



高职高专项目课程系列教材

Visual C#.NET 数据库应用程序开发

Practical Database Programming with
Visual C#.NET

■ 钱冬云 编著



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

图书在版编目(CIP)数据

Visual C#.NET 数据库应用程序开发/钱冬云编著.

—杭州:浙江大学出版社,2010.6

ISBN 978-7-308-07380-6

I. ①V… II. ①钱… III. ①C 语言—程序设计

IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 018260 号

Visual C#.NET 数据库应用程序开发

钱冬云 编著

责任编辑 黄娟琴

文字编辑 李峰伟

封面设计 陈 辉

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州大漠照排印刷有限公司

印 刷 富阳市育才印刷有限公司

开 本 787mm×1092mm 1/16

印 张 18

字 数 427 千

版 印 次 2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷

书 号 ISBN 978-7-308-07380-6

定 价 33.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571) 88925591

序

近年来在政府推动与经济发展需求的刺激下,我国高等职业教育规模有了很大发展。全国职业教育工作会议的召开,又为高职发展迎来了新的历史机遇。然而,我们可以在短短几年内建设起大量被称为“高职”的院校,却无法在短期内形成真正的高职教育,如何凸显特色已成为困扰高职发展的重大课题。

高职发展已由规模扩充进入了内涵建设阶段。内涵建设既需要理论支持,又需要时间积淀,但积极的探索与行动总是有益于这一进程的。已形成的基本共识是,课程建设是高职内涵建设的突破口与抓手。加强高职课程建设的另一个重要出发点是如何让高职生学有兴趣、学有成效。在传统学科知识的学习方面,高职生是难以和本科生相比的。因此,如何开发一套既适合高职生学习特点又能增强其就业竞争能力的课程教材,是高职课程建设面临的另一重大课题。

要有效地解决这些问题,建立能综合反映高职发展多种需求的课程体系,必须进一步明确高职人才培养目标、其课程内容的性质及组织框架。为此,不能仅仅止步于“高职到底培养什么类型人才”的论述,而是要从具体的岗位与知识分析入手。高职专业的定位要通过理清其所对应的工作岗位来解决,而其课程特色应通过特有的知识架构来阐明。也就是说,高职课程与学术性的大学课程相比,其特色不应仅仅体现在理论知识少一些、技能训练多一些,而是要紧紧围绕课程目标重构其知识体系的结构。我们认为,项目课程不失为一个有价值与发展潜力的选择,其历史虽然久远,却可以被赋予新的内涵。

(1) 能力观,即项目课程的目标是培养学生的职业能力。现有高职课程基本上还是知识体系,极少体现这一目标。以职业能力为目标不能只是口号,而是要在各个环节紧紧围绕这一目标来设计课程。比如课程目标的描述,要明确指出预期学生“能够(会)做什么”。能力也不同于操作技能,职业能力更加强调的是在复杂的工作情境中进行分析、判断并采取行动的能力。

(2) 联系观,即要把知识与工作任务之间的联系作为重要课程内容。职业能力的形成并非仅仅取决于获得了大量理论知识,如果这些知识是在与工作任务相脱离的条件下获得的,那么仅仅是些静态的知识,无法形成个体的职业能力。只有能在知识与工作任务之间建立复杂联系的人,才可称为具有职业能力的人。可见,项目课程并非如通常所设想那样只是出于功利目的,而是建立在职业能力形成的联系观基础之上的。

(3) 结构观,即强调对课程结构的整体设计,包括课程体系结构与内容组织结构,因为知识也是影响职业能力形成的重要变量。课程体系结构设计的基本依据是工作体系结构,

内容组织结构设计的基本依据是工作过程中的知识组织关系,其获得的基本手段是工作分析。

(4) 综合观,即综合运用相关操作知识、理论知识来完成工作任务,但项目课程重点关注如何综合运用所获得的操作知识、理论知识来完成工作任务,从而形成在复杂的工作情境中作出判断并采取行动的能力,它也更关注工作任务之间的联系。

(5) 结果观,即以典型产品或服务为载体设计教学活动。通过这种“完整性活动”,学生可获得有工作意义的“产品”,不仅可以增强学生对教学内容的直观感,而且有利于增强学生的成就感。

教材是课程理念的物化,也是教学的基本依据。项目课程的理念要大面积地转化为具体的教学活动,必须有教材作支持。基于这一设想,我们自2004年起,一直致力于与高职院校及教师合作,开发出能体现上述的项目课程理念、符合高职教育水准及特色的专业课程教材,以期对我国高职发展作出贡献。这些教材力图彻底打破以知识传授为主要特征的传统学科课程模式,转变为以工作任务为核心的项目课程模式,让学生通过完成具体项目来构建相关理论知识,并发展职业能力。其课程内容的选取紧紧围绕工作任务完成的需要来进行,同时又充分考虑了高职教育对理论知识学习的需要,并融合了相关职业资格证书对知识、技能和态度的要求。每个项目的学习都要求按以典型产品为载体设计的活动来进行,以工作任务为中心整合理论与实践,实现理论与实践的一体化。为此,有必要通过校企合作、校内实训基地建设等多种途径,采取工学交替、半工半读等形式,充分开发学习资源,给学生提供丰富的实践机会。教学效果评价可采取过程评价与结果评价相结合的方式,通过理论与实践相结合,重点评价学生的职业能力。

