

<<C#2008程序设计基础案例教程>>

图书基本信息

书名：<<C#2008程序设计基础案例教程>>

13位ISBN编号：9787111301554

10位ISBN编号：7111301552

出版时间：2010-4

出版时间：机械工业出版社

作者：江南 编

页数：246

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

C#.NET是程序开发所应用的编程语言的一个主流方向,广泛应用于各类软件项目的开发。根据新浪网的调查,2007年IT技术人员主要使用的基于.NET平台的开发语言是C#,占到56.3%:C#吸取了Delphi, JAVA, C++, VB这些语言的很多优点。

从调查数据分析,目前使用.NETFramework2.0的用户多达67.4%。

显而易见,C#已经占尽商机,甚至已成为参与企业应聘时的第一项基本能力。

掌握C#语言将成为计算机专业学生在专业学习上的一种基本能力。

一本能够理论联系实际、使学生能够掌握基本编程经验的教材能使企业受益匪浅。

本书以应用广泛的C#2008平台为语言基础,用一个完整的项目贯穿全书,把项目分解成若干个模块,每个模块包含课程学习所需的基本知识点,在每一部分的知识点中先提出学习目标,再介绍解决方案,并对解决方案中涉及到的知识点作适当的展开,既有实践操作的步骤,又能够联系实际讲解理论知识,使学生不仅能够学习到操作要点,而且可以结合理论知识进一步提高对其的掌握,为学生学习其他知识打下良好的基础。

本书以高职高专院校计算机专业教学为主,内容通俗易懂,在内容的安排上力求循序渐进,由浅入深,先通过案例介绍让学生理解程序设计过程,使学生对程序有一个直观的了解,再详细阐述设计过程中用到的理论知识,并做到举一反三,同时配套了相应的实验和实训来加深对相应知识点的掌握。

。

<<C#2008程序设计基础案例教程>>

内容概要

《C#2008程序设计基础案例教程》主要介绍C#.NET平台、.NET框架及集成开发环境下的程序设计基础知识，分为数据基础、编程基础和高级应用3部分，旨在帮助读者建立程序设计的基本思想，掌握程序设计的基本方法。

《C#2008程序设计基础案例教程》用一个完整的项目贯穿全书，把项目分解成若干个模块，每个模块包含课程学习所需的基本知识点，在每一部分的知识点中先提出目标，再介绍解决方案，通过大量的实例和实验，并结合理论讲解来帮助学生进一步理解程序设计的方法和思路，掌握面向对象程序设计的基本方法与技术。

