



“十二五”江苏省高等学校重点教材
全国高等职业教育规划教材

C# 可视化程序设计 案例教程

第3版

主编 刘培林 史荧中
参编 黄 翀

C#

电子课件、源代码下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS



“十二五”江苏省高等学校重点教材（编号：2013-1-028）

全国高等职业教育规划教材

C#可视化程序设计案例教程

第3版

主编 刘培林 史荧中

参编 黄 翀

主审 刘德强



机械工业出版社

本书共 10 章, 前 2 章介绍 Windows 应用程序开发入门、窗体与控件; 第 3、4 章讲述 C# 语言与菜单、工具栏、状态栏、对话框; 第 5、6 章为本书的重点及难点, 讲述 Visual Studio 数据库开发环境与数据库应用程序设计; 第 7 章讲述 Visual Studio 高级控件; 第 8 章讲述图形绘制; 第 9 章讲述网络通信编程的基础知识; 第 10 章介绍安装程序的制作方法和应用程序的部署步骤。

全书贯彻“理实一体化”的教学理念, 以学生档案管理系统为载体, 将项目开发分解为若干相对独立的工作任务。工作任务与相关理论知识交互配合, 既是对理论知识的延伸与拓展, 又是对理论知识掌握程度的检验。

本书可作为高职高专院校计算机、电子等专业的教材, 也可作为可视化程序设计的入门语言教程, 还可作为计算机与应用工程技术人员的培训用书或自学参考书。由于书中有大量丰富而实用的数据库应用程序, 故本书还可作为计算机软件程序设计人员的技术参考书。

本书配有授课电子课件和源代码, 需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载, 或联系编辑索取 (QQ: 1239258369, 电话: 010-88379739)。

图书在版编目 (CIP) 数据

C#可视化程序设计案例教程 / 刘培林, 史荧中主编. —3 版. —北京: 机械工业出版社, 2014.9
全国高等职业教育规划教材
ISBN 978-7-111-48298-7

I. ①C… II. ①刘… ②史… 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 (2014) 第 241185 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 鹿 征 责任校对: 张艳霞

责任印制: 刘 岚

涿州市京南印刷厂印刷

2015 年 1 月第 3 版·第 1 次印刷

184mm×260mm·17.75 印张·438 千字

0001—3000 册

标准书号: ISBN 978-7-111-48298-7

定价: 37.80 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面防伪标均为盗版

金书网: www.golden-book.com

前 言

Visual Studio 开发平台是目前许多软件公司使用的重要开发工具,而 C#程序设计语言是计算机相关专业学生应掌握的一门语言。通过对 Visual Studio 开发平台和 C#程序设计语言的学习,读者可以了解 Visual Studio 的开发环境、工程、面向对象、事件驱动程序设计的概念,掌握 C#语言的语法、程序结构、编程方法,掌握 Visual Studio 基本控件的属性、方法、事件及用户程序界面的设计方法,掌握使用界面控件、ADO.NET 对象及其绑定控件设计一个信息管理系统的方法。本书按系统分析员的要求,使用 Visual Studio 开发平台及 C#语言设计窗体界面、编写程序代码、进行程序运行与调试,为读者成为一名应用软件程序员打下必要的基础。

本教材具有以下特点。

- 1) 以学生档案管理系统为载体,采用边讲授知识点边设计模块程序的项目式教学法,讲学做一体。
- 2) 精心设计工作任务,将学生档案管理系统分解为若干相对独立的工作任务。
 - 从形式上看,工作任务是知识与技能的结合。每个工作任务都包含 6 个完整的部分,其中项目描述、项目设计、项目实施、项目测试 4 个部分对应着企业软件开发过程中的需求分析、设计、实施、测试 4 个阶段,而相关知识、项目小结两个部分又兼顾到了教学的实际需要。
 - 从内容及编排上来看,本书的工作任务源自真实项目,与相应理论知识互为补充,难度上循序渐进,适于学习。
- 3) 以“学生档案管理系统”程序作为主要例题与实验题编写,在这个管理信息系统中融入了编者开发程序的大量经验与体会,希望能通过这本实用性很强的教材,使读者能很快地掌握使用 Visual Studio 开发应用程序的基本方法。
- 4) 根据信息管理系统编程的需要,对常用的控件加以重点介绍,并用实例说明控件的属性、方法与事件及具体的使用方法,以避免将教材编成一本 Visual Studio 的说明书,只罗列所有控件的属性、方法与事件。
- 5) 所有应用程序的界面设计均采用商业化应用程序的风格,在课堂教学中就开始培养学生掌握编写商业化应用程序的设计方法和思路。
- 6) 每章后有小结,并配有一定的习题与实验题,便于教师教学和学生自学。各章内容充实,安排合理,衔接自然。

