

## <<C#语言程序设计>>

### 图书基本信息

书名：<<C#语言程序设计>>

13位ISBN编号：9787564114695

10位ISBN编号：756411469X

出版时间：2009-1

出版时间：东南大学出版社

作者：顾洪，李慧 编

页数：216

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>

## &lt;&lt;C#语言程序设计&gt;&gt;

## 前言

自2000年6月微软公司推出Microsoft .NET以来，.NET技术和相应产品已被广泛应用在信息技术领域的各个方面，C#语言越来越成为主流的开发语言。

学习面向对象windows应用程序设计应采用广泛使用且适合教学的工具，而C#语言是为.NET框架量身定做的新一代面向对象的语言，用该语言开发的.NET系统的商用软件较多，通过学习，毕业后学生可具有较好的工作基础。

编者在从事C#语言的教学过程中，很难为学生找到一本合适的C#语言教材。

主要是因为市场上此类教材一般是把C#语言、Windows应用程序开发和ASP.NET数据库开发三者结合在一起介绍，由于内容较多，学时有限，很难将这三部分内容讲深入、讲透彻，而且部分图书的内容难度较大，不适合初学者使用。

为了解决这个问题，使学生尽可能地掌握面向对象编程的思想和语言开发工具，作者编写了本书。

本书采用实例教学法，对于每个重要的知识点都有相应的示例。

这些示例简单明了、针对性强，可以帮助读者快速理解所学的内容。

本书适合作为高等院校相关专业及编程基础教学培训机构的教材，也可作为.NET开发人员的参考书。

全书按照教材体例编写，共12章，各章配有小结，总结相关知识点，章末配有习题以帮助读者巩固所学的内容。

第1章概述，简略介绍.NET技术和C#语言的特点；第2章通过一个简单的实例介绍C#语言程序的基本结构；第3章介绍C#语言中所支持的数据类型以及变量的定义等内容；第4章介绍数据类型间的转换，包括显式转换和隐式转换；第5章介绍C#中与表达式相关的基本内容；第6章介绍C#中的流程控制，这是编程必需的内容；第7章介绍面向对象程序设计的基本概念类及其相关的内容；第8章介绍类中的重要内容：方法；第9章介绍类的重要特性：继承；第10章介绍体现组件编程思想的接口；第11章介绍线程；第12章介绍用C#语言对文件对象进行处理等内容。

## <<C#语言程序设计>>

### 内容概要

C#是目前主流的程序设计语言之一，《C#语言程序设计》以Microsoft Visual Studio 2005为平台，系统地介绍了C#的数据类型、表达式、流程控制、面向对象编程、接口技术、线程技术和文件操作等内容。