全书共分为12章，其中第12章是综合开发实例。

每一章后面均有习题，可供读者复习参考。

《C#2008程序设计基础案例教程》可作为高职高专院校计算机专业的教材或教学参考书，也可作为程序开发人员的入门培训教材或参考书。

书籍目录

出版说明前言第1章 概述 11.1 .NET的基本概念 11.2 .NET框架的工作原理 21.3 浏览开发环境 41.4 创建C#.NET项目 71.4.1 案例功能 71.4.2 案例步骤 71.4.3 案例拓展 91.4.4 案例思考 111.5 了解编程的概念 111.6 处理Windows窗体 141.6.1 案例功能 141.6.2 案例步骤 151.6.3 案例拓展 161.6.4 案例思考 171.7 处理控件 171.7.1 案例功能 171.7.2 案例步骤 171.7.3 案例拓展 181.7.4 案例思考 191.8 代码风格 191.9 习题 20第2章 C#的数据类型 222.1 数据类型介绍 222.2 变量和常量 252.3 数据类型转换 272.3.1 数值转换 272.3.2 装箱和拆箱转换 302.4 创建和使用结构及枚举 312.4.1 结构 312.4.2 枚举 332.5 在数组中存储数据 342.6 习题 35第3章 C#语法基础 373.1 表达式 373.1.1 运算符与表达式类型 373.1.2 运算符的优先级 383.2 选择结构 393.2.1 if语句 393.2.2 switch语句 413.3 循环语句 433.4 跳转语句 493.4.1 break语句 493.4.2 continue语句 503.4.3 return语句 503.5 习题 52第4章 函数 544.1 函数的创建 544.1.1 案例功能 544.1.2 案例步骤 544.1.3 案例拓展 564.1.4 案例思考 564.2 使用函数 574.2.1 案例功能 574.2.2 案例步骤 574.2.3 案例拓展 584.2.4 案例思考 604.3 使用预定义函数 604.3.1 案例功能 614.3.2 案例步骤 614.3.3 案例拓展 624.3.4 案例思考 634.4 习题 64第5章 调试与异常处理 655.1 程序调试 655.1.1 案例功能 655.1.2 案例步骤 665.1.3 案例拓展 685.1.4 案例思考 715.2 错误与异常处理 715.2.1 案例功能 725.2.2 案例步骤 735.2.3 案例拓展 745.2.4 案例思考 785.3 习题 78第6章 用户界面设计 796.1 数据选择 796.1.1 案例功能 796.1.2 案例步骤 806.1.3 案例拓展 836.1.4 案例思考 856.2 容器 856.2.1 案例功能 856.2.2 案例步骤 856.2.3 案例拓展 866.2.4 案例思考 886.3 菜单 886.3.1 案例功能 886.3.2 案例步骤 896.3.3 案例拓展 916.3.4 案例思考 916.4 工具栏 926.4.1 案例功能 926.4.2 案例步骤 926.4.3 案例拓展 936.4.4 案例思考 936.5 状态栏 936.5.1 案例功能 946.5.2 案例步骤 946.5.3 案例拓展 946.5.4 案例思考 956.6 验证窗体数据 956.6.1 案例功能 956.6.2 案例步骤 956.6.3 案例拓展 966.6.4 案例思考 986.7 其他常用控件 986.7.1 案例功能 986.7.2 案例步骤 986.7.3 案例拓展 1006.7.4 案例思考 1016.8 习题 101第7章 面向对象编程 1037.1 理解类 1037.1.1 案例功能 1047.1.2 案例步骤 1047.1.3 案例拓展 1047.1.4 案例思考 1057.2 使用类 1057.2.1 案例功能 1057.2.2 案例步骤 1057.2.3 案例拓展 1067.2.4 案例思考 1067.3 继承 1067.3.1 案例功能 1067.3.2 案例步骤 1077.3.3 案例拓展 1077.3.4 案例思考 1077.4 多态和命名空间 1077.4.1 案例功能 1087.4.2 案例步骤 1087.4.3 案例拓展 1087.4.4 案例思考 1087.5 属性 1097.5.1 案例功能 1097.5.2 案例步骤 1097.5.3 案例拓展 1107.5.4 案例思考 1107.6 委托、事件 1107.6.1 案例功能 1107.6.2 案例步骤 1117.6.3 案例拓展 1147.6.4 案例思考 1147.7 实训 1147.8 习题 114第8章 数据流和文件 1168.1 System.IO命名空间简介 1168.1.1 案例功能 1168.1.2 案例步骤 1178.1.3 案例拓展 1198.1.4 案例思考 1208.2 文件及文件夹的管理 1218.2.1 案例功能 1218.2.2 案例步骤 1218.2.3 案例拓展 1258.2.4 案例思考 1308.3 文本文件和读写操作 1308.3.1 案例功能 1318.3.2 案例步骤 1318.3.3 案例拓展 1338.3.4 案例思考 1348.4 习题 135第9章 ADO.NET基础 1369.1 ADO.NET简介 1369.1.1 案例功能 1379.1.2 案例步骤 1389.1.3 案例拓展 1399.1.4 案例思考 1429.2 连接操作数据源 1429.2.1 案例功能 1449.2.2 案例步骤 1449.2.3 案例拓展 1469.2.4 案例思考 1489.3 构建DataSet 1489.3.1 案例功能 1499.3.2 案例步骤 1499.3.3 案例拓展 1509.3.4 案例思考 1529.4 使用数据适配器 1529.4.1 案例功能 1539.4.2 案例步骤 1549.4.3 案例拓展 1579.4.4 案例思考 1619.5 实训 1619.6 习题 162第10章 如何提高代码的重用性 16310.1 存储过程的使用 16310.1.1 案例功能 16610.1.2 案例步骤 16610.1.3 案例拓展 17010.1.4 案例思考 17010.2 自定义类的使用 17010.2.1 案例功能 17110.2.2 案例步骤 17210.2.3 案例拓展 17810.2.4 案例思考 17810.3 实训 17810.4 习题 179第11章 ADO.NET和XML 18011.1 Web Service简介 18011.2 使用XML Web Service 18111.2.1 案例功能 18311.2.2 案例步骤 18311.2.3 案例拓展 18411.2.4 案例思考 18411.3 使用ADO.NET读取和写入XML 18411.3.1 案例功能 18511.3.2 案例步骤 18511.3.3 案例拓展 18611.3.4 案例思考 18811.4 构建和使用Web Service 18811.4.1 案例功能 18911.4.2 案例步骤 19011.4.3 案例拓展 19111.4.4 案例思考 19111.5 实训 19111.6 习题 191第12章 图书管理系统综合开发指南 19312.1 系统需求分析与设计 19312.1.1 需求分析 19312.1.2 系统设计 19312.2 数据库设计 19412.2.1 数据库的需求分析 19512.2.2 数据库的逻辑结构设计 19512.2.3 创建数据表间的关系 19612.3 系统设计过程 19612.3.1 设计图书管理系统的界面 19612.3.2 数据库通用模块设计 20912.3.3 存储过程设计 21412.3.4 用户登录模块设计 21512.3.5 图书查询模块设计 21712.3.6 借阅查询模块设计 22312.3.7 添加书类模块设计 22312.3.8 书库管理模块设计 22412.3.9 借书模块设计 22612.3.10 还书模块设计 23012.3.11 读者管理模块设计 23212.3.12 备忘录模块设计 23612.3.13 窗口显示方式设计 23912.4 系统的部署与维护 24012.4.1