在开发新教材的同时,我们也在实验性地进行教学尝试。结果表明,尽管要全面实施项目教学在目前还存在一定困难,如教师能力、实训条件不足等,但这种教学模式的确有利于大大提高学生的学习兴趣与教学质量。学生不仅感受到了知识的应用价值,而且学会了如何应用这些知识。只要教师勇于创新,敢于挑战传统教学模式,其中的许多问题是不难克服的。今后,我们将深化对教学过程的研究,为项目课程实施提供详细案例,同时开发教学辅助材料,以更好地促进项目课程的实施。

由于项目课程教材的结构与内容和传统教材相比差别很大,因此其开发是一个非常艰苦的过程。为了使得这套教材更能符合高职学生的实际情况,我们坚持所有编写任务均由高职教师承担,他们为这套教材的成功出版付出了巨大努力。备感欣慰的是,参与这个项目的高职院校对我们的工作都非常支持,他们不仅组织了大量精干教师和企业专家参与教材开发,而且为我们创造了许多优越条件,没有他们的大力支持,要取得这些成果是难以想像的。在此,还要感谢编委会专家对这个项目的热心支持与精心指导。

实践变革总是比理论创造复杂得多。尽管我们尽了很大努力,但所开发的项目课程教材还是非常有限的。由于这是一项尝试性工作,在内容与组织方面也难免有不到之处,尚需在实践中进一步完善。但我们坚信,只要不懈努力,不断发展和完善,最终一定会实现这一目标。

石伟平 徐国庆

2009年11月于华东师范大学

前 言

PREFACE

高职高专教育是以就业为导向,以培养应用型技术人才为目标的职业技术教育。本书结合高职高专院校的教学特色,遵循“以就业为导向”的原则,结合软件开发岗位的实际需求,设计教材内容。以“实用”为基础,以“必需”为尺度,选取理论知识;采用项目式教学,结合各项工作任务,突出培养应用能力和解决实际问题的能力。

C#语言是一种全新的面向对象的语言,是Microsoft公司推出的新一代开发环境Microsoft .NET的核心语言,它的出现代表了程序设计语言下一步的发展方向。C#语言可以利用.NET和开发环境的所有新特性,开发各种功能丰富的应用程序。

本书以实用为中心,以提高读者的数据系统设计和开发能力为宗旨,根据数据库应用程序开发的要求,采用了以“高校学生学籍管理系统”为工程。结构紧凑,内容承上启下,共十二个项目(较大的项目又分为几个模块)。项目一介绍了高校学生学籍管理系统的开发环境;项目二详细介绍了开发系统的需求分析和总体设计;项目三介绍了如何创建数据库、数据表、视图和存储过程等内容;项目四介绍了数据库应用程序开发的基础,连接数据库;项目五介绍了设计软件的各类界面;项目六到项目八介绍了高校学生学籍管理系统中各类数据的录入、删除、修改、查询和浏览功能的实现;项目九介绍了报表的设计;项目十概述了制作帮助系统的过程;项目十一介绍了在线查询功能的实现;项目十二将开发完成的高校学生学籍管理系统制作成安装程序。通过逐个项目工作任务的完成,开发了完整的“高校学生学籍管理系统”。利用项目化组织内容,有利于培养和提高高职高专学生的技术应用能力,缩小在校学习与生产岗位需求之间的距离。

教材结构新颖,可学、可用、可查。每个项目包含教学目标、工作任务、相关实践知识、相关理论知识,部分项目还设置了拓展知识,以便学有余力的学生学习。学生通过完成十二个项目的工作任务,达到构建知识的目的;通过完成课后实践题,达到巩固所学知识、激发创新意识,提高分析问题和解决问题能力的目的。

本书由浙江工贸职业技术学院钱冬云编著,黎浩宏、陈欲劭、戴国森参编。由于计算机科学技术发展迅速,又因作者自身水平和编写时间所限,书中难免有错误及不妥之处,恳请广大读者给予批评指正,提出宝贵意见,作者邮箱:qiandongyun@163.com。

编 者

2010.6

目 录

CONTENTS

项目一 高校学生学籍管理系统的开发环境	1
项目二 高校学生学籍管理系统分析与设计	12
模块一 软件需求分析 / 12	
模块二 总体设计 / 17	
项目三 创建高校学生学籍管理系统数据库	31
项目四 高校学生学籍管理系统数据库连接	41
模块一 自动配置数据源 / 41	
模块二 编程方式连接数据库 / 51	
项目五 高校学生学籍管理系统界面设计	67
模块一 设计主界面窗体 / 67	
模块二 设计用户操作模块界面 / 87	
项目六 高校学生学籍管理系统信息录入界面设计	105
项目七 高校学生学籍管理系统数据浏览界面设计	138
模块一 表格式浏览系部信息 / 138	
模块二 树形结构浏览各系部的专业和班级信息 / 151	
模块三 信息查询界面 / 167	
项目八 高校学生学籍管理系统数据修改界面设计	194
模块一 修改专业信息 / 194	