本教材的此次修订是第 2 版修订,修订内容主要是结合物联网技术的发展要求增加了网络通信的知识,同时为了满足不同学校数据库教学使用的多样性要求采用了 SQL Server 2005 数据库。

本书由无锡职业技术学院刘培林、史荧中主编,中国船舶重工集团公司第 702 研究所黄翀参与编写,无锡职业技术学院刘德强主审。李萍、杨文珺、颜惠琴等老师在本书的编写中给予了许多建议,特别是教学中总结出的优秀建议,在此谨致谢意。在本书的编写过程中,参考了目前国内比较优秀的有关 C#程序设计方面的书籍,在此谨向有关作者表示感谢。

由于作者水平有限,书中难免会有疏漏和错误之处,恳请读者批评指正。

编 者

目 录

前言

第 1 章 Windows 应用程序开发入门	1
1.1 C#概述	1
1.1.1 C#与.NET 框架的关系	1
1.1.2 .NET Framework 框架概述	1
1.1.3 公共语言运行库	2
1.2 C#应用程序类型	3
1.3 可视化程序设计	4
1.3.1 面向对象的程序设计	4
1.3.2 可视化程序设计概述	5
1.3.3 事件驱动的程序设计	6
1.4 Visual Studio 2010 介绍	6
1.4.1 安装 Visual Studio 2010	6
1.4.2 Visual Studio 2010 集成开发环境介绍	7
1.4.3 Windows 应用程序的开发步骤	10
工作任务 1 熟悉系统开发环境：欢迎使用 VS 2010 开发平台	11
本章小结	12
习题 1	12
实验 1	12
第 2 章 窗体与控件	13
2.1 属性、方法和事件	13
2.1.1 属性	13
2.1.2 方法	14
2.1.3 事件	14
2.2 窗体	15
2.2.1 窗体的主要属性	15
2.2.2 窗体的主要事件	17
2.3 常用控件	18
2.3.1 控件命名	18
2.3.2 标签 (Label)	19
2.3.3 文本框 (TextBox、RichTextBox 和 MaskedTextBox)	19
2.3.4 按钮 (Button)	21
2.3.5 列表框 (ListBox)	22
2.3.6 组合框 (ComboBox)	24
2.3.7 单选按钮 (RadioButton)	24

2.3.8	复选框 (CheckBox)	26
2.3.9	图片框 (PictureBox)	26
2.3.10	分组框 (GroupBox)	27
2.3.11	定时器控件 (Timer)	27
2.4	控件布局	27
2.4.1	调整控件的位置和大小	28
2.4.2	控件的对齐	28
2.4.3	调整控件的间距	28
工作任务 2	用户登录程序设计	28
工作任务 3	班级管理程序设计	31
工作任务 4	学生档案查询程序设计	33
本章小结		37
习题 2		37
实验 2		38
第 3 章	C#程序设计语言	39
3.1	C#程序组成	39
3.1.1	类	40
3.1.2	类代码	41
3.1.3	代码行书写规则	42
3.2	C#的数据类型、变量、常量与表达式	44
3.2.1	数据类型	45
3.2.2	常量	46
3.2.3	变量	47
3.2.4	运算符与表达式	48
3.2.5	C#中常用公共类及其函数	52
3.3	程序结构与流程控制语句	59
3.3.1	程序的 3 种基本结构	59
3.3.2	分支程序	60
3.3.3	循环语句	65
3.4	数组	67
3.4.1	数组概述	68
3.4.2	一维数组的定义与引用	70
3.4.3	二维数组的定义与引用	72
3.4.4	数组列表 (ArrayList)	73
3.4.5	对数组或数组列表使用 foreach	77
3.5	类与对象	79
3.5.1	基本概念	79
3.5.2	类	80
3.5.3	对象	82