《C#语言程序设计》注重理论性与实用性的结合，内容循序渐进，示例代码简洁易懂，每章内容都有相应的小结和习题，便于读者总结和练习，并免费提供源代码。

《C#语言程序设计》适合作为高等院校相关专业C#课程教材，也可供广大C#开发用户参考。

## 书籍目录

第1章 概述1.1.NET概述1.2.NET与C#1.2.1 支持多种编程语言的.NET结构框架1.2.2 面向.NET的全新开发工具C#1.3 C#语言的特点1.3.1 简洁的语法1.3.2 面向对象设计1.3.3 与web的紧密结合1.3.4 安全性与错误处理1.3.5 版本处理技术1.3.6 灵活性和兼容性1.4 公用语言运行时环境与公用语言规范1.4.1 理解CLR1.4.2 可操控执行的含义1.4.3 CLR的突出特色1.4.4 公用语言规范第2章 编写第一个应用程序2.1 Welcome程序2.2 代码分析2.2.1 名字空间2.2.2 类和类的方法2.2.3 程序的输入和输出2.3 运行程序2.4 添加注释2.5 小结第3章 数据类型3.1 值类型3.1.1 简单类型3.1.2 结构类型3.1.3 枚举类型3.2 引用类型3.2.1 类3.2.2 数组3.2.3 代理3.3 变量3.3.1 命名变量3.3.2 变量的类型3.4 常量3.5 泛型3.5.1 泛型的定义3.5.2 泛型的引用3.5.3 常用的泛型集合3.6 小结第4章 类型转换4.1 隐式类型转换4.1.1 隐式数值转换4.1.2 隐式枚举转换4.1.3 隐式引用转换4.2 显式类型转换4.2.1 显式数值转换4.2.2 显式枚举转换4.2.3 显式引用转换4.3 小结第5章 表达式5.1 操作符5.1.1 操作符的分类5.1.2 操作符的优先级5.2 算术操作符和算术表达式5.2.1 加法运算5.2.2 减法运算5.2.3 乘法运算5.2.4 除法运算5.2.5 求余运算5.3 赋值操作符和赋值表达式5.3.1 简单赋值5.3.2 复合赋值5.4 关系操作符和关系表达式5.4.1 比较运算5.4.2 is操作符5.4.3 as操作符5.4.4 关系表达式5.5 逻辑操作符和逻辑表达式5.5.1 逻辑操作符5.5.2 逻辑表达式5.6 位运算5.7 其他特殊操作符5.7.1 三元操作符5.7.2 自增和自减操作符5.7.3 new操作符5.7.4 typeof操作符5.8 小结第6章 流程控制6.1 条件分支语句6.1.1 if语句6.1.2 switch语句6.2 循环语句6.2.1 for语句6.2.2 foreach语句6.2.3 while语句6.2.4 do-while语句6.3 跳转语句6.3.1 break语句6.3.2 continue语句6.3.3 goto语句6.3.4 return语句6.4 循环与跳转综合举例6.5 异常处理6.5.1 try-catch语句6.5.2 try-catch-finally语句6.5.3 throw语句6.6 小结第7章 面向对象的程序设计7.1 面向对象的基本概念7.1.1 面向对象技术的由来7.1.2 基本概念7.2 类7.2.1 类的申明7.2.2 类的成员7.2.3 构造函数和析构函数7.2.4 属性7.3 常用类操作和数据处理7.3.1 Convert类7.3.2 String类7.3.3 StringBuilder类7.3.4 DateTime类和 TimeSpan类7.3.5 Math类7.3.6 Random类7.4 小结第8章 方法8.1 方法的申明8.2 方法中的参数8.2.1 值参数8.2.2 引用型参数8.2.3 输出参数8.2.4 数组型参数8.3 静态和非静态的方法8.4 方法的重载8.5 操作符重载8.5.1 问题的提出8.5.2 使用成员方法重载操作符8.6 小结第9章 继承9.1 C#的继承机制9.1.1 概述9.1.2 覆盖9.1.3 索引指示器9.1.4 base保留字9.2 多态性9.2.1 C#中的多态性9.2.2 虚方法9.2.3 在派生类中对虚方法进行重载9.3 抽象与密封9.3.1 抽象类9.3.2 抽象方法9.3.3 密封类9.3.4 密封方法9.4 继承中关于属性的一些问题9.5 小结第10章 接口10.1 组件编程技术10.2 接口定义10.2.1 申明10.2.2 接口成员的定义10.2.3 对接口成员的访问10.2.4 接口成员的全权名10.3 接口的实现10.3.1 类对接口的实现10.3.2 显式接口成员执行体10.4 接口实现的继承机制10.4.1 接口的继承10.4.2 接口的重实现10.5 抽象类与接口10.6 应用接口示例10.7 小结第11章 线程11.1 创建并启动线程11.1.1 创建线程11.1.2 启动线程11.2 控制线程的执行11.2.1 挂起线程11.2.2 恢复线程11.2.3 终止线程11.2.4 设置线程的优先级11.3 线程间的同步11.4 线程使用示例11.5 小结第12章 文件和流12.1 用流读写文件12.1.1 FileStream类读写字节12.1.2 BinaryReader、BinaryWriter类读写基本数据类型12.1.3 StreamReader和StreamWriter类读写字符串12.1.4 序列化12.1.5 Stream类的其他派生类12.2 File类和FileInfo类12.2.1 File类常用的方法12.2.2 判断文件是否存在12.2.3 删除文件12.2.4 复制文件12.2.5 移动文件12.2.6 设置文件属性12.2.7 获取文件的属性12.3 Directory类和DirectoryInfo类12.3.1 Directory类常用的方法12.3.2 判断目录是否存在12.3.3 创建目录12.3.4 删除目录12.3.5 移动目录12.3.6 获取目录下所有子目录12.3.7 获取目录下的所有文件12.3.8 设置目录属性12.4 文件读写示例12.4.1 文本文件的读写12.4.2 二进制文件的读写12.4.3 拆分和合并文件12.5 小结参考文献