模块二	修改部门信息 / 197	
模块三	休学信息管理 / 201	
模块四	转学信息管理 / 210	
项目九	高校学生学籍管理系统数据报表打印设计	228
模块一	将休学表数据导出到 Excel 表 / 228	
模块二	调用 Office 打印转学学生信息 / 232	
模块三	利用水晶报表打印课程信息 / 237	
项目十	高校学生学籍管理系统帮助文件	247
项目十一	高校学生学籍管理系统在线查询	256
模块一	在线查询 / 256	
模块二	在线数据录入 / 267	
项目十二	创建高校学生学籍管理系统安装程序	272
附录	软件需求规格说明书(模板)	278
参考文献		280

项目一 高校学生学籍管理系统的开发环境

一、教学目标

1. 学会使用开发高校学生学籍管理系统的工具；
2. 熟悉开发环境。

二、工作任务

在使用高校学生学籍管理系统程序时,经常需出现一些提示信息,便于使用者熟悉操作软件。创建一个提示信息应用程序,提示信息如图 1-1 的左图所示;当用户单击【提示信息】按钮时,出现“欢迎使用 C#.NET 语言开发软件!”的提示框,如图 1-1 的右图所示。

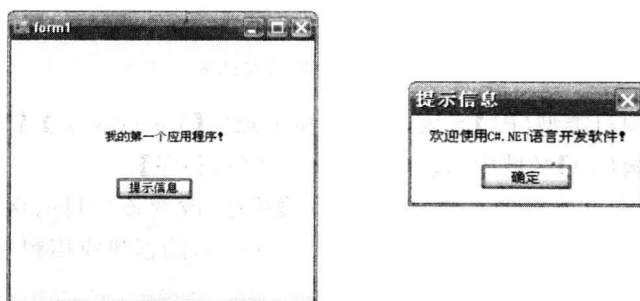


图 1-1 显示提示信息程序

三、相关实践知识

1. 启动 Visual Studio 2005 开发工具

单击【开始】|【程序】|【Microsoft Visual Studio 2005】|【Microsoft Visual Studio 2005】,将出现如图 1-2 所示界面,该界面为 Visual Studio 2005 起始页。

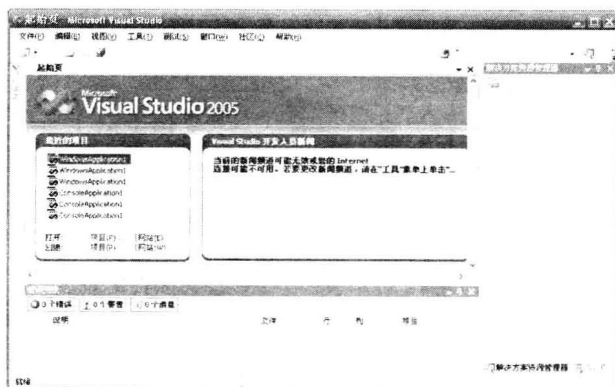


图 1-2 Visual Studio 2005 起始页

说明:

本书中“高校学生学籍管理系统”在 Windows XP sp2 系统下开发,由于操作系统的不同,个别步骤有可能会有不同。

2. 创建 C# .NET Windows 应用程序

(1) 在 Visual Studio 2005 集成开发环境(IDE)中,选择【文件】|【新建】|【项目】菜单项,将会弹出【新建项目】对话框,如图 1-3 所示。

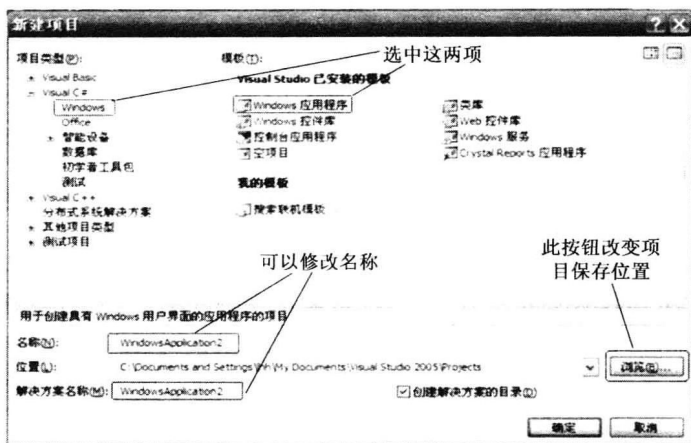


图 1-3 新建项目对话框

(2) 在左侧的【项目类型(P)】区域,选择树型节点中【Visual C#】|【Windows】节点。

(3) 在右侧【模板(T)】区域中,选择【Windows 应用程序】。

(4) 在【位置】对应文本框的右侧,单击【浏览】按钮,改变该项目的保存位置。设置项目保存位置后,单击【确定】按钮,此时将出现如图 1-4 所示的创建应用程序的初始界面。