工作任务 5 学生成绩评定模块设计	83
工作任务 6 学生信息管理模块设计	86
本章小结	90
习题 3	91
实验 3	92
第 4 章 菜单、工具栏、状态栏与对话框	95
4.1 菜单设计	95
4.1.1 主菜单	95
4.1.2 上下文菜单	99
4.2 工具栏设计	101
4.2.1 创建工具栏	101
4.2.2 工具栏的属性	102
4.2.3 工具栏的事件	103
4.3 状态栏设计	105
4.4 对话框设计	106
4.4.1 对话框的属性	106
4.4.2 对话框的应用	107
4.5 MDI 多窗体程序设计	109
4.5.1 创建 MDI 主窗体	109
4.5.2 建立 MDI 子窗体	109
工作任务 7 创建文本编辑器	110
工作任务 8 学生档案管理系统窗体设计	113
本章小结	116
习题 4	117
实验 4	117
第 5 章 数据库应用程序的可视化设计	118
5.1 数据库基础	118
5.1.1 数据库基本知识	118
5.1.2 关系数据库的基本概念	119
5.1.3 学生档案管理系统数据库	120
5.1.4 创建案例数据库	123
5.1.5 结构化查询语言 (SQL) 简介	126
5.2 类型化数据集	134
5.2.1 利用服务器资源管理器建立数据连接	134
5.2.2 类型化数据集的创建	136
5.2.3 类型化数据集的参数化查询	137
5.2.4 创建学生档案管理系统的数据集	138
5.3 数据库应用程序的结构与设计步骤	139
5.3.1 数据库应用程序结构	139

5.3.2 数据库应用程序的设计步骤	139
5.4 数据源控件与数据访问窗体控件	140
5.4.1 BindingSource 控件	140
5.4.2 BindingNavigator 控件	141
5.4.3 DataGridView 控件	143
5.4.4 Label 控件	147
5.4.5 TextBox 控件	147
5.4.6 ListBox 控件	147
5.4.7 ComboBox 控件	148
5.5 报表	148
5.5.1 报表简介	148
5.5.2 使用报表的一般步骤	149
工作任务 9 系部编码表维护 (类型化数据集应用)	152
工作任务 10 班级编码表维护 (窗体控件综合应用)	155
工作任务 11 学生档案查询 (数据集综合应用)	158
工作任务 12 学生档案统计 (报表应用)	160
工作任务 13 学生档案打印 (报表应用)	163
本章小结	164
习题 5	165
实验 5	166
第 6 章 ADO.NET 数据库访问技术	169
6.1 ADO.NET 数据库访问技术	169
6.1.1 ADO.NET 主要组件	169
6.1.2 ADO.NET 访问数据库的方式	170
6.2 ADO.NET 常用对象及应用	171
6.2.1 Connection 对象	172
6.2.2 Command 对象	173
6.2.3 DataReader 对象	176
6.2.4 DataAdapter 对象	178
6.2.5 DataSet 对象	180
6.2.6 CommandBuilder 对象	184
6.2.7 DataView 对象	185
工作任务 14 用户登录程序设计 (续) Command 对象应用	186
工作任务 15 系部编码表维护 (用 DataReader、Command 对象)	188
工作任务 16 系部编码表维护 (用 DataSet、DataAdapter、Command-BUILDER 对象)	191
工作任务 17 设计学生档案查询程序	193
工作任务 18 设计学生档案录入程序	200
工作任务 19 设计学生档案维护程序	205
本章小结	209

