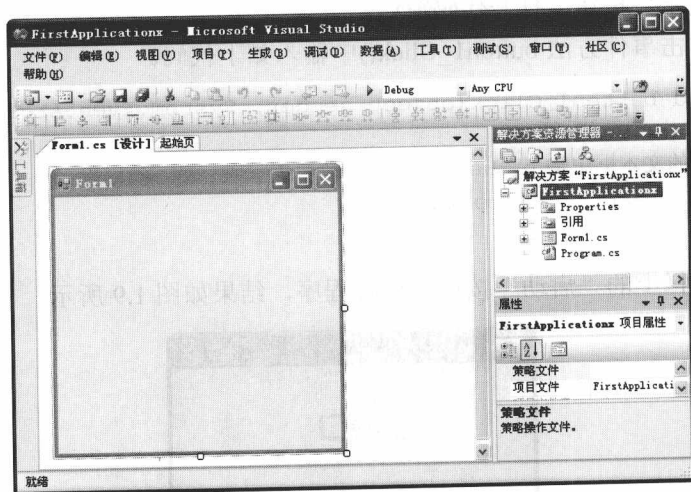


图 1.7 Windows 项目工作界面

【工作过程 2】 设计程序界面并设置控件属性。

- (1) 鼠标移到左边的“工具箱”上，出现“工具箱”窗口。
- (2) 单击“工具箱”窗口上的“Windows 窗口”，出现“Windows 窗口”选项卡。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；再拖一个 textBox 控件到窗体上(上面显示 textBox1)，并摆放好它的位置。
- (3) 单击选中 Button1，然后单击 VS 右下角的“属性”窗格，从属性中找到 Text 属性，它右边的值为 button1。修改值为“显示”，如图 1.8 所示。
- (4) 单击选中 textBox1，然后单击 VS 右下角的“属性”窗格，从属性中找到 Name 属性，它右边的值为 textBox1。这是该控件的名字，可以通过这个名字来控制控件。