## 章节摘录

第1章 概述 1.4 公用语言运行时环境与公用语言规范 了解了.NET的结构之后,来看.NET的I]用其结构创造的运行环境。

公用语言运行时环境,是C#及其支持的.NET平台开发工具的运行基础。

具体来说,它为用户的应用提供了以下益处:跨语言集成、跨语言异常处理、内存管理自动化、强化的安全措施、版本处理技术、组件交互的简化模型。

1.4.1 理解CLR .NET提供了一个运行时环境,叫做公用语言运行时,它管理着代码的执行,并使得开发过程变得更加简单,这是一种可操控的执行环境。

在这种环境下的跨语言集成、跨语言异常处理、增强的安全性、版本处理与开发支持、简单的组件交互模型以及调试服务等技术的实现,需要运行时环境能够向可操控代码提供服务。

语言编译器需要产生一种元数据,它将提供在使用语言中的类型、成员、引用的信息,元数据与代码一起存储,每个可加载的CLR映像均包含了元数据,运行时环境使用元数据定位并载人,在内存中展开对象实例、解决方法调用、产生本地代码、强制执行安全性,并建立运行时环境的边界。

运行时环境自动处理对象的展开与引用,当对象不再被使用时运行时环境负责它们的释放,被运行时环境这样的生命期管理的对象称为可操控代码。

自动内存管理消除了内存溢出,同时也解决了其他一些常见的语法错误,如果代码是可操控的,仍然可以在需要的时候使用非可控代码,或者在.NET应用中同时使用可控与非可控代码。

CLR使设计跨语言的组件与应用变得更加容易,以使不同语言设计的对象能够彼此间进行通信,并且它们的行为能够紧密地综合与协调。

举个例子,定义了一个类,可以在另一种不同的语言中从该类中派生一个类或者调用它其中的一个方法,也可以向另一种语言中类的方法传递该类的一个实例。

这种跨语言的集成之所以可能,是因为以运行时为目标的语言编译器与工具使用一种运行时所定义的公用类型系统,它们通过遵守运行时的规则、公用语言规范来定义新的类型生成,使用、保持并绑定类型。

.....

## <<C#语言程序设计>>

### 编辑推荐

《C#语言程序设计》共12章，第1章概述，简略介绍.NET技术和C#语言的特点；第2章通过一个简单的实例介绍C#语言程序的基本结构；第3章介绍C#语言中所支持的数据类型以及变量的定义等内容；第4章介绍数据类型间的转换，包括显式转换和隐式转换；第5章介绍C#中与表达式相关的基本内容；第6章介绍C#中的流程控制，这是编程必需的内容；第7章介绍面向对象程序设计的基本概念类及其相关的内容；第8章介绍类中的重要内容：方法；第9章介绍类的重要特性：继承；第10章介绍体现组件编程思想的接口；第11章介绍线程；第12章介绍用C#语言对文件对象进行处理等内容。各章配有小结，总结相关知识点，章末配有习题以帮助读者巩固所学的内容。

## <<C#语言程序设计>>

### 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>