习题 6	210
实验 6	210
第 7 章 C#窗体应用程序高级控件	211
7.1 日期控件 (MonthCalendar)	211
7.1.1 MonthCalendar 控件	211
7.1.2 DateTimePicker 控件	212
7.2 树形控件 (TreeView)	214
7.3 分页控件 (TabControl)	216
7.4 进度条控件 (ProgressBar)	218
7.5 列表控件 (ListView)	220
工作任务 20 校历数据表录入程序设计	224
工作任务 21 用 TreeView 控件设计学生档案查询程序	226
工作任务 22 用 ListView 和 ProgressBar 控件设计显示学生信息查询进度程序	229
工作任务 23 用 ListView 控件设计班级相册程序	232
本章小结	234
习题 7	234
实验 7	235
第 8 章 图形绘制 GDI+简介	236
8.1 图形绘制概述	236
8.1.1 System.Drawing 命名空间	236
8.1.2 Graphics 类	236
8.1.3 GDI+坐标系	237
8.2 利用画笔绘制基本图形	237
8.3 画刷与区域填充	238
8.4 绘制较复杂的图形	240
8.5 图形变换	240
工作任务 24 系部班级统计图形绘制	242
本章小结	244
习题 8	245
实验 8	245
第 9 章 C#网络通信编程	246
9.1 网络通信编程概述	246
9.2 套接字 (Socket) 编程	247
9.3 线程类 Thread	251
工作任务 25 简单聊天通信程序设计	253
本章小结	260
习题 9	260
第 10 章 应用程序部署	261

10.1 部署概述	261
10.2 创建和部署基于 Windows 的应用程序	263
10.2.1 创建一个基于 Windows 的应用程序	263
10.2.2 创建部署项目	263
10.2.3 将基于 Windows 的应用程序添加到安装程序中	264
10.2.4 部署应用程序（基本安装程序）	265
10.3 应用程序可选部署功能	266
10.3.1 为基于 Windows 的应用程序创建快捷方式	266
10.3.2 创建文件关联	267
10.3.3 添加注册表项	268
10.3.4 添加自定义安装对话框	268
10.3.5 安装示例文件	269
10.3.6 添加启动条件	270
10.3.7 设置系统必备组件	270
10.3.8 部署应用程序到其他计算机	271
工作任务 26 学生档案管理系统安装程序	271
本章小结	272
习题 10	273
实验 9	273
参考文献	274

第1章 Windows 应用程序开发入门

C#是一种面向对象的、运行于.NET Framework 之上的高级程序设计语言,具有许多优良特性和广泛的应用前景。本章简要介绍 C#的基础知识,内容包括 .NET Framework 框架、公共语言运行库、C#应用程序类型和 Windows 应用程序开发等。通过本章的学习,读者可以了解 C#与.NET Framework、Visual Studio 2010 集成开发环境的关系,掌握 Visual Studio 2010 集成开发环境下基于 C#的 Windows 应用程序的开发步骤。

理论知识

1.1 C#概述

1.1.1 C#与.NET 框架的关系

C#(读作“C sharp”)是微软公司推出的一种以 C/C++为基础的新的开发语言。作为一种新的程序设计语言,C#的特点主要体现在以下两个方面。

1) 它是专门为配合 Microsoft 的.NET Framework 使用而设计开发的。.NET Framework 为使用 C#语言设计和开发桌面和网络应用程序提供了一个功能强大的平台。

2) 它是一种基于现代面向对象设计方法的语言。C#语言的开发和设计是 Microsoft 在近 20 年众多面向对象语言应用经验基础之上完成的,它吸收了其他语言的优点,使应用程序的开发变得更加简单和高效。

就其本身而言,C#只是一种程序设计语言,尽管它的应用是基于面向.NET 环境的代码之上,但它本身并不是.NET 框架的一部分。因此,.NET 支持的一些特性,C#并不完全支持,而.NET 也不支持 C#语言支持的一些特性,如运算符重载。由于使用 C#设计和开发的应用程序需要在.NET Framework 之上运行,所以对于 C#语言而言,应用程序的实现依赖于.NET。鉴于这种依赖关系,在开始介绍 C#程序设计语言之前,有必要先对.NET Framework 进行简单的了解。

1.1.2 .NET Framework 框架概述

1. 什么是.NET

.NET 是 Microsoft XML Web Services 平台。XML Web Services 允许应用程序通过 Internet 进行通信和数据共享,而不管所采用的是何种操作系统、设备或编程语言。.NET 平台可以创建 XML Web Services,并将这些服务集成在一起。它大致上可分为几种主要语言,如 Visual Basic .NET、Visual C#、Visual J#、Visual C++ .NET 等,在 Visual Studio 2010 平台中又增加了新的语言 F#。无论使用的是哪一种语言,在.NET 这个平台上都将编译成微软

