

图书基本信息

书名：<<.NET平台与C#面向对象程序设计>>

13位ISBN编号：9787121103360

10位ISBN编号：7121103362

出版时间：2010-4

出版时间：电子工业出版社

作者：周羽明，刘元婷 编著

页数：420

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

由于经济危机等不利因素的影响，世界经济处在一种不确定中。IT行业也不能独善其身，同样面临着严峻的挑战。

很多IT企业开始收缩产品线，裁减开发团队规模以应对这场危机。

然而在这样的形势下，我们看到世界基础软件开发以及中国的外包产业却逆风飞扬，呈现出一种前所未有的所谓“危机、危机、危中寻机”的态势。

十年寒窗，等我们毕业走向社会以后，却发现自己学到的知识与社会所有脱节。

特别是计算机行业，技术发展日新月异。

但是在学校所学知识真的就没有用么？

不！

这就像武侠小说，这十年我们已经练就了内功，但是却不会一套拳法、剑法，怎么能闯荡江湖。

特别是计算机专业的学生，数学和计算机基础的学习，已经让我们有了不浅的内功，只需要把这些内功发挥出来。

所以，我们可能需要的就是一套武林最正派的外家功夫！

.NET技术丛书 微软公司一直引领IT行业的发展，平台占据市场绝大多数份额。

而对于一个计算机专业的从业人员来说，对微软整体技术的把握与发展，也是大多数IT从业人员的必然选择。

“ .NET技术丛书 ” 将带领我们从基础开始进入微软平台开发领域，本套丛书包含：《.NET平台与C#面向对象程序设计》、《.NET平台下Windows程序设计》、《.NET平台下Web程序设计》。

三本书分别面向基础的语言与面向对象的思想，Windows平台与Web平台。

提供最实用的市场主流知识和技术实训实验，让我们全面掌握微软开发平台的方方面面。

本套丛书全部作者均来自一线开发人员，具有多年的实践项目经验，除封面署名作者外其他参与编写人员有：张丹、许建平、朱益铭、沈宁、俞峰。

按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验最快地学习所有技术的一招一式。

除了知识点以外，详细地讲解了150多个实验，手把手地带领读者从零开始，进入到.NET开发的各个方面的知识点。

200多个基础项目实验的源码，而当我们学习知识和实验后，还有4个不同方向的中小型真实项目源码供我们理解，掌握它们以后就可以达到胜任著名外企开发职位或一般企业初级项目经理职位的水准。

到此，我们可以真正地下山，闯荡江湖了！

？

内容概要

微软公司一直引领IT行业的发展，平台占据市场绝大多数份额。

对微软整体技术的把握与发展，也是大多数IT从业人员的必然选择。

本书带读者全面学习掌握微软.NET平台、.NET平台上最重要的语言C#、C#不同的版本之间的区别、面向对象的编程思想以及UML的基础知识。

相信通过学习，您可以全面的掌握.NET平台基础知识和C#面向对象的程序设计。

本书侧重实用性，按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验让我们以最快的速度全面掌握微软平台与技术。

