



全国高职高专教育“十一五”规划教材

C# 技术基础

杨学全 主编 刘永辉 张红强 副主编



高等教育出版社
Higher Education Press

全国高职高专教育“十一五”规划教材

C# 技术基础

杨学全 主编

刘永辉 张红强 副主编

高等教育出版社

内容简介

本书是全国高职高专教育“十一五”规划教材。

本书是以技术能力培养为主线,采用任务驱动模式编写的案例型教材。本书在内容安排上遵循“循序渐进”与“难点分解”的原则,通过典型案例,从基本概念和实际应用出发,由浅入深、循序渐进地讲述 C# 2.0 语言基础知识、面向对象的程序设计、可视化程序设计和数据库应用系统开发等内容。

本书所附光盘中提供了书中全部例题的源代码。

本书可作为应用性、技能型人才培养的各类教育“C#语言程序设计”课程的教学用书,也可供各类培训、计算机从业人员和程序设计爱好者参考使用。

图书在版编目(CIP)数据

C#技术基础 / 杨学全主编. —北京: 高等教育出版社, 2008.1

ISBN 978-7-04-022475-7

I. C… II. 杨… III. C 语言-程序设计-高等学校: 技术学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 182236 号

策划编辑 刘怀恩 责任编辑 萧 潇 封面设计 张志奇 责任绘图 黄建英
版式设计 张 岚 责任校对 杨雪莲 责任印制 韩 刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京汇林印务有限公司

开 本 787×1092 1/16
印 张 22.25
字 数 540 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 [http:// www.hep.edu.cn](http://www.hep.edu.cn)
[http:// www.hep.com.cn](http://www.hep.com.cn)
网上订购 [http:// www.landaco.com](http://www.landaco.com)
[http:// www.landaco.com.cn](http://www.landaco.com.cn)
畅想教育 [http:// www.widedu.com](http://www.widedu.com)

版 次 2008 年 1 月第 1 版
印 次 2008 年 1 月第 1 次印刷
定 价 31.80 元 (含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 22475-00

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010)58581897/58581896/58581879

传 真：(010)82086060

E-mail: dd@hep.com.cn

通信地址：北京市西城区德外大街 4 号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)58581118

前 言

课程建设与改革是提高教育、教学质量的核心,也是教学改革的重点和难点,更是培养生产、建设、管理、服务等一线高素质技能型专门人才的关键;经过多年的探索和实践,高职高专院校逐步构建起以技术应用能力培养为主线,以就业为导向的计算机应用类专业实践教学课程体系。实践课程体系中的每门课程都是根据计算机信息技术领域和中小企业信息化职业岗位(群)的任职要求,参照国家信息技术人才的职业资格标准开发和建设的,课程教学内容体现了实用性、综合性和职业性。

《C#技术基础》是实践课程体系中程序设计课程的教材。本书是一本以职业技术能力培养为主线,采用任务驱动模式编写的案例型教材。教材融“教、学、做”为一体,注重基本知识与基本技术讲解(教),给出具有实用价值的案例供学生模仿(学),通过课程设计强化学生能力的培养(做)。本教材适用于计算机应用类专业或非计算机专业的程序设计课程教学,是软件工程、信息系统开发、开发工具等课程的前驱课程。

本书共分为11章,从基本概念和实际应用出发,由浅入深、循序渐进地讲述C# 2.0语言基础知识、面向对象的程序设计、可视化程序设计和数据库应用系统开发等内容;通过典型示例,清晰地讲解语言基础知识、面向对象程序设计和可视化程序设计的基本技术,给出了界面设计和文本编辑器设计案例,供学生模仿学习;将“学生选课管理信息系统”开发案例作为课程设计内容,培养学生的实际软件编程能力。根据职业技能培养的要求,结合案例,每章给出了上机实训操作内容和要求,以便于学生更好地学习和掌握C#程序设计的基本知识与软件编程的基本技能。通过本课程的学习,学生能够掌握C#语言的基本知识与技术,理解并树立面向对象的程序设计思想,培养运用Visual C# 2005 .NET开发工具进行编程的能力,提高软件开发的水平。

建议教学学时数为72学时,也可根据具体情况选择讲授内容。

本书讲解力求准确、简练,强调知识的层次性和技能培养的渐进性,精心设计例题和习题,课程设计题目具有实用性,强调学生程序设计基本技能的培养。

本书由杨学全主编,其中第1章由张少轩编写,第2章由刘甜编写,第3、4、5章由杨学全编写,第6章由张红强编写,第7章由李淑娣编写,第8章由杨健编写,第9~11章由刘永辉编写,最后由杨学全统稿,滕桂法教授审稿。