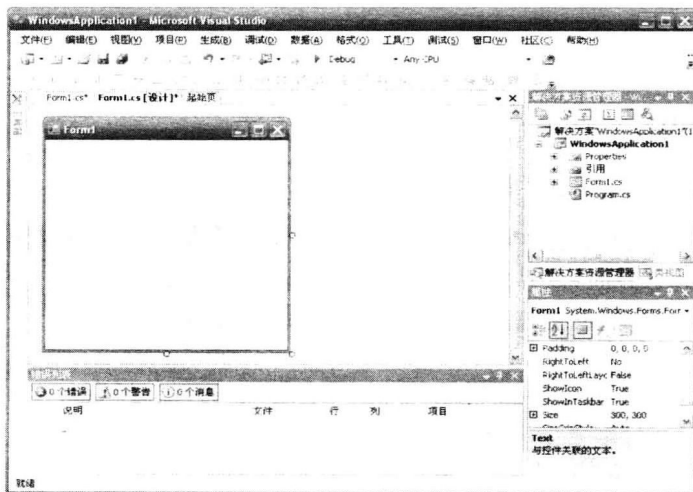


图 1-4 创建应用程序的初始界面

3. 向窗体添加控件

在图 1-4 所示的 Form1 窗体中,添加一个 Label(标签)控件和一个 Button(按钮)控件,步骤如下。

(1) 显示【工具箱】。默认情况下, Visual Studio 2005 C#.NET 集成开发环境界面中, 【工具箱】窗口是隐藏的, 将鼠标指向左侧的【工具箱】图标, 【工具箱】窗口自动显示, 如图 1-5 所示。

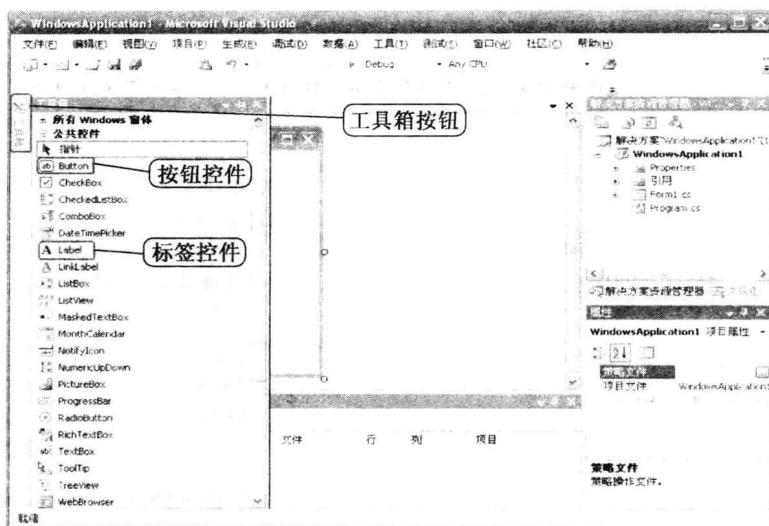


图 1-5 显示【工具箱】窗口

(2) 展开【公共控件】节点, 如图 1-5 所示。选中【Button】控件, 按住鼠标左键, 将【Button】控件拖到 Form1 窗体上, 放到合适的位置; 然后用相同的方法, 将【Label】控件拖到 Form1 窗体上合适的位置。界面的设计效果如图 1-6 所示。

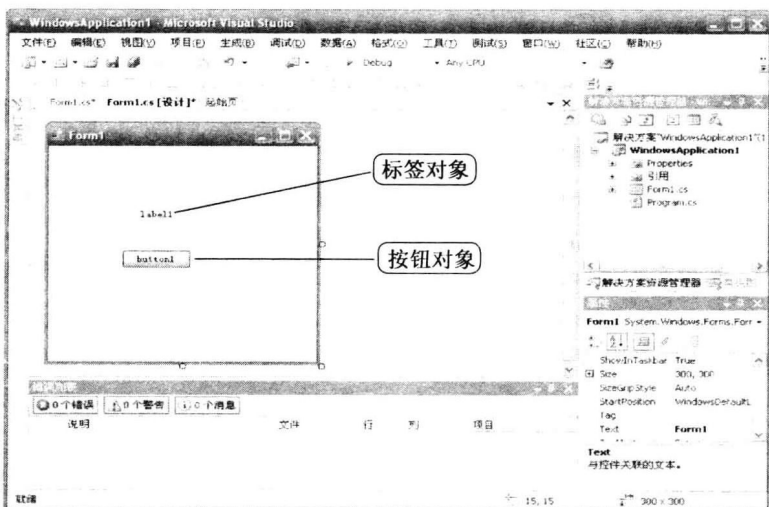


图 1-6 窗体设计效果图

4. 设置标签和按钮的属性

由于在默认情况下, 标签和按钮对象上显示的都是英文, 如要设置这两个对象显示的信息为指定的中文, 就必须分别设置其对应的属性, 方法如下。

(1) 在 Form1 窗体上选中 label1 标签(因为该标签的名称为 label1),此时在标签的周围有一个矩形虚线框。

(2) 在 Visual Studio 2005 C#.NET 集成开发环境中右下角的属性窗口中,拖动属性窗口右侧垂直滚动条,选中【Text】属性,在【Text】属性右侧输入“我的第一个应用程序!”,如图 1-7 所示。