Windows Installer简介 240
12.4.2 部署C# Windows应用程序 241
12.4.3 安装及卸载C# Windows应用程序 244
12.5 实训 245
12.6 习题 245
参考文献 246

章节摘录

在.NET框架公布于众之前，在Windows平台上所进行的面向构件的软件开发大都采取COM构件的形式。

组件对象模型（Component Object Model，COM）也称为构件对象模型，是用于创建可重用二进制构件的一个编程标准。

按照这个标准，开发者可以编写较少的代码来解决较小的问题域中的问题。

通过将问题分解为构件，从而简化解决方案，用于解决某一问题的构件或解决其他类似问题。

COM在具有诸多优点的同时，也存在不少缺点。

COM标准中最常见的一个问题，被业界比喻为“DLL地狱”。

当COM接口在注册表中被登记和索引之后，如果又发布了COM构件的新版本，“DLL地狱”问题便会产生。由于COM标准与二进制构件的紧耦合性质，版本控制经常让软件开发者感到头疼。

不仅如此，如果DLL在文件系统中的存储位置发生变化，而DLL在注册表中的信息没有被相应地修改，则DLL中的COM接口将成为不可访问的接口。

例如，当我们安装了一个新的应用程序后，如果这个应用程序使用了与其他应用程序共享的某个构件的新版本，便可能导致新安装的应用程序不能正常工作，还会破坏使用这个共享构件的所有其他应用程序。

COM构件在软件开发中还存在其他一些问题，包括内存管理困难、拖延开发时间、所提供的典型GUI控件不能完全满足许多开发任务的需要，以及缺乏语言之间的互操作性，如C++和Visual Basic之间的互操作性等。

为了解决软件开发中的上述问题和其他问题，Microsoft开始研究后来被称为COM+2.0的解决方案。这一解决方案旨在设计实现一个托管环境。

在该环境中，执行的代码提供了增强的类型安全性、代码安全性以及丰富的几乎令人难以置信的实用类库和函数库，使得开发者更容易完成开发工作。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>