衷心感谢河北农业大学的滕桂法教授、保定职业技术学院的陈志强教授和刘海军教授,他们的辛勤工作使我们受益匪浅。衷心感谢北京万达信息技术有限公司的白皓先生在百忙之中对全书进行了认真审阅,并提出了宝贵的意见。



由于作者水平有限，加之时间仓促，书中难免有错漏之处，敬请读者批评指正，我们将不胜感激。

编 者

2007年11月

目 录

第 1 章 Visual C# 2005 .NET 概述.....1	
1.1 .NET 简介.....1	
1.1.1 什么是.NET.....1	
1.1.2 什么是.NET 框架.....2	
1.1.3 为什么要使用.NET.....5	
1.2 Visual Studio 2005 .NET 简介.....5	
1.2.1 Visual Studio 2005 .NET 集成 开发环境概述.....5	
1.2.2 安装 Visual Studio 2005 .NET 集成开发环境.....5	
1.3 Visual C# 2005 .NET 简介.....10	
1.3.1 Visual C# 2005 语言.....10	
1.3.2 启动 Visual Studio 2005 .NET 集成开发环境.....11	
1.3.3 创建第一个基于 C# 的 Windows 应用程序项目.....12	
1.3.4 Visual C# 2005.NET 集成 开发环境.....12	
1.4 编写一个简单的 Visual C# 2005 .NET 程序.....19	
1.4.1 创建控制台应用程序.....19	
1.4.2 创建 Windows 窗体应用程序.....22	
1.4.3 打开和更名保存程序.....23	
1.5 C#应用程序分析.....24	
1.5.1 应用程序项目文件组成.....24	
1.5.2 C#程序结构分析.....25	
1.5.3 程序的输入/输出.....28	
本章小结.....31	
实训 1 创建一个简单的应用程序.....31	
习题 1.....32	
第 2 章 C#语言基础.....34	
2.1 C#的基本语法.....34	
2.1.1 C#编程规则.....34	
2.1.2 数据类型.....36	
2.1.3 常量与变量.....38	
2.1.4 类型转换.....41	
2.2 运算符和表达式.....45	
2.2.1 算术运算符及其表达式.....45	
2.2.2 赋值运算符及其表达式.....47	
2.2.3 关系运算符及其表达式.....48	
2.2.4 逻辑运算符及其表达式.....49	
2.2.5 字符串连接运算符及其 表达式.....50	
2.2.6 其他运算符.....50	
2.2.7 运算符的优先级.....51	
2.3 流程控制语句.....52	
2.3.1 条件语句.....52	
2.3.2 循环语句.....57	
2.3.3 循环语句的嵌套.....61	
本章小结.....63	
实训 2 C#语言基础.....63	
习题 2.....64	
第 3 章 数组、枚举与结构.....65	
3.1 数组.....65	
3.1.1 一维数组.....65	
3.1.2 二维数组.....67	
3.2 数组与 Array 类.....69	
3.2.1 Array 类的 Length 属性.....69	
3.2.2 Array 类的 Sort()方法.....70	
3.2.3 Array 类的 BinarySearch() 方法.....70	
3.3 枚举.....71	
3.3.1 声明枚举.....71	
3.3.2 引用枚举.....72	
3.4 结构.....73	
3.4.1 结构的声明.....74	