(3) 设置按钮的显示信息和标签的方法类似。

控件对象的其他属性设置方法也按相似的步骤进行,以后不再赘述。

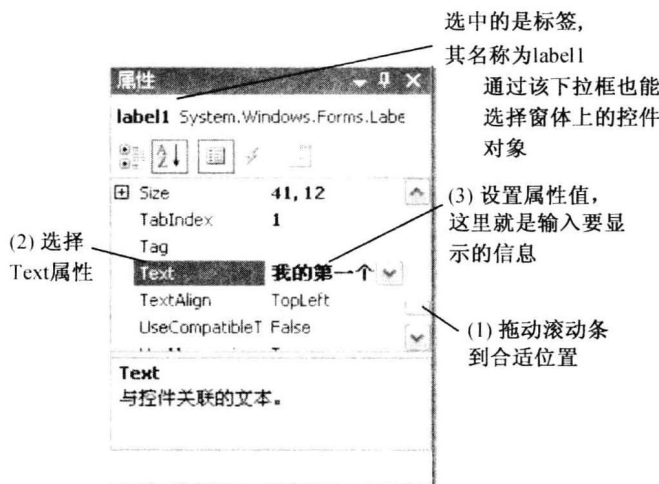


图 1-7 属性窗口

说明:

① 如果没出现【属性】窗口,单击【视图(V)】|【属性窗口(W)】菜单,将显示【属性】窗口。

② 在【属性】窗口中,【Text】属性含义为控件上显示的文本内容。

5. 编写单击按钮事件处理方法

在 Form1 窗体上,双击【button1】按钮,Visual Studio 2005 C#.NET 开发环境将进入代码编辑器环境,光标自动定位在【button1】按钮的单击事件处理方法 button1_Click 中,只要输入如图 1-8 所示的代码即可。

说明:

button1_Click 为【button1】按钮鼠标单击事件处理方法。在运行该项目时,只要单击该按钮,button1_Click 事件处理方法中的所有代码(即该事件处理方法下面的“{”和“}”内的代码,这里只有一行代码)会被计算机自动运行;而代码 `MessageBox.Show("欢迎使用 C#.NET 语言开发软件!", "提示信息")` 的作用是显示如图 1-1 右侧的信息框,将信息框的标题设置为“提示信息”,信息框显示的信息设置为“欢迎使用 C#.NET 语言开发软件!”。注意,该行代码最后要有分号。



图 1-8 程序代码编辑器

6. 编译与调试运行程序

单击【调试(D)】菜单的【启动调试】菜单项或工具栏中的绿三角 ，或者按下 F5 键，启动程序调试。运行应用程序时，出现“我的第一个应用程序!”窗体。当单击【提示信息】按钮时，验证出现“欢迎使用 C#.NET 语言开发软件!”的提示框。

四、相关理论知识

1. Visual Studio 2005 简介

Visual Studio 2005 是 Microsoft(微软公司)推出的新一代软件开发工具，它包括 C#、Visual Basic、Visual C++ 和 .NET Framework。 .NET Framework 称为 .NET 框架，它是一种托管的、类型安全的代码执行环境。Visual Studio 6.0 之前版本的语言编写程序可以在 Windows 环境中直接运行，而 Visual Studio 2003 和 Visual Studio 2005 是基于 .NET 环境，所以用这两个开发工具的语言编写的程序必须在 .NET Framework 环境下运行。为有别于 Visual Studio 6.0 之前的语言，称之为 C#.NET，Visual Basic.NET 和 Visual C++ .NET 语言。用 .NET 语言编写的程序必须在机器上安装好 .NET Framework 后方可运行。 .NET Framework 可以到微软官方网站上下载，Visual Studio 2005 .NET Framework 是 2.0 版本，而 Visual Studio 2003 .NET Framework 是 1.0 版本。

.NET Framework 包括 CLR(Common Language Runtime, 公共语言运行库)和 .NET Framework 类库。CLR 负责管理和执行 .NET 框架代码，符合 Visual Studio .NET 编译器规则的代码在执行时需要 CLR 的支持，这些代码称为托管代码(Managed Code)。用 .NET 语言编写的程序被编译时，应用程序不是被直接编译成二进制代码，而是被编译成 MSIL(Microsoft Intermediate Language, 微软中间语言)，称为 MSIL 代码。在 CLR 中运行时，当应用程序加载后，CLR 可根据需要将其再次编译成可执行代码。这种编译称为 JIT(Just-in-Time, 即时编译)。

.NET Framework 类库是 Microsoft 已经编写好的一个内容丰富的受管制的类代码集合。它可以完成以前要通过 Windows API 来完成的绝大多数任务，使程序员从烦琐的 API

函数调用中解脱出来,专心于应用程序事务的处理。

.NET Framework 使我们可以更容易地开发基于 Web 和基于 Windows 的应用程序,允许使用不同的语言一起进行开发工作。

2. C# 概述

C# 语言从 C 和 C++ 发展而来,其语法和 C、C++ 和 Java 的很多语法都相同。它汲取了包括 C、C++、Java 在内的多种语言的精华,是一种简单、完备、类型安全和完全面向对象的高级程序设计语言。C# 是专门为 .NET 应用而开发的语言,与 .NET 框架完美结合。使用 C# 可以创建传统的 Windows 客户端应用程序、XML Web Service、分布式组件、客户端和服务端应用程序和数据库应用程序等。

3. Visual Studio 2005 集成开发环境

(1) 起始页

起始页界面如图 1-2 所示,包括【最近的项目】、【开始】、【Visual Studio 标题新闻】和【Visual Studio 开发人员新闻】四项内容。在【最近的项目】中可以方便地打开或新建项目和网站,在【开始】中可以选择感兴趣的内容进行入门学习,另外两项可以查看相应的新闻。如果界面上没有【起始页】,若要显示【起始页】,可以选择【视图(V)】|【其他窗口(E)】|【起始页(G)】菜单。