中间语言 (Microsoft Intermediate Language, MSIL) 以达到无缝集成的目的。

Windows 操作系统只需要安装 Microsoft .NET Framework 即可运行 .NET 程序。Windows Server 2003 是内建 .NET 支持的第一个操作系统。

在 .NET 开发平台下, 所有语言 (C#、VB.NET、J#、[Managed C++]、F#) 都会被编译为 MSIL, 再由公共语言运行库 (Common Language Runtime, CLR) 负责运行。CLR 是微软公司开发平台 .NET Framework 运行的基础, 提供了 .NET 程序运行的底层环境。

2. .NET Framework

.NET Framework 是支持生成和运行下一代应用程序和 Web 服务的内部 Windows 组件, 提供了托管执行环境、简化的开发和部署以及与各种编程语言的集成, 旨在实现下列目标。

1) 提供一个一致的面向对象的编程环境, 而无论对象代码是在本地存储和执行, 还是在本地执行但在 Internet 上发布, 或者是在远程执行的。

2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。

3) 提供一个可提高代码 (包括由未知的或不完全受信任的第三方创建的代码) 执行安全性的代码执行环境。

4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。

5) 使开发人员的经验在面对类型大不相同的应用程序 (如基于 Windows 的应用程序和基于 Web 的应用程序) 时保持一致。

6) 按照工业标准生成所有通信, 以确保基于 .NET Framework 的代码可与任何其他代码集成。

.NET Framework 具有两个主要组件——公共语言运行库和 .NET Framework 类库 (包括 ADO.NET、ASP.NET、Windows 窗体和 Windows Presentation Foundation)。

公共语言运行库是 .NET Framework 的基础。将运行库看作一个在执行时管理代码的代理, 它提供内存管理、线程管理和远程处理等核心服务, 并且还强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码检查。事实上, 代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码, 而不是以运行库为目标的代码称为非托管代码。

.NET Framework 的另一个主要组件是类库, 它是一个综合性的面向对象的可重用类型集合, 可以使用其开发多种应用程序, 这些应用程序包括传统的命令行或图形用户界面 (GUI) 应用程序, 也包括基于 ASP.NET 所提供的最新创新的应用程序 (如 Web 窗体和 XML Web Services)。

.NET Framework 的基本结构如图 1-1 所示。

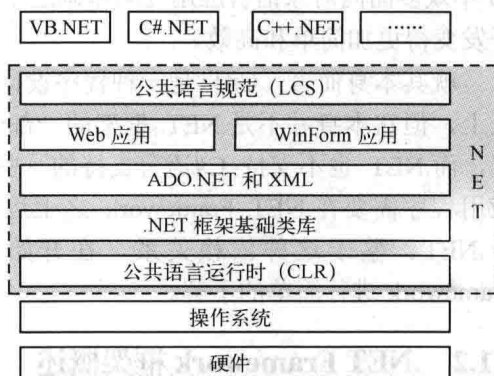


图 1-1 .NET 框架基本结构图

1.1.3 公共语言运行库

.NET Framework 的核心是运行库的执行环境, 称为公共语言运行库 (CLR) 或 .NET 运行库。通常将在 CLR 的控制下运行的代码称为托管代码 (Managed Code)。

但是, 在 CLR 执行开发的源代码之前, 需要编译它们 (C#或其他语言)。在 .NET 中编

译分为两个阶段。

1) 把源代码编译为微软中间语言 (MSIL)。

2) CLR 把 MSIL 编译为平台专用的代码。

这个两阶段的编译过程非常重要,正是将代码编译为中间语言使得.NET 具有了许多优点。

微软中间语言与 Java 字节代码共享同一种理念:它们都是一种低级语言,语法很简单(使用数字代码,而不是文本代码),可以快速地转换为内部机器码。对于代码来说,这种精心设计的通用语法有很重要的优点。

(1) 平台无关性

首先,这意味着包含字节代码指令的同一文件可以放在任一平台中,编译过程的最后阶段可以很容易地完成,这样代码就可以运行在特定的平台上。换言之,编译为中间语言就可以获得.NET 平台无关性,这与编译为 Java 字节代码就会得到 Java 平台无关性是一样的。