3.4.2 结构成员的访问	74	第 6 章 窗体与常用控件	146
3.4.3 结构应用示例	76	6.1 Windows 窗体	146
本章小结	80	6.1.1 创建窗体	147
实训 3 数组、枚举与结构	80	6.1.2 窗体的属性、事件、方法	147
习题 3	82	6.1.3 窗体中控件的操作	151
第 4 章 C#面向对象程序设计基础	84	6.1.4 窗体的基本操作	153
4.1 面向对象程序设计概述	84	6.1.5 窗体操作实例	154
4.1.1 面向对象程序设计的引入	84	6.2 Label、TextBox 与 Button	
4.1.2 面向对象的基本概念	85	控件	155
4.1.3 面向对象的基本特征	85	6.2.1 Label 控件	155
4.2 C#语言中的类和对象	87	6.2.2 TextBox 控件	156
4.2.1 类的定义	87	6.2.3 Button 控件	157
4.2.2 对象	90	6.2.4 Label、Textbox 和 Button	
4.2.3 类的成员及作用域	94	控件应用实例	158
4.3 类的继承	101	6.3 RadioButton、CheckBox 与	
4.3.1 创建子类	101	GroupBox 控件	159
4.3.2 base 与 this 的使用	103	6.3.1 RadioButton 控件	160
4.3.3 访问控制的深入探讨	108	6.3.2 GroupBox 控件	160
本章小结	112	6.3.3 CheckBox 控件	160
实训 4 面向对象的程序设计	112	6.3.4 RadioButton、CheckBox 和	
习题 4	114	GroupBox 控件示例	160
第 5 章 面向对象高级编程	116	6.4 列表框控件	164
5.1 接口、委托与事件	116	6.4.1 ListBox 控件	164
5.1.1 接口	117	6.4.2 CheckedListBox 控件	168
5.1.2 委托与事件	123	6.4.3 ComboBox 控件	172
5.2 类的多态	129	6.5 其他常用控件	174
5.2.1 方法重载	129	6.5.1 PictureBox 控件	174
5.2.2 方法重写	130	6.5.2 Timer 控件	175
5.2.3 抽象类与抽象方法	132	6.5.3 DateTimePicker 控件	177
5.3 运算符重载和泛型	134	本章小结	178
5.3.1 运算符重载	134	实训 6 窗体及常用控件的使用	178
5.3.2 泛型	138	习题 6	179
5.4 命名空间	142	第 7 章 窗体界面高级编程	181
5.4.1 命名空间的概念	142	7.1 窗体界面设计概述	181
5.4.2 使用命名空间	143	7.2 菜单设计	182
本章小结	144	7.2.1 主菜单设计	182
实训 5 面向对象高级编程	144	7.2.2 快捷菜单设计	187
习题 5	145	7.3 工具栏与状态栏设计	189
		7.3.1 工具栏设计	190

7.3.2 状态栏设计	195	习题 8	254
7.3.3 创建可停靠的菜单和 工具栏	199	第 9 章 数据库编程基础	255
7.4 MDI 窗体设计	205	9.1 数据库基础知识	255
7.4.1 创建 MDI 应用程序	206	9.1.1 数据库系统	255
7.4.2 创建 MDI 应用程序实例	207	9.1.2 关系数据库	256
本章小结	210	9.1.3 开发数据库的步骤	256
实训 7 窗体界面设计	211	9.1.4 常用 SQL 语句	257
习题 7	211	9.1.5 创建本章所用示例数据库	259
第 8 章 文件与对话框	213	9.2 ADO.NET 基础	261
8.1 文件管理控件	213	9.2.1 ADO.NET 简介	261
8.1.1 文件管理控件简介	213	9.2.2 Connection 对象	262
8.1.2 DriveListBox 控件	214	9.2.3 Command 对象	263
8.1.3 DirListBox 控件	215	9.2.4 DataReader 对象	268
8.1.4 FileListBox 控件	216	9.2.5 DataSet 对象	270
8.2 文件夹操作	218	9.2.6 DataTable、DataRow 和 DataColumn 对象	271
8.2.1 创建文件夹	219	9.2.7 DataAdapter 对象	271
8.2.2 删除文件夹	221	9.3 数据绑定	272
8.2.3 查看和设置文件夹信息	222	9.3.1 简单数据绑定	272
8.3 文件操作	224	9.3.2 使用 DataGridView 控件	273
8.3.1 文件的概念	224	9.4 水晶报表	275
8.3.2 文件的管理	225	9.4.1 建立报表	276
8.3.3 文件的读/写操作	226	9.4.2 浏览和打印报表	279
8.4 通用对话框设计	232	本章小结	280
8.4.1 通用对话框概述	232	实训 9 访问数据库	280
8.4.2 OpenFileDialog 控件	233	习题 9	282
8.4.3 SaveFileDialog 控件	236	第 10 章 Web 服务、程序调试与 异常处理	284
8.4.4 FontDialog 控件	240	10.1 Web 服务基础	284
8.4.5 ColorDialog 控件	241	10.1.1 Web 服务关键协议介绍	285
8.5 打印、设置类对话框	243	10.1.2 创建简单的 Web 服务	285
8.5.1 PrintDocument 控件	243	10.1.3 访问 Web 服务	288
8.5.2 PageSetupDialog 控件	246	10.2 Web 服务应用实例	290
8.5.3 PrintDialog 控件	247	10.2.1 Web 服务和 DataSet	291
8.5.4 PrintPreviewDialog 控件	248	10.2.2 Web 服务和二进制文件	291
8.6 自定义对话框	249	10.2.3 Web 服务和应用程序 状态	292
8.6.1 创建自定义对话框	249	10.2.4 Web 服务和 Session	293
8.6.2 使用自定义对话框	250	10.3 程序错误类型与程序调试	294
本章小结	252		
实训 8 文件与对话框应用	253		