(2) 【解决方案资源管理器】窗口

【解决方案资源管理器窗口】如图 1-9 所示,用于管理和监控解决方案中的项目,提供项目及其文件的有组织的视图,并且提供对项目 and 文件相关命令的便捷访问。如果集成环境中没有出现该窗口,可通过选择【视图(V)】|【解决方案资源管理器(W)】菜单来显示该窗口。在【解决方案资源管理器窗口】中包含有 6 个按钮,常用的有 4 个按钮,分别是属性、显示所有文件、查看代码和查看设计器按钮。



图 1-9 解决方案资源管理器窗口

(3) 【工具箱】窗口

【工具箱】窗口如图 1-10 所示。初始时,【工具箱】是隐藏在 Visual Studio 2005 集成开发环境窗口的左侧。如果在集成环境中没有出现该窗口,可以通过选择【视图】|【工具箱】菜单来显示该窗口。

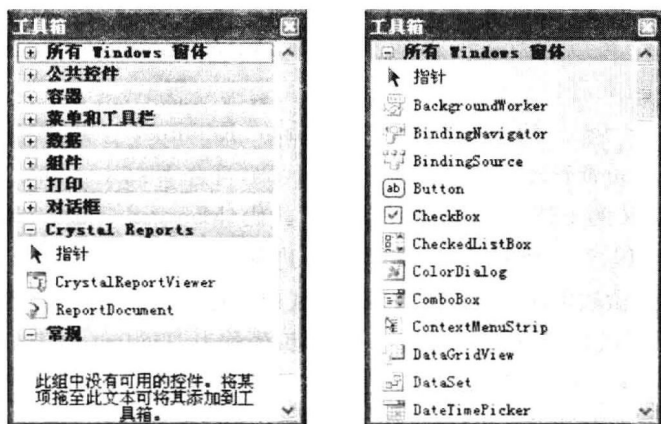


图 1-10 【工具箱】窗体

在【工具箱】窗口中,用树型视图存放了开发 .NET Windows 应用程序和开发 ASP.NET 网页的常用控件。【工具箱】窗口已经为软件开发人员将 Visual Studio 2005 集成开发环境提供的控件分好了类,软件开发人员可以非常方便地利用工具箱中的各种控件,快速地开发应用程序。

(4) 属性窗口

如图 1-11 所示的【属性】窗口可以编辑和查看文件、项目和解决方案的属性,常用在设计 Windows 应用程序窗体时查看和修改被选中的控件对象的属性和事件,如图 1-12 所示。

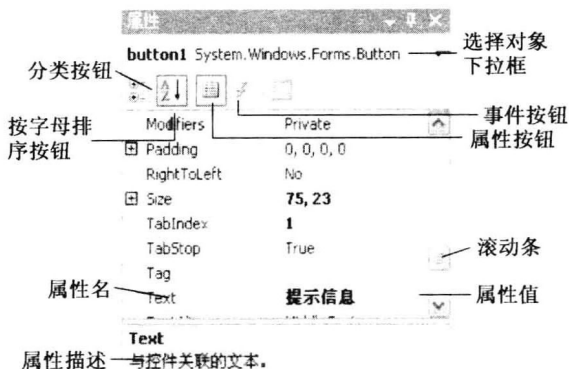


图 1-11 【属性】窗口



图 1-12 【属性】窗口中添加事件

(5) 设计器窗口和代码编辑器窗口

设计器窗口用来设计应用程序的界面,如图 1-13 的左图所示,可以从【工具箱】中把各种控件添加到窗体上来,并安排它们的布局。代码编辑器窗口如图 1-13 的右图所示。设计器窗口与代码编辑器窗口切换有两种方法:

① 将鼠标移至窗体上,单击右键,选择【查看代码】菜单。

② 单击【解决方案资源管理器】|【查看代码】按钮。

同样,从代码编辑器窗口切换到【窗体设计器】窗口,也可以有两种方法:

① 在代码编辑器上,单击右键,选择【查看设计器】菜单。

② 单击【解决方案资源管理器】窗口的【查看设计器】按钮。

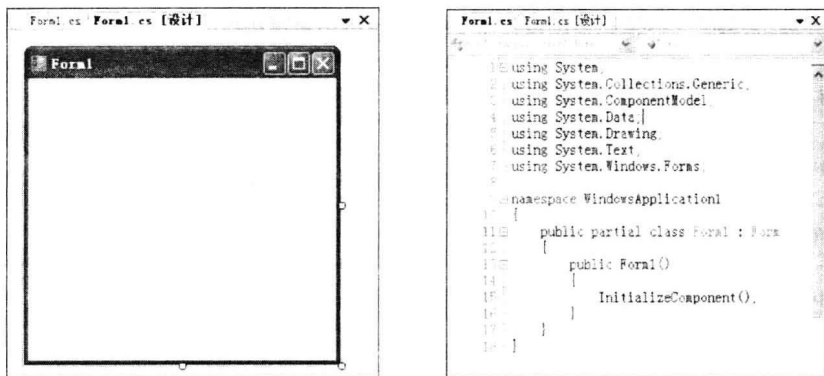


图 1-13 【窗体设计器】窗口

说明：

默认情况下,代码编辑器窗口没有显示行号,要设置行号,操作步骤如下。

① 选择【工具(T)】|【选项(O)】选项,出现如图 1-14 所示界面。

② 在左侧树型视图中,选择【文本编辑器】节点,选中【C#】,在右侧的下方,选中复选按钮【行号】,单击【确定】按钮即可。

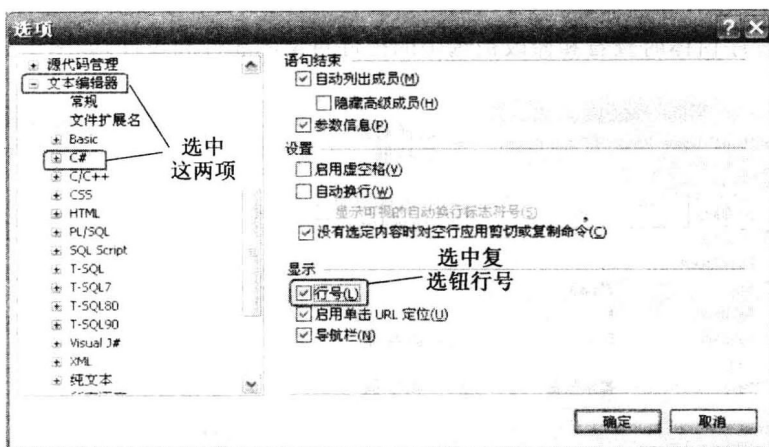


图 1-14 【选项】对话框窗体

(6) Visual C#.NET Windows 项目相关文件

创建项目后,Visual C#.NET 将会自动生成解决方案名的一个文件夹,在该文件夹下,包含 Visual C#.NET Windows 项目产生的相关文件和文件夹。详细说明如表 1-1 所示。

表 1-1 项目产生的文件和文件夹

文件(夹)名称	说 明
bin 文件夹	包含 debug 子文件夹,含有 .exe 和 .pdb 文件,.exe 文件是生成的可执行文件,.pdb 文件包含完整的调试信息
obj 文件夹	包含 debug 子文件夹,含有编译过程中生成的中间代码
properties 文件夹	此文件包含程序集属性的设置
program.cs	应用程序文件,包含应用程序代码
Form1.cs	窗体代码文件,包含窗体代码
Form1.Designer.cs	与窗体及控件有关的文件,该文件中的代码是用户设计界面时由系统自动产生的,一般情况下不要改动
WindowsApplication1.csproj	项目文件
WindowsApplication1.csproj.user	用户配置文件
WindowsApplication1.sln	解决方案文件

(7) Visual Studio 2005 环境的三种状态

Visual Studio 2005 环境可以有三种状态,当用户在设计窗体界面和编写代码时,处于设计状态;当用户在运行项目程序时,处于运行状态, Visual Studio 2005 环境窗口的标题栏上有“正在运行”字样出现;当用户在调试项目程序时, Visual Studio 2005 环境窗口的标题栏上有“正在调试”字样出现。显然,在运行和调试项目程序时,用户不可以修改代码和设计窗体界面,当修改代码时,会出现错误提示框,如图 1-15 所示。

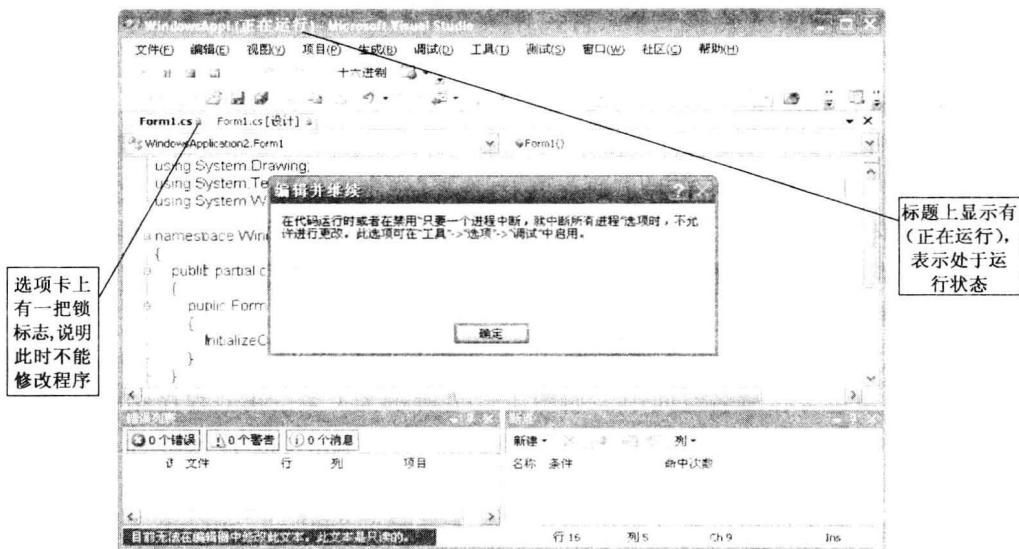


图 1-15 在运行时修改代码

(8) 设计程序运行时窗体上的控件获得焦点的顺序

当运行程序时,如果窗体上的控件很多,用键盘操作将是比较方便的,按 Tab 键,就可以将焦点自动定位到后续的控件上。要设置控件获得的焦点顺序,就必须设置控件的 TabIndex 属性,该属性是控件的 Tab 顺序索引。

