

## <<C#技术开发综合应用>>

### 图书基本信息

书名：<<C#技术开发综合应用>>

13位ISBN编号：9787302225966

10位ISBN编号：7302225966

出版时间：2010-6

出版时间：清华大学

作者：温怀玉//陈长忆

页数：371

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>

## <<C#技术开发综合应用>>

### 前言

C#是在继承C++和Java等语言优点的基础上发展而来的。

新的C#集成于VisualStudio。

NET2008，是程序设计语言研究的重大成果，它能较好地适应软件工程的需要，是目前最主流的程序设计语言之一。

C#以微软。

NET作为工作平台，除了Windows基本功能外，在用户交互界面、Web应用、数据库应用等方面得到了广泛应用。

在网络编程时代，C#备受开发人员的欢迎。

《C#技术开发综合应用》特点 学习《C#技术开发综合应用》不需要太多的编程基础知识，我们将从最基本的开发环境和开发基础知识开始，逐步引领读者深入C#应用开发的各个方面，直至完成综合的C#应用程序。

《C#技术开发综合应用》的知识体系结构如下图所示，遵循了循序渐进的原则，引领读者从基础入门，逐步深入学习各个知识点，最后开发出完整的C#应用软件系统。

## <<C#技术开发综合应用>>

### 内容概要

本书遵循循序渐进的原则，引领读者从基础入门，逐步深入学习各个知识点，直至最后开发出完整的系统。

全书共11章，内容全面，体系结构合理，涵盖了最基本的理论概念，以知识理论的应用为主，通过大量的应用示例，由浅入深，循序渐进地进行了讲述，最后给出了完整的C#项目开发案例，使读者较快地实现从入门到精通。

本书适合作为高等院校计算机及其相关专业的教材，也可供C#程序员参考学习，或供C#技术培训班作为教材使用，可使开发人员迅速从入门走向精通，同时对于缺乏项目实战经验的程序员来说可快速积累项目开发经验。

另外，为了帮助读者更好地学习本书，我们还提供了配套的免费电子教案和全书所有示例的源代码（所有程序均经过了作者精心调试），读者可通过<http://www.tup.com.cn>下载。