10.3.1 错误类型	294	11.1.1 系统需求分析	308
10.3.2 程序调试	296	11.1.2 系统总体设计	308
10.4 异常处理	302	11.1.3 系统数据库设计	309
10.4.1 异常类	302	11.2 学生选课管理信息系统的	
10.4.2 异常处理方法	303	实现	315
本章小结	306	11.2.1 运行效果	315
实训 10 Web 服务与错误调试	306	11.2.2 代码实现	317
习题 10	307	11.2.3 发布应用程序	342
第 11 章 学生选课管理信息系统		本章小结	344
开发案例	308	课程设计 学生选课系统功能扩充	345
11.1 学生选课管理信息系统的		参考文献	346
分析与设计	308		

1

第1章 Visual C# 2005 .NET 概述



学习目标

学完本章后，学生应能够：

- 描述.NET 的概念和组成以及.NET 框架的结构和各组成部分的作用。
- 描述 Visual Studio 2005.NET 开发环境及其在软件开发中的作用。
- 使用 Visual Studio 2005.NET 安装盘安装集成开发环境。
- 运用 Visual Studio 2005.NET 工具，创建简单控制台应用程序和 Window 窗体应用程序。
- 叙述应用程序项目文件的作用，解释 C#应用程序的结构。



预备知识

学习本章前，学生应具备：

计算机应用基础知识，一定的 Windows 应用程序操作技能，对 Windows 应用程序窗口、菜单、工具栏等一些基本概念的认知。

Visual C# 2005 .NET 是 Microsoft 公司推出的 Visual Studio 2005 .NET 集成开发环境 (Integrated Development Environment, IDE) 的一部分，是一种完全面向对象的专门为.NET 设计的程序语言。本章主要介绍.NET 和.NET 框架 (.NET Framework) 的有关概念及 Visual C# 2005 .NET 集成开发环境的使用。

1.1 .NET 简介

随着 Internet 的飞速发展，人类已进入网络计算和数字化时代。如何适应时代发展的要求，创新网络模式下的软件开发理念，更新软件开发技术就成为业界的迫切任务。Microsoft 公司为了适应 Internet 发展和新一代网络计算的需要，于 2000 年 6 月 22 日正式对外宣布了“.NET”战略。

1.1.1 什么是.NET

Microsoft 公司总裁兼首席执行官史蒂夫·鲍尔默说：“.NET 代表了一个集合、一个环境、一个编程的基本结构，可以作为一个平台来支持下一代的互联网。.NET 也是一个用户环境，是

一组基本的用户服务，可以作用于客户端、服务器端和任何地方，具有很好的一致性，并有新的创意。因此，它不仅是一个用户体验，而且是开发人员体验的集合，这就是对.NET 概念的描述。”由此可以看出，Microsoft .NET 是以公共语言运行时（Common Language Runtime, CLR）为基础，以 Web 服务为核心技术，为信息、人、系统、智能设备提供无缝链接的一组软件产品（如 SmartClient、服务器、开发工具）、技术（如 Web 服务）或服务（.NET 服务，如.NET Passport）。

.NET 以公共语言运行时为基础，实现了跨平台和跨语言开发；不管所采用的是哪种操作系统、设备或编程语言，以 XML Web 服务为核心技术，应用程序就能够通过 Internet 进行通信和共享数据。.NET 由 5 个主要部分组成，如图 1-1 所示。

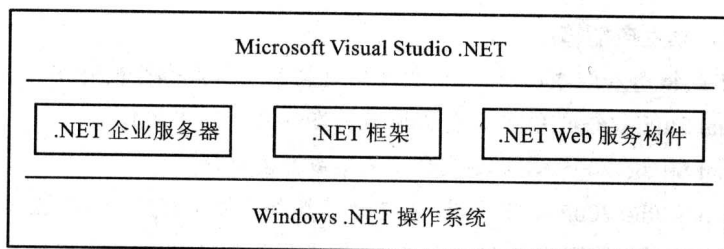


图 1-1 分布式系统平台

- Windows.NET 操作系统：主要包括在手机、微型计算机和服务器群集上运行的 Windows 操作系统（如 Windows 2003 Server）以及各种应用软件服务（如 IIS、Active Directory 等）。
- .NET 企业级服务器：主要包括 Commerce Server 2000、SQL Server 2005、Exchange Server 2000 等。
- .NET Web 服务构件：.NET 提供一系列高度分布、可编程的公用性网络服务，可以从任何支持 SOAP 的平台上访问.NET 服务构件。
- .NET 框架：是.NET 的核心部分，提供了建立和运行.NET 应用程序所需的编辑、编译等核心服务。
- Microsoft Visual Studio .NET：简称 MS VS .NET，是一套集成的开发环境。