本书可作为高校专业学生的专业课参考书目，初学者的学习资料，更是所有.NET平台C#爱好者的实验指导书。

书籍目录

第1章 微软.NET平台介绍 1.1 .NET Framework概述 1.1.1 Microsoft .NET计划 1.1.2 .NET Framework 1.2 公共语言运行库 1.3 .NET Framework类库 1.3.1 .NET Framework类库概述 1.3.2 命名约定 1.3.3 重要命名空间说明 1.4 通用类型系统 1.4.1 值类型 1.4.2 引用类型 1.5 托管代码的执行过程 1.6 托管模块 1.7 元数据 1.8 程序集 1.8.1 单文件程序集 1.8.2 多文件程序集 1.8.3 程序集的功能 1.8.4 程序集解决DLL Hell问题 1.8.5 两种程序集和两种部署方式 1.9 应用程序域 1.9.1 应用程序域和程序集 1.9.2 应用程序域和线程 1.9.3 编程应用程序域 1.10 内存管理 1.10.1 C++开发人员 1.10.2 Visual Basic开发人员 1.10.3 COM开发人员 1.11 异常处理 1.12 代码设计规范 1.12.1 大小写样式 1.12.2 标识符的大小写规则 1.13 命名规则 1.13.1 命名类、结构和接口的规则 1.13.2 命名类成员的规则 1.13.3 命名参数的规则 1.13.4 命名命名空间 1.13.5 命名资源 1.14 .NET Framework .5新特性 1.14.1 .NET Compact Framework 1.14.2 ASP.NET 1.14.3 Common Language Runtime 1.14.4 Threading 1.14.5 Networking 1.14.6 Windows Communication Foundation 1.14.7 Windows Forms 第2章 微软.NET平台动手实验 2.1 实验要求与实验目标 2.1.1 实验要求 2.1.2 实验目标 2.2 实验1：安装Microsoft .NET Framework SDK 2.2.1 实验目标 2.2.2 实验步骤 2.3 实验2 创建一个简单的.NET应用程序 2.3.1 实验目标 2.3.2 实验步骤 2.3.3 代码分析 2.4 实验3：值类型与引用类型 2.4.1 实验目标 2.4.2 实验步骤 2.4.3 装箱拆箱操作 2.4.4 转换参考 2.5 实验4：查看元数据 2.5.1 实验目标 2.5.2 实验步骤 2.6 实验5：使用程序集编程 2.6.1 实验目标 2.6.2 实验步骤 2.7 实验6：异常处理 2.7.1 实验目标 2.7.2 实验步骤 2.7.3 异常处理的最佳实践 2.8 实验7：使用.NET Framework工具 2.8.1 实验目标 2.8.2 实验步骤 Steps 2.9 实验8：FxCop 2.9.1 实验目标 2.9.2 实验步骤 Steps 2.10 实验9：使用Visual Studio创建应用程序 2.10.1 实验目标 2.10.2 实验步骤 Steps 2.10.3 Visual Studio IDE 2.11 实验10：控制程序集版本 2.11.1 实验目标 2.11.2 实验步骤 第3章 C#语言（2.0版本~3.5版本） 3.1 第1个C#应用程序 3.1.1 Hello World 3.1.2 Main()和命令行参数 3.2 数据类型 3.2.1 基元类型 3.2.2 整数类型 3.2.3 布尔类型 3.2.4 实数类型 3.2.5 字符类型 3.2.6 Checked与Unchecked 3.2.7 值类型与引用类型 3.2.8 值类型的装箱与拆箱 3.3 数组 3.3.1 作为对象的数组 3.3.2 一维数组 3.3.3 多维数组 3.3.4 交错数组 3.3.5 对数组使用foreach 3.3.6 将数组作为参数传递 3.3.7 使用ref和out传递数组 3.4 结构及枚举 3.4.1 使用结构 3.4.2 枚举 3.5 变量和常量 3.5.1 变量 3.5.2 常量 3.6 类型转换 3.6.1 隐式转换 3.6.2 显式类型转换 3.7 操作符及表达式 3.7.1 操作符及表达式 3.7.2 算术操作符 3.7.3 赋值操作符和表达式 3.7.4 关系操作符和表达式 3.7.5 逻辑操作符和表达式 3.7.6 移位运算符 3.7.7 条件逻辑运算符 3.7.8 条件运算符 3.8 语句 3.8.1 选择语句 3.8.2 迭代语句 3.8.3 跳转语句 3.8.4 异常处理语句 3.9 类及其成员 3.9.1 类 3.9.2 成员 3.10 委托及事件 3.10.1 委托 3.10.2 使用委托 3.10.3 事件 3.10.4 使用事件 3.11 泛型 3.11.1 泛型类和泛型方法 3.11.2 泛型的优点 3.11.3 泛型类型参数 3.11.4 泛型类 3.11.5 泛型方法 3.12 迭代器 3.12.1 概述 3.12.2 使用 3.12.3 yield语句 3.13 异常及其处理 3.14 命名空间 3.15 迭代程序 3.16 迭代程序实现 3.17 递归迭代 3.18 局部类型 3.19 匿名方法 3.19.1 传递参数到匿名方法 3.19.2 实现 3.19.3 一般匿名方法 3.19.4 示例 3.20 委托推理 3.21 属性和索引的可见性 3.22 静态类 3.22 全局命名空间限定符 3.23 内联警告 3.24 .NET .5平台的新性能 3.24.1 隐式类型本地变量 3.24.2 为对象和集合初始值设定项 3.24.3 匿名类型 3.24.4 扩展方法 3.24.5 自动实现属性 3.24.6 分部方法 第4章 C#语言实验 4.1 实验要求与实验目标 4.1.1 实验要求 4.1.2 实验目标 4.2 命令行参数 4.2.1 实验目标 4.2.2 实验步骤 4.3 方法 4.3.1 实验目标 4.3.2 实验步骤 4.4 值和枚举类型 4.4.1 实验目标 4.4.2 实验步骤 4.5 属性 4.5.1 实验目标 4.5.2 实验步骤 4.6 属性编程 4.6.1 实验目标 4.6.2 实验步骤 4.7 XML文档注释 4.7.1 实验目标 4.7.2 实验步骤 4.8 if-else语句 4.8.1 实验目标 4.8.2 实验步骤 4.9 switch语句 4.9.1 实验目标 4.9.2 实验步骤 4.10 循环语句 4.10.1 实验目标 4.10.2 实验步骤 4.11 变长参数 4.11.1 实验目标 4.11.2 实验步骤 4.12 数组 4.12.1 实验目标 4.12.2 实验步骤 4.13 结构 4.13.1 实验目标 4.13.2 实验步骤 4.14 C#语言与面向对象版本控制 4.14.1 实验目标 4.14.2 实验步骤 4.15 自定义转换 4.15.1 实验目标 4.15.2 实验步骤 4.16 抽象类 4.16.1 实验目标 4.16.2 实验步骤 4.17 const关键字 4.17.1 实验目标 4.17.2 实验步骤 4.18 readonly关键字 4.18.1 实验目标 4.18.2 实验步骤 4.19 静态成员 4.19.1 实验目标 4.19.2 实验步骤 4.20 值传递与引用传递 4.20.1 实验目标 4.20.2 实验步骤 4.21 索引器 4.21.1 实验目标 4.21.2 实验步骤 4.22 ref关键字 4.22.1 实验目标 4.22.2 实验步骤 4.23 out关键字 4.23.1 实验目标 4.23.2 实验步骤 4.24 as运算符 4.24.1 实验目标 4.24.2 实验步骤 4.25 is运算符 4.25.1 实验目标 4.25.2 实验步骤