同时，本书还配有习题参考答案，方便读者学习。

## &lt;&lt;C#技术开发综合应用&gt;&gt;

## 书籍目录

第1章 C#综合开发环境	1.1 .NET Framework	1.1.1 Microsoft .NET的概念	1.1.2 .NET Framework
的基本构成	1.1.3 .NET Framework的获取和安装	1.2 Visual Studio IDE	1.2.1 安装IIS
Visual Studio 2008的安装和卸载	1.2.3 Visual Studio 2008启动和退出	1.2.4 Visual Studio 2008集成开	1.2.2
发界面	1.3 Visual C#语言	1.3.1 Visual C#语言简介	1.3.2 Visual C#语言新特征
1.4 编写第一	1.4.1 基于控制台的应用程序	1.4.2 基于窗体的应用程序	1.5 疑难问题解析
个C#应用程序	1.4.1 基于控制台的应用程序	1.4.2 基于窗体的应用程序	1.5 疑难问题解析
1.6 本章小结	1.7 练习与实践	第2章 C#语言基础	2.1 常量与变量
2.1 常量与变量	2.1.1 变量	2.1.2 常量	2.1.1 变量
2.2 C#语言的基本数据类型	2.2.1 整数类型	2.2.2 浮点类型	2.2.3 十进制类型
2.2.4 符	2.2.5 布尔类型	2.2.6 类型转换	2.2.4 符
2.2.5 布尔类型	2.2.6 类型转换	2.3 C#语言的运算符和表达式	2.3.1 C#的运算符
2.3.1 C#的运算符	2.3.2 运算符的优先级	2.3.3 算术运算符与算术表达式	2.3.4 赋值运算符与赋值表达式
2.3.2 运算符的优先级	2.3.3 算术运算符与算术表达式	2.3.4 赋值运算符与赋值表达式	2.3.5 关系运算符与关系表达式
2.3.3 算术运算符与算术表达式	2.3.4 赋值运算符与赋值表达式	2.3.5 关系运算符与关系表达式	2.3.6 逻辑运算符与逻辑表达式
2.3.4 赋值运算符与赋值表达式	2.3.5 关系运算符与关系表达式	2.3.6 逻辑运算符与逻辑表达式	2.3.7 位运算符与位表达式
2.3.5 关系运算符与关系表达式	2.3.6 逻辑运算符与逻辑表达式	2.3.7 位运算符与位表达式	2.3.8 条件运算符与条件表达式
2.3.6 逻辑运算符与逻辑表达式	2.3.7 位运算符与位表达式	2.3.8 条件运算符与条件表达式	2.3.9 其他运算符
2.3.7 位运算符与位表达式	2.3.8 条件运算符与条件表达式	2.3.9 其他运算符	2.4 C#语言的控制语句
2.3.8 条件运算符与条件表达式	2.3.9 其他运算符	2.4 C#语言的控制语句	2.4.1 选择语句
2.3.9 其他运算符	2.4 C#语言的控制语句	2.4.1 选择语句	2.4.2 循环语句
2.4 C#语言的控制语句	2.4.1 选择语句	2.4.2 循环语句	2.4.3 跳转语句
2.4.1 选择语句	2.4.2 循环语句	2.4.3 跳转语句	2.5 C#语言的复杂数据类型
2.4.2 循环语句	2.4.3 跳转语句	2.5 C#语言的复杂数据类型	2.5.1 枚举
2.4.3 跳转语句	2.5 C#语言的复杂数据类型	2.5.1 枚举	2.5.2 结构
2.5 C#语言的复杂数据类型	2.5.1 枚举	2.5.2 结构	2.6 疑难问题解析
2.5.1 枚举	2.5.2 结构	2.6 疑难问题解析	2.7 本章小结
2.5.2 结构	2.6 疑难问题解析	2.7 本章小结	2.8 练习与实践
2.6 疑难问题解析	2.7 本章小结	2.8 练习与实践	第3章 面向对象
2.7 本章小结	2.8 练习与实践	第3章 面向对象	3.1 面向对象基本概念
2.8 练习与实践	第3章 面向对象	3.1 面向对象基本概念	3.2 类及其构成
第3章 面向对象	3.1 面向对象基本概念	3.2 类及其构成	3.2.1 类的定义
3.1 面向对象基本概念	3.2 类及其构成	3.2.1 类的定义	3.2.2 访问修饰符
3.2 类及其构成	3.2.1 类的定义	3.2.2 访问修饰符	3.2.3
3.2.1 类的定义	3.2.2 访问修饰符	3.2.3	3.2.4 类的成员
3.2.2 访问修饰符	3.2.3	3.2.4 类的成员	3.2.5 命名空间
3.2.3	3.2.4 类的成员	3.2.5 命名空间	3.3 字段成员
3.2.4 类的成员	3.2.5 命名空间	3.3 字段成员	3.3.1 常量成员
3.2.5 命名空间	3.3 字段成员	3.3.1 常量成员	3.3.2 变量成员
3.3 字段成员	3.3.1 常量成员	3.3.2 变量成员	3.4 方法成员
3.3.1 常量成员	3.3.2 变量成员	3.4 方法成员	3.4.1 方法的声明
3.3.2 变量成员	3.4 方法成员	3.4.1 方法的声明	3.4.2 方法的调用
3.4 方法成员	3.4.1 方法的声明	3.4.2 方法的调用	3.4.3 方法参数
3.4.1 方法的声明	3.4.2 方法的调用	3.4.3 方法参数	3.4.4 静态方法
3.4.2 方法的调用	3.4.3 方法参数	3.4.4 静态方法	3.4.5 方法的重载
3.4.3 方法参数	3.4.4 静态方法	3.4.5 方法的重载	3.5 属性成员
3.4.4 静态方法	3.4.5 方法的重载	3.5 属性成员	3.5.1 属性的定义
3.4.5 方法的重载	3.5 属性成员	3.5.1 属性的定义	3.5.2 get访问函数