(2) 提高性能

前面把 MSIL 和 Java 字节代码做了比较,实际上,MSIL 比 Java 字节代码的作用还要大。MSIL 总是即时编译的(称为 JIT 编译),而 Java 字节代码常常是解释性的,其缺点是在运行应用程序时,把 Java 字节代码转换为内部可执行代码的过程会导致性能的损失。

(3) 语言的互操作性

使用 MSIL 不仅支持平台无关性,还支持语言的互操作性。简言之,就是能将任何一种语言编译为中间代码,编译好的代码可以与从其他语言编译过来的代码进行交互操作,如 Visual Basic 2010、Visual C++ 2010、Visual J# 2010、脚本语言、COM 和 COM+。

1.2 C#应用程序类型

C#程序设计语言可以快速、方便地设计和开发出多种类型的应用程序。

1. Windows 控制台应用程序

C#可以用于创建控制台应用程序。控制台应用程序是指仅使用文本,运行在 DOS 窗口中的应用程序。在进行单元测试、创建 UNIX/Linux daemon 进程时,就要使用控制台应用程序。

2. ASP.NET 应用程序

ASP 是用于创建带有动态内容的 Web 页面的一种 Microsoft 技术。ASP 页面基本是一个嵌有服务器端 Visual Basic Script 或 Java Script 代码块的 HTML 文件。当客户浏览器请求一个 ASP 页面时,Web 服务器就会发送页面的 HTML 部分,并处理服务器端脚本。这些脚本通常会查询数据库的数据,在 HTML 中标记数据。ASP 是客户建立基于浏览器的应用程序的一种便利方式。

ASP.NET 是 ASP 的修订版本,它解决了 ASP 的许多问题。但 ASP.NET 页面并没有替代 ASP,而是可以与原来的 ASP 应用程序在同一个服务器上同时并存,可以用 C#编写 ASP.NET。

3. Windows 窗体应用程序

C#和.NET 非常适合于 Web 开发,它们还为所谓的“胖客户端”应用程序提供了极好的

支持, 这种“胖客户端”应用程序必须安装在处理大多数操作的最终用户的机器上, 这种支持来源于 Windows 窗体。

要设计一个图形化的窗口界面, 只需要把控件从工具箱拖放到 Windows 窗体上即可。要确定窗口的行为, 应为该窗体的控件编写事件处理例程。Windows Form 项目编译为.exe 文件, 该文件必须与 .NET 运行库一起安装在最终用户的计算机上。与其他 .NET 项目类型一样, Visual Basic 2010 和 C# 都支持 Windows Form 项目。

4. Windows 控件

Web 窗体和 Windows 窗体的开发方式一样, 但应为它们添加不同类型的控件。Web 窗体使用 Web 服务器控件, Windows 窗体使用 Windows 控件。

Windows 控件比较类似于 ActiveX 控件。在执行 Windows 控件后, 它会编译为必须安装到客户机器上的 DLL。实际上, .NET SDK 提供了一个实用程序, 为 ActiveX 控件创建包装器, 以便把它们放在 Windows 窗体上。与 Web 控件一样, Windows 控件的创建需要派生于特定的类 System.Windows.Forms.Control。C# 支持创建自定义控件。

5. Windows 服务

Windows 服务 (最初称为 NT 服务) 是一个在 Windows NT/2000/XP/2003 (但没有 Windows 9x) 后台运行的程序。当希望程序连续运行, 响应事件, 但没有用户的明确启动操作时, 就应使用 Windows 服务。例如 Web 服务器上的 World Wide Web 服务, 它们监听来自客户的 Web 请求。

用 C# 编写 Windows 服务是非常简单的。System.ServiceProcess 命名空间中的 .NET Framework 基类可以处理许多与 Windows 服务相关的样本任务。另外, Visual Studio 2010 允许创建 C# Windows Service 项目, 为基本 Windows 服务编写 C# 源代码。

1.3 可视化程序设计

1.3.1 面向对象的程序设计

【例 1-1】 用 C++ 定义描述矩形 (按钮或窗体) 的类 Rectangle (事先设置好按钮的位置与大小)。

1) 矩形可用左上角坐标 (Left, Top) 与高、宽 (Height, Width) 来描述, 因此, 描述矩形类的私有数据成员为 Left、Top、Height、Width。

