第一图书网, tushu007.com

<<C#程序设计案例教程>>

图书基本信息

书名：<<C#程序设计案例教程>>

13位ISBN编号：9787302270195

10位ISBN编号：7302270198

出版时间：2012-1

出版时间：清华大学出版社

作者：蔡朝晖，安向明，张宇　编著

页数：321

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

内容概要

本书应用“项目驱动”最新教学模式，通过完整的项目案例系统地介绍了使用c#进行程序设计与开发的方法和技术。
全书论述了c#核心技术概述、c#基础语法、流程控制、c#面向对象核心语法、c#面向对象高级语法、异常处理、使用winform建立用户图形界面、多线程高级编程、文件与流、tcp／udp网络编程以及ado．net数据库开发等内容。

本书注重理论与实践相结合，内容详尽，提供了大量实例，突出应用能力的培养，将一个实际项目的知识点分解在各章作为案例讲解，是一本实用性突出的教材。
本书适合作为普通高等院校计算机专业c#课程的教材，也可供设计开发人员学习参考。

<<C#程序设计案例教程>>

书籍目录

第1章　c#核心技术概述
　1.1　microsoft.net介绍
　1.2　c#概述
　1.3　c#程序开发实例
　本章总结
　习题
第2章　艾斯医药系统项目案例介绍
　2.1　项目概述
　2.2　需求分析
　2.3　系统分析设计
　2.4　项目运行指南
第3章　c#基础语法
　3.1　c#基本语法要求
　3.2　数据类型
　3.3　常量与变量
　3.4　运算符和表达式
　3.5　数据类型转换
　3.6　项目案例
　本章总结
　习题
第4章　流程控制
　4.1　顺序流程
　4.2　分支流程
　4.3　循环流程
　4.4　跳转流程
　4.5　项目案例
　本章总结
　习题
第5章　c#面向对象核心语法
　5.1　面向对象的概念
　5.2　封装
　5.3　继承
　5.4　多态
　5.5　项目案例
　本章总结
　习题
第6章　c#面向对象高级语法(一)
　6.1　静态变量和方法
　6.2　密封类和方法
　6.3　抽象类和抽象方法
　6.4　接口
　6.5　项目案例
　本章总结
　习题
第7章　c#面向对象高级语法(二)

7.1 运算符重载
7.2 数组
7.3 字符串
7.4 集合
7.5 委托与事件
7.6 泛型
7.7 项目案例
本章总结
习题
第8章 异常处理
8.1 异常处理机制
8.2 捕获异常
8.3 使用finally块
8.4 抛出异常
8.5 项目案例
本章总结
习题
第9章 使用winform建立用户图形界面
9.1 窗体编程概述
9.2 窗体编程基础
9.3 窗体控件和组件简介
9.4 项目案例
本章总结
习题
第10章 多线程高级编程
10.1 多线程的概念
10.2 线程状态
10.3 线程的同步
10.4 线程池
10.5 项目案例
本章总结
习题
第11章 文件与流
11.1 文件系统中的目录和文件管理
11.2 基于流的文件读写操作
11.3 xml文件操作
11.4 项目案例
本章总结
习题
第12章 tcp／udp网络编程初步
12.1 网络编程简介
12.2 基于tcp／ip的网络编程
12.3 基于udp／ip的网络编程
12.4 项目案例
本章总结
习题
第13章 ado.net数据库开发

13.1 ado.net概述
13.2 连接数据库
13.3 ado.net和数据库的交互
13.4 数据集dataset
13.5 项目案例
本章总结
习题
感谢

章节摘录

1.C语言　　C语言是由20世纪60年代的结构化程序设计发展而来的。
在结构化程序设计之前，由于程序逻辑易于退化为所谓的“绝缘代码”（由大量紊乱且难以跟踪的跳转、调用和返回所导致），所以大型程序很难编写。
结构化语言通过添加明确的控制语句、带有局部变量的子程序和其他的改进来处理这种问题。
使用结构化语言，使得编写适度的大型程序成为可能。
C语言将强大功能、简洁性和可表达性成功地结合到了一起，成为20世纪80年代应用最广泛的结构化程序设计语言。
　　但是，随着程序设计的发展，C语言的局限性逐渐暴露：一旦工程达到一定大小，C程序就常常变得难以理解和维护。
该局限性与程序、程序设计员和所使用的工具等有关。
　　2.C++　　20世纪70年代后期，一些工程的规模达到了结构化程序设计方法学和C语言能处理的极限。
为了解决这个问题，开始引入新的编程思想，即OOP（Object-Oriented Programming）。
1979年，Bjarne Stroustrup在新泽西州Murray Hill的Bell实验室开始开发C++语言，希望在C的基础之上添加围绕面向对象程序设计而展开的大部分功能，被称为“有类的C语言”，后来更名为C++，因此，从本质上来讲，C++是C的面向对象版本。
基于C基础，Stroustrup提供了一种实现OOP的灵活移植方式。
C程序设计员不需要学习整个新语言，只需要学习一些新的面向对象方法学的特征。
　　C++从20世纪80年代开始慢慢得到应用，并得到了很大的发展。
在20世纪90年代，C++成为主流，使用它的人成倍增加。
而到了20世纪90年代末，它成为应用最广泛的程序设计语言。
　　但是要注意一点：C++的开发不是为了创建一种新的程序设计语言。
相反，它只是C语言的扩充和改进。
　　3.Java　　程序语言的下一步改进是Java语言。
1991年，Sun Microsystems率先使用Java。
Java沿用了C++的语法和基本原理，最大的创新在于跨平台、计算环境的可移植性。
当时，随着Internet的崛起，许多不同类型的CPU和操作系统连接到一起，程序很难从一个环境移植到另一个环境。
Java将程序的源代码翻译成中间语言，从而实现了移植性。
　　Java以C为基础，而其对象模型是由C++进化而来，Gosling在开发Java时也不需要创建整个新语言，而是把精力集中于新的、改进的特征。
随着Java的创建，C和C++成为建立新计算机语言的公认底层基础。
　　4.C#　　Java语言的开发成功解决了Internet环境中的可移植问题，但是它还有自身欠缺的特征。
　　（1）缺少交叉语言的互操作性。
　　……

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