3.5 属性成员	3.5.1 属性的定义	3.5.2 get访问函数	3.5.3 set访问函数
3.5.1 属性的定义	3.5.2 get访问函数	3.5.3 set访问函数	3.6 构
3.5.2 get访问函数	3.5.3 set访问函数	3.6 构	3.6.1 实例构造函数
3.5.3 set访问函数	3.6 构	3.6.1 实例构造函数	3.6.2 this关键字
3.6 构	3.6.1 实例构造函数	3.6.2 this关键字	3.6.3 重载构造函数
3.6.1 实例构造函数	3.6.2 this关键字	3.6.3 重载构造函数	3.6.4 静态构造函数
3.6.2 this关键字	3.6.3 重载构造函数	3.6.4 静态构造函数	3.7 索引器成员
3.6.3 重载构造函数	3.6.4 静态构造函数	3.7 索引器成员	3.8.1 索引器的定义
3.6.4 静态构造函数	3.7 索引器成员	3.8.1 索引器的定义	3.8.2 使用索引器
3.7 索引器成员	3.8.1 索引器的定义	3.8.2 使用索引器	3.9 疑难问题解析
3.8.1 索引器的定义	3.8.2 使用索引器	3.9 疑难问题解析	3.10 本
3.8.2 使用索引器	3.9 疑难问题解析	3.10 本	3.11 练习与实践
3.9 疑难问题解析	3.10 本	3.11 练习与实践	第4章 面向对象高级编程
3.10 本	3.11 练习与实践	第4章 面向对象高级编程	4.1 继承
3.11 练习与实践	第4章 面向对象高级编程	4.1 继承	4.1.1 基类与派生类
第4章 面向对象高级编程	4.1 继承	4.1.1 基类与派生类	4.1.2 base
4.1 继承	4.1.1 基类与派生类	4.1.2 base	4.1.3 隐藏基类成员
4.1.1 基类与派生类	4.1.2 base	4.1.3 隐藏基类成员	4.1.4 密封类
4.1.2 base	4.1.3 隐藏基类成员	4.1.4 密封类	4.1.5 Object类
4.1.3 隐藏基类成员	4.1.4 密封类	4.1.5 Object类	4.2 多态
4.1.4 密封类	4.1.5 Object类	4.2 多态	4.2.1 多态的概念
4.1.5 Object类	4.2 多态	4.2.1 多态的概念	4.2.2 多态的实现
4.2 多态	4.2.1 多态的概念	4.2.2 多态的实现	4.3 抽象类
4.2.1 多态的概念	4.2.2 多态的实现	4.3 抽象类	4.3.1 抽象类的定义
4.2.2 多态的实现	4.3 抽象类	4.3.1 抽象类的定义	4.3.2 抽象方法
4.3 抽象类	4.3.1 抽象类的定义	4.3.2 抽象方法	4.4 接口
4.3.1 抽象类的定义	4.3.2 抽象方法	4.4 接口	4.4.1 接口的概念
4.3.2 抽象方法	4.4 接口	4.4.1 接口的概念	4.4.2 接口的成员
4.4 接口	4.4.1 接口的概念	4.4.2 接口的成员	4.4.3 接口的实现
4.4.1 接口的概念	4.4.2 接口的成员	4.4.3 接口的实现	4.5 委托
4.4.2 接口的成员	4.4.3 接口的实现	4.5 委托	4.5.1 委托的定义
4.4.3 接口的实现	4.5 委托	4.5.1 委托的定义	4.5.2 委托的使用
4.5 委托	4.5.1 委托的定义	4.5.2 委托的使用	4.6 事件
4.5.1 委托的定义	4.5.2 委托的使用	4.6 事件	4.6.1 事件的定义
4.5.2 委托的使用	4.6 事件	4.6.1 事件的定义	4.6.2 事件的使用
4.6 事件	4.6.1 事件的定义	4.6.2 事件的使用	4.7 疑难问题解析
4.6.1 事件的定义	4.6.2 事件的使用	4.7 疑难问题解析	4.8 本
4.6.2 事件的使用	4.7 疑难问题解析	4.8 本	4.9 练习与实践
4.7 疑难问题解析	4.8 本	4.9 练习与实践	第5章 程序错误和异常处理
4.8 本	4.9 练习与实践	第5章 程序错误和异常处理	第6章 Visual Studio 2008中的Windows窗体与控件
4.9 练习与实践	第5章 程序错误和异常处理	第6章 Visual Studio 2008中的Windows窗体与控件	第7章 ASP.NET创建Web应用程序
第5章 程序错误和异常处理	第6章 Visual Studio 2008中的Windows窗体与控件	第7章 ASP.NET创建Web应用程序	第8章 ADO.NET数据库应用开发
第6章 Visual Studio 2008中的Windows窗体与控件	第7章 ASP.NET创建Web应用程序	第8章 ADO.NET数据库应用开发	第9章 在C#中使用XML
第7章 ASP.NET创建Web应用程序	第8章 ADO.NET数据库应用开发	第9章 在C#中使用XML	第10章 C#
第8章 ADO.NET数据库应用开发	第9章 在C#中使用XML	第10章 C#	的I/O与注册表操作
第9章 在C#中使用XML	第10章 C#	的I/O与注册表操作	第11章 C/S与B/S结合的综合示例——人事工资管理系统
第10章 C#	的I/O与注册表操作	第11章 C/S与B/S结合的综合示例——人事工资管理系统	习题参考答案
的I/O与注册表操作	第11章 C/S与B/S结合的综合示例——人事工资管理系统	习题参考答案	参考文献
第11章 C/S与B/S结合的综合示例——人事工资管理系统	习题参考答案	参考文献	