在添加控件时,该属性值将由 Visual C#.NET 自动分配一个值,其序号是由 0 开始顺序分配的,如添加的第一个控件的 TabIndex 属性值为 0,第二个控件的 TabIndex 属性值就为 1,以此类推。在运行程序时,TabIndex 属性值为 0 的控件将自动获得焦点,按 Tab 键,焦点将会定位到 TabIndex 属性值为 1 的控件,以此类推。

设置控件 TabIndex 属性值可以在属性窗口中进行设置,但非常烦琐。使用布局工具栏上【Tab 键顺序】按钮进行设置,使用起来非常方便。

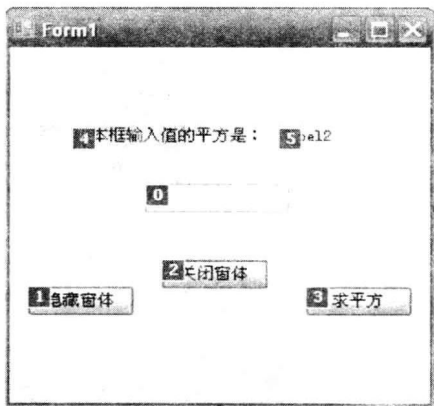


图 1-16 设置窗体上控件的 TabIndex 属性值

① 选中窗体或窗体上的任何控件,单击【布局】工具栏上【Tab 键顺序】按钮(也可选择【视图(V)】|【Tab 键顺序(B)】菜单),在每个控件左上角上将出现蓝底的数字,该数字即为对应控件

件的 TabIndex 属性值,如图 1-16 所示界面。

② 第一次单击程序运行时要获得焦点的控件,然后按照控件获得焦点的次序依次单击对应的控件。设计完成后,单击【布局】工具栏上【Tab 键顺序】按钮(也可选择【视图(V)】|【Tab 键顺序(B)】菜单),当每个控件左上角上出现的蓝底数字消失时即可。

说明:

有一些控件在程序运行时是不能获得焦点的,如标签控件。能获得焦点的控件都有一个属性,即“TabStop”属性,不能获得焦点的控件没有这个属性。只有将“TabStop”属性设置为 True 的控件,在运行程序时,按 Tab 键才能获得焦点。控件的“TabStop”属性值默认为 True。

(9) 锁定控件和窗体

当完成设计窗体后,锁定控件或窗体,设置不允许改变控件在窗体上的位置。选择窗体或窗体上的任何控件,右击鼠标,在弹出的菜单上,选择【锁定控件】菜单即可。此时可以看到选中的窗体或窗体上的控件左上角有一个锁定图标,这时将不能改变控件在窗体上的位置。解除锁定的方法与锁定的方法相同,不再赘述。默认情况下,窗体和控件都没有被锁定。

4. 命名空间

命名空间是 C# 代码的基本组织形式,是组织应用程序的一种结构,类、结构等各种类型定义都必须包含在命名空间中。如图 1-17 所示为 C#.NET 程序结构。命名空间在默认情况下会自动创建,命名空间的名字就是项目名,由关键字 namespace 声明,如图中的第 8 行代码“namespace WindowsApplication1”,其中 WindowsApplication1 是命名空间的名字。命名空间中的代码都必须包含在一对大括号“{”和“}”之中。“{”表示开始,“}”表示结束,且大括号必须成对使用,少了任何一个都会出错。第 9 行的“{”表示命名空间由此开始,第 41 行的“}”表示命名空间到此结束,而第 10~40 行的代码都属于命名空间 WindowsApplication1。

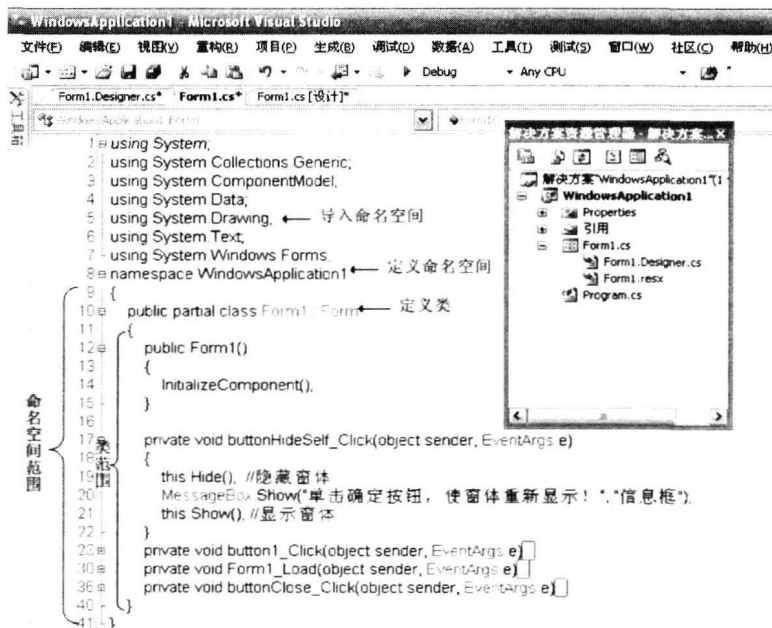


图 1-17 C#.NET 程序结构