4.26 sizeof方法 2.26.1 实验目标 2.26.2 实验步骤 4.27 Array类 4.27.1 实验目标 4.27.2 实验步骤 4.28 ArrayList类 4.28.1 实验目标 4.28.2 实验步骤 4.29 分部类 4.29.1 实验目标 4.29.2 实验步骤 4.30 委托 4.30.1 实验目标 4.30.2 实验步骤 4.31 事件 4.31.1 实验目标 4.31.2 实验步骤 4.32 委托的使用 4.32.1 实验目标 4.32.2 实验步骤 4.33 事件 4.33.1 实验目标 4.33.2 实验步骤 4.34 索引器1 4.34.1 实验目标 4.34.2 实验步骤 4.35 索引器2 4.35.1 实验目标 4.35.2 实验步骤 4.36 线程 4.36.1 实验目标 4.36.2 实验步骤 4.37 线程池 4.37.1 实验目标 4.37.2 实验步骤 4.38 线程同步和交互 4.38.1 实验目标 4.38.2 实验步骤 4.39 非托管代码 4.39.1 实验目标 4.39.2 实验步骤 4.40 代码安全性控制 4.40.1 实验目标 4.40.2 实验步骤 4.41 类库设计 4.41.1 实验目标 4.41.2 实验步骤 4.42 显式接口实现 4.42.1 实验目标 4.42.2 实验步骤 4.43 自定义属性类 4.43.1 实验目标 4.43.2 实验步骤 4.44 条件方法 4.44.1 实验目标 4.44.2 实验步骤 4.45 Console类增强 4.45.1 实验目标 4.45.2 实验步骤 4.46 抽象属性 4.46.1 实验目标 4.46.2 实验步骤 4.47 实例构造函数 4.47.1 实验目标 4.47.2 实验步骤 4.48 静态类与静态类成员 4.48.1 实验目标 4.48.2 实验步骤 4.49 反射 : GetType 4.49.1 实验目标 4.49.2 实验步骤 4.50 反射 : 晚期绑定 4.50.1 实验目标 4.50.2 实验步骤 4.51 泛型入门 4.51.1 实验目标 4.51.2 实验步骤 4.52 泛型类库 4.52.1 实验目标 4.52.2 实验步骤 4.53 泛型 4.53.1 实验目标 4.53.2 实验步骤 4.54 泛型与非泛型代码的性能比较1 4.54.1 实验目标 4.54.2 实验步骤 4.55 泛型与非泛型代码的性能比较2 4.55.1 实验目标 4.55.2 实验步骤 4.56 匿名委托1 4.56.1 实验目标 4.56.2 实验步骤 4.57 匿名委托2 4.57.1 实验目标 4.57.2 实验步骤 4.58 迭代器入门 4.58.1 实验目标 4.58.2 实验步骤 4.59 可空类型入门 4.59.1 实验目标 4.59.2 实验步骤 4.60 迭代器 4.60.1 实验目标 4.60.2 实验步骤 4.61 可空类型 4.61.1 实验目标 4.61.2 实验步骤 4.62 迭代器与Yield 4.62.1 实验目标 4.62.2 实验步骤 4.63 集合类 4.63.1 实验目标 4.63.2 实验步骤 第5章 面向对象的设计思想与UML 5.1 面向对象技术 5.1.1 面向对象的概念 5.1.2 面向对象分析 5.1.3 面向对象设计 5.1.4 面向对象编程 5.2 类及其成员 5.2.1 类 5.2.2 类成员 5.2.3 访问修饰符 5.2.4 静态类和静态类成员 5.3 构造函数和析构函数 5.3.1 构造函数 5.3.2 使用构造函数 5.3.3 实例构造函数 5.3.4 私有构造函数 5.3.5 静态构造函数 5.3.6 析构函数 5.4 方法 5.4.1 声明 5.4.2 参数 5.4.3 返回值 5.4.4 传递参数 5.4.5 重载 5.4.6 操作符重载 5.5 字段及属性 5.5.1 字段 5.5.2 属性 5.5.3 非对称访问器可访问性 5.6 索引器 5.6.1 使用索引器 5.6.2 属性和索引器间比较 5.7 嵌套类型 5.8 继承 5.8.1 继承 5.8.2 抽象类和密封类 5.8.3 多态性 5.8.4 Override和New使用指南 5.9 接口 5.9.1 接口及其成员 5.9.2 接口属性 5.9.3 接口中的索引器 5.9.4 接口实现 5.9.5 显式接口实现 5.9.6 抽象类和接口 5.10 分部类 5.11 UML基础 5.11.1 UML的出现 5.11.2 UML的内容 5.11.3 UML的主要特点 5.11.4 UML的应用领域 5.12 类设计器 5.12.1 功能 5.12.2 使用类关系图 5.12.3 设计类和类型 5.12.4 查看类关系图中的类型和关系 5.12.5 重构类和类型 第6章 面向对象的设计思想与UML实验 6.1 继承 6.1.1 实验目标 6.1.2 实验步骤 6.2 base关键字 6.2.1 实验目标 6.2.2 实验步骤 6.3 this关键字 6.3.1 实验目标 6.3.2 实验步骤 6.4 继承、封装和多态 6.4.1 实验目标 6.4.2 实验步骤 6.5 new关键字 6.5.1 实验目标 6.5.2 实验步骤 6.6 接口 6.6.1 实验目标 6.6.2 实验步骤 6.7 抽象方法与版本控制 6.7.1 实验目标 6.7.2 实验步骤 6.8 版本控制 6.8.1 实验目标 6.8.2 实验步骤 6.9 运算符重载1 6.9.1 实验目标 6.9.2 实验步骤 6.10 运算符重载2 6.10.1 实验目标 6.10.2 实验步骤