## 章节摘录

2. 解决方案资源管理器 Visual Studio 2008 提供了解决方案和项目两类容器，用于帮助用户有效地管理开发工作中所用到的项目，如外部库的引用、文件夹、源文件等，一个方案中可能包含几个项目。

解决方案资源管理器位于 IDE 右上方。

如果在 IDE 中已经创建了方案或项目，项目中的所有文件将在解决方案资源管理器窗口中以分层树视图的方式显示。

通过解决方案资源管理器，可以打开文件进行编辑、向项目中添加新文件，以及查看解决方案、项目和属性等。

选择“视图”—“解决方案管理器”命令或按 Ctrl+Alt+L 键，即可打开解决方案资源管理器窗口。

解决方案资源管理器中用粗体显示的项目是启动项目，即程序执行的开始点。

可以在任意项目上单击鼠标右键，在弹出的快捷菜单中选择“设置为启动”命令，将该项目设置为启动项目。

解决方案资源管理器窗口上方有如下几个按钮。

- “查看代码”：为选定的文件打开代码编辑窗口。
- “刷新”：刷新当前方案的状态。
- “显示所有文件”：显示当前方案中包含的所有文件夹和文件。
- “属性”：显示当前选定对象的特定性质。
- “类视图”：显示当前项目中包含的所有类、方法、字段、属性、事件等信息。

3. 属性窗口 属性窗口位于 IDE 的右下方，主要用来查看项目、类或控件的各种特性。

在 Windows 窗体的“设计”视图下，在属性窗口中可以设置控件的属性并链接用户界面控件的事件。在某个控件上单击鼠标右键，在弹出的快捷菜单中选择“属性”命令，此时在属性窗口中就会显示出该控件的属性。

右击解决方案资源管理器中的“属性”节点，在弹出的快捷菜单中选择“打开”命令，在打开的对话框中选择“项目设计器”选项卡，从中可以修改生成选项、安全要求、部署详细信息以及设置许多其他项目属性。

4. 编译器、调试器和错误列表窗口 Visual Studio 提供了一套可靠的生成和调试工具。

在编译项目之前，可以在项目设计器中设置编译器选项。

在编译时，只需选择“生成”-“生成”命令，IDE 就会调用 C# 编译器。

生成过程是调试过程的开始，它可帮助用户检测编译时错误（包含不正确的语法、写错的关键字和输入的不匹配）。

如果应用程序编译成功，则状态栏中将显示“生成成功”的消息；如果存在编译错误，则将在设计器窗口的下方出现带有错误说明的“错误列表”窗口。

双击某个错误提示行可以转到源代码中相应的问题行；按 F1 键可以查看该项错误的帮助文档。

调试器可以显示为多种类型的窗口，随着应用程序的运行而显示变量的值和类型信息。

在调试器中调试时，可以使用代码编辑器窗口指定在某一行暂停执行，也可以逐语句或逐过程执行代码。

## <<C#技术开发综合应用>>

### 编辑推荐

《C#技术开发综合应用》适合作为高等院校计算机及其相关专业的教材，也可供C#程序员参考学习，或供C#技术培训班作为教材使用。

<<C#技术开发综合应用>>

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>