2) 编写初始化数据成员的构造函数。

3) 编写计算矩形面积的函数 Area()。

4) 主函数中定义矩形对象 r, 初值为 (1500, 1500, 600, 1500)。

调用 Area() 函数, 计算矩形面积并输出显示。

```
# include <iostream.h>
```

```
Class Rectangle
```

```
{
```

```
private:
```

```
//定义矩形数据成员
```

```

    int Left, Top, Height, Width;
public:
    //定义带参构造函数
    Rectangle(int L, int T, int H, int W)
    {
        Left=L; Top=T; Height=H; Width=W;
    }
    //定义计算矩形面积成员函数
    int Area(void)
    {
        return Height * Width;
    }
};

void main (void)
{
    Rectangle r1(1500,1500,600,1500);
    cout<<"矩形 r1 的面积="<<r1.Area()<< endl;
}

```

下面列出在面向对象程序设计中常用的一些概念。

1) 类 (Class): 由描述事物的数据及处理数据的函数组成的导出数据类型, 如按钮、窗体等。

2) 对象 (Object): 用类定义的变量称为对象, 如在主函数中用 **Rectangle** 定义的矩形对象 **r1**。

3) 属性 (Property): 将描述对象特性的数据成员称为属性, 如矩形左上角坐标 (**Left**, **Top**)、高宽 (**Height**, **Width**) 均为对象 **r1** 的属性。属性值可以在构造函数中进行赋值, 如 **Left=600**, **Top=600**, **Height=600**, **Width=1500**。

对于公有数据成员, 属性可在主函数中直接修改, 对私有属性则只能通过接口函数进行设置和修改。

4) 方法 (Method): 将处理数据的成员函数称为方法, 如成员函数 **Area()** 为计算矩形面积的方法。

调用方式为: 对象.方法[参数], 如 **r1.Arear ()**。

1.3.2 可视化程序设计概述

可视化程序设计是指在窗体中使用控件设计程序界面, 编写控件事件驱动程序的设计方法。可视化程序设计的界面设计过程中基本不用编写程序代码, 因为这些工作在 C# 的集成开发环境 IDE 中已经帮开发者完成了。

用 Visual Studio 2010 开发窗体应用程序包括两部分工作: 一是设计图形用户界面; 二是编写程序代码。Visual Studio 2010 提供了一个“画板”(窗体), 也就是用户界面, 还提供一个“工具箱”, 在“工具箱”中放了许多被称为“控件”的工具, 例如有制作按钮的工具、有制作文本框的工具、有显示图形数据的工具等。可以从工具箱中取出所需工具, 放到“画板”中适当的位置上, 这样就形成了“用户界面”, 也就是说, 屏幕上的用户界面是用 Visual Studio 2010 提供的可视化设计工具——“控件”直接“画”出来的, 而不是用程序

“写”出来的。当然最直观且最复杂的界面设计也是由程序编写出来的，只不过这些编程工作不用读者来做，而是由开发平台替程序员来完成。

1.3.3 事件驱动的程序设计

1. 对象事件

对象对用户的操作进行响应的动作称为事件。如当鼠标单击按钮对象时，在窗体的标签控件上显示“您好！”，用鼠标单击按钮触发了事件，而显示“您好！”的这一动作是对事件的响应。

对象事件是 C# 为每个对象设置的响应过程，如按钮对象就包括了 Click、KeyDown、KeyUp、KeyPress 等事件。当程序设计者使用事件时，IDE 已经为事件准备好了事件函数的框架，设计者只需完成具体的实现代码即可。

2. 事件驱动程序设计

按设计要求编写控件事件驱动程序，执行程序时，触发控件执行事件驱动程序，完成规定任务的程序设计方法称为事件驱动程序设计。

1.4 Visual Studio 2010 介绍

Visual Studio 2010 是微软推出的应用于 .NET 4.0 开发的首选工具，其功能强大而且方便易用，提供了在设计、开发、调试和部署应用程序时所需的工具。Visual C# 集成开发环境 (IDE) 是 Visual Studio IDE 中的一种。打开 Visual Studio 2010 选择 C# 即可进入 Visual C# 集成开发环境，里面有些工具是与其他 Visual Studio 语言共享的，还有一些工具（如 C# 编译器）是 Visual C# 特有的。