1.1.2 什么是.NET 框架

.NET 框架是.NET 中最重要的部分，是一个开发平台，是一个多语言的组件开发和执行环境，支持多种编程语言（如 C#语言、VB.NET 语言等）。.NET 框架支持多种应用程序开发，如典型的 Windows 应用程序、控制台应用程序、Web 应用程序、Web 服务等各种类型的应用程序。使用.NET 框架可以满足企业快速开发应用程序的大部分需要，并且使得 Internet 上的各应用程序之间可以使用 Web 服务进行沟通。目前，.NET 框架的主流版本是 2.0（Visual Studio 2005 .NET），本书的内容就是基于 2.0 版本讲解的。

如图 1-2 所示，从层次上看，.NET 框架建立在操作系统之上，包含公共语言运行时、基础类库（BCL）、数据库访问组件（ADO.NET 和 XML）、基于 ASP.NET 的 Web 服务和 Web 窗体、

Windows 桌面应用程序编程组件（Windows 窗体）。

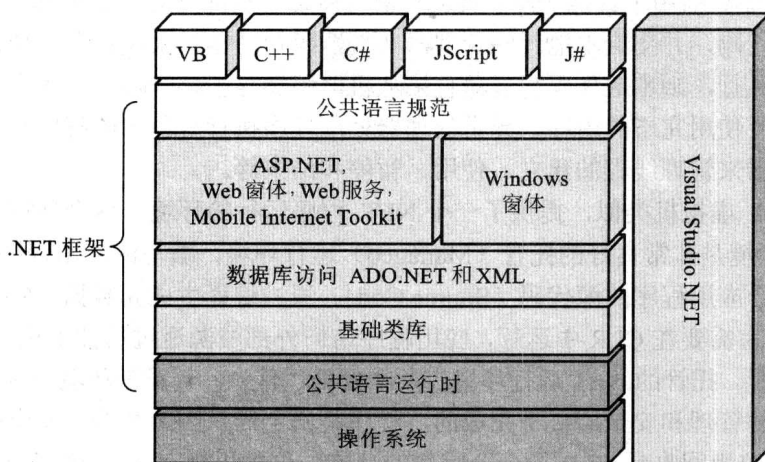


图 1-2 .NET 框架

1. 公共语言运行时

公共语言运行时（CLR）是一个软件引擎，用于加载应用程序、检查错误、进行安全许可认证、执行和清空内存。CLR 是 .NET 框架的核心部分，是虚拟运行层，其作用类似于 Java 虚拟机（JVM），它提供了一个 .NET 运行环境。总而言之，CLR 负责所有使用 .NET 库开发的应用程序的执行。

公共语言运行时由中间语言编译器（Intermediate Language Compiler）、程序集加载器（Assembly Loader）、类加载器（Class Loader）、实时编译器（Just-In-Time Compiler）、本地代码管理器（Native Code Manager）、垃圾回收（Garbage Collection）、异常管理器（Exception Manager）、线程支持（Thread Support）和调试引擎（Debug Engine）等部件组成。中间语言编译器负责将由 C# 或 .NET 平台下其他语言编写的程序源代码编译为扩展名为 EXE 或 DLL 的文件，这些文件是由中间语言（Microsoft Intermediate Language, MSIL）描述的。这些文件称为程序集，程序集是不能直接被 CPU 执行的可执行文件，这一点类似 Java 的字节码。程序集加载器负责加载程序集，并检查程序集是否有被编译的权限以及相关的安全性原则。类加载器负责加载程序集中引用的类库与程序集一并进入实时编译器。实时编译器负责编译由中间语言描述的程序集以及所引用的类库，使程序集变成可以在本地执行的机器码。程序管理器负责实时编译器编译程序集所产生的本地代码的缓存和执行管理，如果是第二次执行程序，就不会再对程序集进行实时编译，而是直接从本地代码缓存中执行。垃圾回收负责回收应用程序不再使用的某些内存，并将这些内存释放等待其他应用程序使用。异常管理器负责程序异常的统一管理并实现了跨语言的异常处理。这里需要补充的是，程序异常是指程序运行中出现的错误（如除零、磁盘写入时出错等）。线程支持负责线程的建立、使用、暂停和结束。调试引擎负责 .NET 平台下跨语言的程序调试。