章节摘录

第1章 微软.NET平台介绍 1.1 .NET Framework概述 1.1.1 Microsoft.NET计划 .NET是为数字生活和信息时代的每一个个体、分子、单位和组织提供支撑，并且用来建立.NET体验的基础软件平台。

该平台提供编程模型和一些工具，可编程XML Web服务，保证让用户，而非应用程序来控制交互和交流的过程，也是一种向用户提供个性化、简单、一致且安全的应用程序、服务和设备的方法。

微软公司推出.NET计划的目标是将范围广泛的微软产品及其各种服务组织起来，置于各种互联设备共同的视野范围之内，其中包括服务器、固定和移动的设备等。

在技术层次，微软瞄准了如下3个层面。

- (1) Web服务。
- (2) 部署平台（服务器和客户端）。
- (3) 开发平台。

XML Web服务正在成为业界标准，各种异构的编程框架和软件系统要采用XML Web服务相互沟通。换言之，XML Web服务是工业界为了实现软件技术服务的顺畅提供而采用的一种通信服务标准，.NET框架可以通过提供模型和工具帮助开发人员实现这种服务。

采用XML Web通信标准，.NET提供包括数据、电子邮件、商务基础结构、集成的商务进程、Web应用和管理、安全，以及移动实时访问等服务。

开发人员使用Microsoft.NET框（Microsoft.NETFramework）来创建解决方案，开发实际的软件产品，帮助用户展开真正的实施。

编辑推荐

《.NET平台与C#面向对象程序设计》按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验最快地学习所有技术的一招一式。

提供所有的实验源代码，在学习知识和实验后，还有四个不同方向的中小型真实项目源码供我们理解，到此，我们可以真正地下山，闯荡江湖了！

如果你是计算机专业的毕业生，《.NET平台与C#面向对象程序设计》能最快地把大学的知识转换成就业的资本和能力，最快地发挥出我们的积累，创造机会。

如果想进入计算机行业，《.NET平台与C#面向对象程序设计》能让我们最快地学到最实用的技术，给我们带来更多的发展与工作机会，以及以后的方向。

未来是我们的！

市场最主流的技术与平台 专为快速学习和就业而设计 详细的实验步骤和讲解 手把手带您熟悉微软技术 知识+实验=快速掌握+就业

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>