1.4.1 安装 Visual Studio 2010

1. 安装要求

支持的操作系统：Windows Server 2003、Windows 7、Windows Vista、Windows XP。

支持的数据库：Access 2000 或 SQL Server 2005 及以上版本。

处理器：1.6GHz Pentium 处理器或与之相当的处理器（最低）。

RAM：1024MB（最低）；如果在虚拟机上运行，则为 1.5 GB。

硬盘：5400 r/min 硬盘。若不安装 MSDN，则安装驱动器上需要有 3GB 空间。

显示器：1024×768 像素或更高的显示分辨率运行的支持 DirectX 9 的视频卡。

DVD-ROM 驱动器。

2. 安装步骤

Visual Studio 2010 的安装步骤非常简单，放入 Visual Studio.NET 2010 安装光盘后按提示进行安装即可，具体步骤如下。

1) 打开安装程序所在的目录，运行安装程序 Visual Studio 2010 \autorun.exe，弹出 Visual Studio 2010 安装界面，选择 [安装 Microsoft Visual Studio 2010]，如图 1-2 所示；

2) 安装程序会自动加载安装组件；

3) 接受用户许可协议；

4) 选择 [完全安装] 或 [自定义]，并设置好安装目录；

- 5) 安装程序开始安装各种组件，如图 1-3 所示；
- 6) 安装组件时会有一个重启过程。当所有组件安装完后，出现安装成功的界面。



图 1-2 Visual Studio 2010 的安装界面

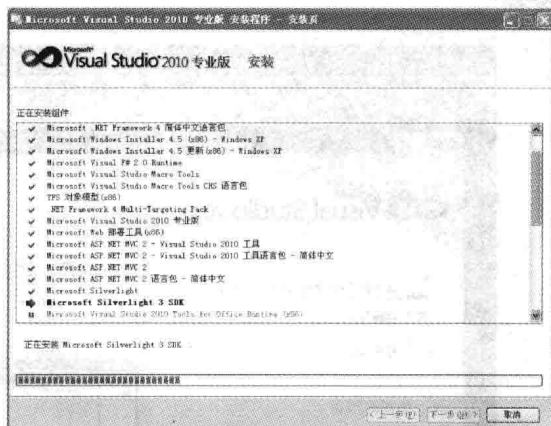


图 1-3 安装 Visual Studio 2010 的组件

1.4.2 Visual Studio 2010 集成开发环境介绍

选择“开始”→“所有程序”→“Microsoft Visual Studio 2010”命令，会显示如图 1-4 所示的菜单。

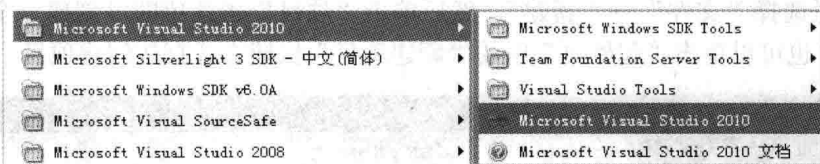


图 1-4 Microsoft Visual Studio 2010 菜单

其中：

“Microsoft Windows SDK（简称 Windows SDK 或 SDK）Tools”是由微软公司出品的一个软件开发包，向在微软的 Windows 操作系统和 .NET 框架上开发软件和网站的程序员提供头文件、库文件、示例代码、开发文档和开发工具。

“Team Foundation Server（简称 TFS）Tools”提供了一个强健的系统，不仅提供了源码管理，而且包括了项目跟踪和开发支持，对团队开发有着极大的帮助。

“Visual Studio Tools”是用于 Visual Studio .NET 开发的一些辅助工具，有一些工具是 Visual Studio 系列开发工具一直都有的，如命令提示、远程调试器等。

“Microsoft Visual Studio 2010”是 Microsoft Visual Studio 执行的快捷方式，单击可以进入 Microsoft Visual Studio 2010 开发环境。

“Microsoft Visual Studio 2010 文档”是 Microsoft Visual Studio 2010 的帮助文档，提供应用程序开发帮助信息。

单击“Microsoft Visual Studio 2010”将进入 Microsoft Visual Studio 开发环境，然后单击“启动 Visual Studio”按钮即可进入开发界面。