用户所编写的应用程序在 CLR 环境下执行的简化过程是：程序源代码经过中间语言编译器编译成程序集后，首先程序集加载器将程序集、元数据和程序资源（如图标、文字等）加载并



检查其安全性,然后将程序集交由实时编译器(JIT)进行编译(由中间语言到本地机器语言的编译),如果该程序集引用了其他类库,类加载器将所需的类库一并加载编译。程序集编译后形成可执行的本地代码,并交给本地代码管理器(机器码管理器)来执行和管理,如果程序集的机器码是第二次执行,则本地代码管理器直接从机器码缓存中执行程序。程序执行过程中,垃圾回收器释放已经使用完毕的内存,异常处理器处理程序执行过程中的程序异常。多线程应用程序由线程管理器来管理线程的建立、使用、暂停和结束等。

CLR 与 Java 虚拟机类似,提供了一个.NET 程序运行的环境。这个环境有很多突出的特点:.NET 运行环境是非常良好的托管(Managed)运行环境,编写的应用程序都是在 CLR 这个容器中运行的,应用程序的源代码(Source Code)通过编译生成元数据(Meta)和中间语言(MSIL),中间语言需要在 CLR 中运行,应用程序访问外部资源透过的技术也都在 CLR 的管理之下,这就是托管。托管的.NET 运行环境有两个好处:首先,开发是环境一致的;其次,整个运行是在 CLR 的管理和安全监督下完成的,程序是安全的,因为某些不安全操作是不允许在 CLR 中运行的。垃圾回收是 CLR 的一个重要的功能,这种思想在 Java 语言中也有所体现。垃圾回收保证应用程序不再使用的内存会被自动地回收和释放,供其他程序使用,这减轻了程序员的负担,也保障了系统的稳定性。

2. 框架类库

在传统的开发环境中,每种语言都有自己的函数库,函数库中的函数都不尽相同,这样实现跨语言编程就比较困难。.NET 框架的类库在公共语言运行时之上,它是一个可供不同语言调用的、分层的、面向对象的 reusable 类的集合。这样 C# 语言和 Visual Basic (以下简称 VB) 语言都可以很容易用相同的方式使用框架类库中的类。在框架类库中提供了大量的基础类,如窗体控件、通信协议、输入/输出、数据访问等。这些类以命名空间(Namespace)的分级结构来划分功能,并使用点符号(.)来分隔各层次,使查找和使用类库非常容易,System 是这个基础类库的根。使用框架类库可以开发命令行应用程序(即控制台应用程序)、Windows 桌面应用程序、Web 应用程序、Web 服务和 Windows 服务等。

3. ADO.NET 和 XML

几乎所有的程序都要访问各种不同的数据,这些数据可能存储在简单的文本文件中,也可能存储在大型的关系数据库中。在.NET 中访问数据库的技术就是 ADO.NET。ADO.NET 提供了一组类,可以用来完成链接数据库、运行命令、返回记录集等任务。ADO.NET 提供了对非链接编程模型的支持。ADO.NET 也提供了对 XML 的支持,通过 XMLReader、XMLWriter、XMLNavigator 和 XMLDocument 对象来快速访问 XML 数据。

4. 基于 ASP.NET 的 Web 窗体和 Web 服务

ASP.NET 的主要目标是降低 Web 应用开发的门槛,它采用类似于 VB 的事件驱动模式的编程模型,在此模型中开发页面就像在传统的桌面应用程序开发中向窗体中添加控件一样简单。这让程序员可以用熟悉的桌面应用程序开发方式来实现 Web 应用的程序的开发。Web 窗体为 Web 应用程序开发提供了窗体化的应用程序开发工具,可以以“所见即所得”的方式完成页面的设计。Web 服务为构建分布式的应用程序提供了各种功能模型,用于显示其他应用程序和“智能”客户端的功能,使应用程序之间可以交换信息。Web 服务体现了一个超出传统应用程序的优点,它使用标准的互联网协议和数据格式,可以很容易地穿透组织边界,实现程序的

功能。

5. WinForms 用户界面

.NET 框架也可以支持传统的 Windows 应用程序的开发, WinForms 是用来创建标准的 Win32 应用程序的新方式, 它提供了创建桌面程序的窗体类、控件类等。同样它也利用了公共语言运行时, 减少了客户端应用程序开发工作量。

这里只是对 .NET 框架进行简要地讲解, 读者不必对这些内容深究, 随着后续内容的学习, 会逐渐对 .NET 框架有较深入的理解。

1.1.3 为什么要使用 .NET

Microsoft .NET 的策略是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行合理延伸。这样, 开发人员必将创建出摆脱设备硬件束缚的应用程序, 以便轻松实现互联网连接。.NET 的最终目的就是让用户在任何地方、任何时间, 利用任何设备都能访问所需的信息、文件和程序。用户不需要知道这些文件放在什么地方, 只需发出请求, 然后只管接收就可以了, 所有后台的复杂性是完全屏蔽起来的。由此可见, Microsoft .NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

1.2 Visual Studio 2005 .NET 简介

1.2.1 Visual Studio 2005 .NET 集成开发环境概述

早期的程序设计首先需要用字处理软件编辑源程序, 然后用编译程序进行编译, 再用连接程序进行连接, 最后用各种专门工具查看和管理编程需要的资源, 这对开发者来说非常不方便。为了解决这一问题, 出现了集成开发环境, 其主要目的是方便软件开发人员进行开发, 它具有界面友好的可视化环境, 整合的源代码编辑器、编译器、调试器和资源编辑器; 开发人员可利用插件模式, 整合各种开发工作。

Visual Studio 2005 .NET 是 Microsoft 推出的最新款开发工具软件。Visual Studio .NET 开发环境从最初的 Visual Studio 97 开始, 经历了 Visual Studio 6.0、Visual Studio 2002 .NET、Visual Studio 2003 .NET 到目前的 Visual Studio 2005 .NET。Visual Studio 2005 .NET 开发工具包含 Visual C# 2005 .NET、Visual Basic 2005 .NET、Visual C++ 2005 .NET 和 Visual J# 2005 .NET 等程序语言, 目前已经成为了重要的集成开发环境。进行 .NET 开发的工具并非只有 Visual Studio .NET, Windows 自带的记事本程序、Borland C# Builder 和 MonoDevelop 等都可以实现 .NET 下的应用程序开发。本书的讲解及书中的例题和开发案例均使用 Visual C# 2005 .NET 开发。

1.2.2 安装 Visual Studio 2005 .NET 集成开发环境

作为一个强大的开发工具, Visual Studio 2005 .NET 对系统的环境要求较高, 硬件方面推荐 Pentium 4 3.0 GHz 以上 CPU、512 MB 以上内存空间, 80 GB 以上硬盘空间, 显示器分辨率 1 024×768 像素以上。软件要求 Windows 2000、Windows XP 或更高版本的操作系统, Access 2000、SQL Server 2000 以上版本的后台数据库。

1. 安装 Visual Studio 2005 .NET

安装 Visual Studio 2005 .NET 的步骤如下。

① 将 Visual Studio 2005 .NET 安装光盘放入光驱中，就会出现如图 1-3 所示的安装界面。



图 1-3 Visual Studio 2005 安装界面 (1)

② 单击“安装 Visual Studio 2005”链接，进入如图 1-4 所示的安装界面，此处用户可以根据自己的意愿选择是否参加 Microsoft 的帮助改进安装活动。

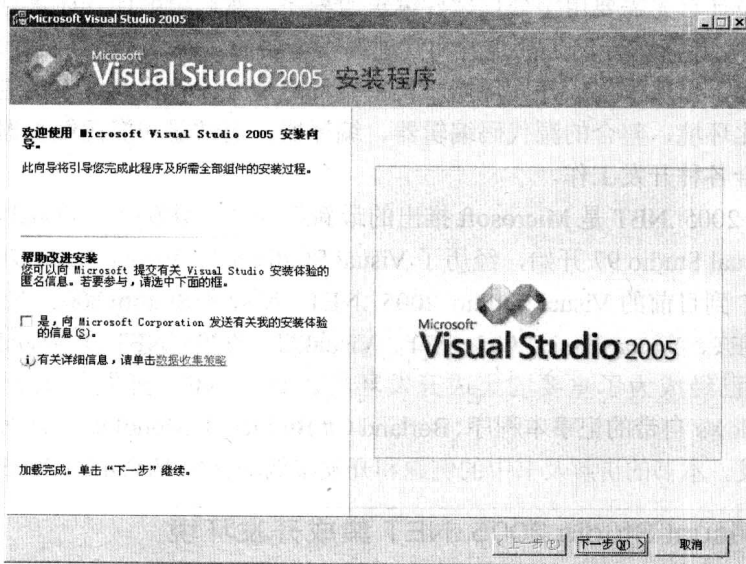


图 1-4 Visual Studio 2005 安装界面 (2)

③ 单击“下一步”按钮，出现如图 1-5 所示的安装窗口。在此窗口用户需要同意最终用户许可协议，并在“产品密钥”文本框中输入密钥，才能够继续安装。

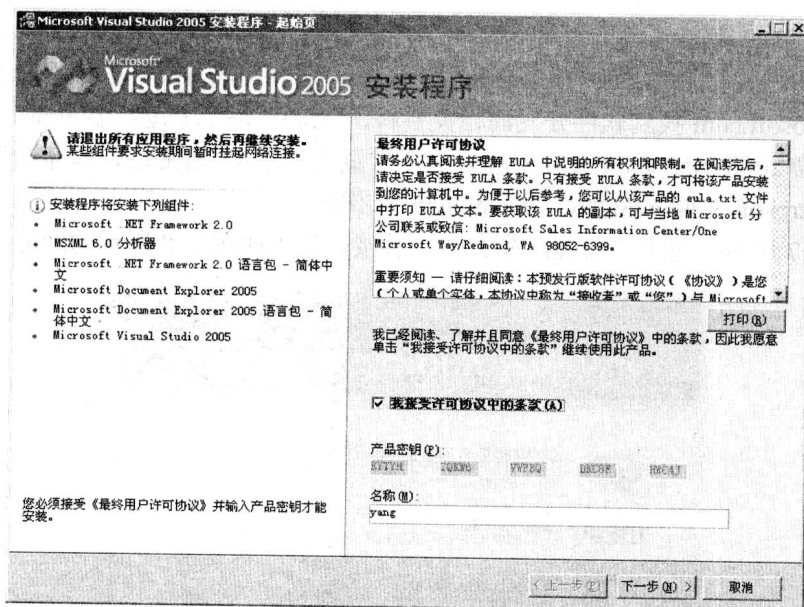


图 1-5 Visual Studio 2005 安装程序起始页

④ 单击“下一步”按钮，出现如图 1-6 所示的安装选项页窗口。在此窗口中可以修改安装路径和安装模式。对于熟悉 Visual Studio 2005 的用户来说，可以选择完全或自定义安装模式，对于初学者最好选用默认的安装模式和默认路径。

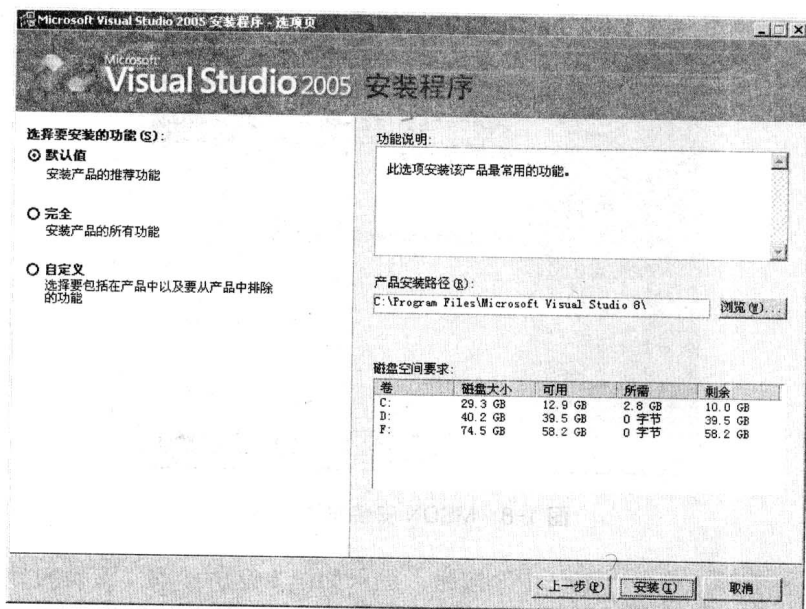


图 1-6 Visual Studio 2005 安装程序选项页

⑤ 单击“安装”按钮进入安装过程，安装过程中安装程序会复制程序文件并对系统进行必要的设置。最后出现安装成功窗口，单击“完成”按钮即可完成安装。