第一图书网, tushu007.com

<<Visual C#2008核心编程>>

图书基本信息

书名：<<Visual C#2008核心编程>>

13位ISBN编号：9787302200819

10位ISBN编号：7302200815

出版时间：2009-5

出版时间：清华大学出版社

作者：（美）马歇尔 著，施平安 等译，耿肇英 审

页数：620

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

时间在流逝，技术也在迅猛发展。
在希冀中，微软的．NET战略早已经变成现实，带来全新、快速而敏捷的企业计算能力，也给软件开发商和软件开发人员提供了支持未来计算的高效Web服务开发工具。
在希冀中，我们欣喜地看到，微软的每一个技术创新，都对中国开发人员产生巨大的推动作用，使得越来越多的人加入微软开发阵营。
微软出版社为了配合VisualStudio的推广和普及，邀请项目开发组的核心开发人员和计算机图书专业作家精心编写了微软ITPro系列图书。
该丛书白面市以来，在美国图书销量排行榜上一直高居前列，颇受读者好评，成为程序开发人员和网络开发人员了解微软技术的权威工具书。
随着新的开发平台的发布，该系列得以大幅度扩充，在美国及欧洲图书市场广受好评。
从2002年开始，清华大学出版社为了满足中国广大程序开发人员、网络开发人员以及计算机用户学习最新技术的渴望，在微软出版社的配合下，先后推出了《微软．NET程序员系列》和《微软．NET程序设计系列》。
这两套书阵容庞大，几乎涵盖．NET技术及其应用的各个方面；也正因为如此，翻译和编辑加工的工作量也大得惊人。
但为了保持国外优秀技术图书的魅力，同时使读者领会新技术的真谛，本丛书的翻译和编辑都是经过严格筛选的、具有很高的翻译水平或丰富编辑经验的技术人员。
同时，我们还聘请微软公司相关产品组的技术专家审读每一本书，确保在技术上准确无误。

内容概要

本书围绕软件生命周期的设计、实现、维护和调试阶段展开讲述，系统地介绍了Visual C# 2008。
全书共分5个部分。
第Ⅰ部分介绍该语言的基本概念；第Ⅱ部分介绍创建一个C#应用程序所必需的核心技能，包括Visual C# 2008新引入的LINQ语言；第Ⅲ部分重点介绍额外的语言特征；第Ⅳ部分全面介绍如何调试托管代码；第Ⅴ部分介绍内存管理和不安全代码。
　本书内容丰富，著述清楚，结构上由浅入深，并提供了大量范例和提示，适合所有专业的开发人员和编程爱好者阅读。

作者简介

Donis Marshal是一名软件工程师、顾问和教师，具有近25年的开发经验，编写过成千上万行代码。Donis开设了30多门课程，并在全球各地培训开发人员，主题涉及C#、Microsoft Visual Basic、NET Framework，调试技术和最佳实践。
他著有多部作品，包括Programming Micarosoft Visual C# 2005：The Language和NET Security Programming。

第一图书网, tushu007.com

<<Visual C#2008核心编程>>

书籍目录

第Ⅰ部分 核心语言 第1章 Visual C#编程简介 1.1 Visual C# 2008编程示例 1.1.1 C#示例程序 1.1.2
LINQ示例程序 1.2 Visual C# 2008程序的公共要素 1.2.1 命名空间 1.2.2 Main入口点 1.2.3 局部
量 1.2.4 可空类型 1.2.5 表达式 1.2.6 选择语句 1.2.7 迭代语句 1.3 C#核心语言特征
号和标记 1.3.2 标识符 1.3.3 关键字 1.3.4 基元 1.4 类型预览 第2章 类型 2.1 类 2.1
员 2.1.2 成员函数 2.2 结构 2.3 枚举 2.4 等值与标识 2.5 继承预览 第3章 继承 3.1 继承
3.2 System.Object 3.2.1 Object.Equals方法 3.2.2 Object.GetHashCode方法 3.2.3 Object.GetType方法
3.2.4 Object.ToString方法 3.2.5 Object.MemberwiseClone方法 3.2.6 Object.ReferenceEquals方法 3.3
Employee类 3.4 实现继承 3.5 重写继承的行为 3.5.1 virtual和override关键字 3.5.2 重载与重写
3.5.3 重写事件 3.5.4 扩展方法 3.6 new修饰符 3.7 抽象类 3.8 密封类 3.9 构造函数和析构函
3.10 接口 3.10.1 实现接口 3.10.2 显式接口成员实现 3.10.3 接口的再实现 3.11 多态性 3.1
接口多态性 3.11.2 new修饰符和多态性 3.12 强制类型转换 3.13 属性继承 3.14 Visual Studio 200
预览 第Ⅱ部分 核心技术 第4章 Visual Studio 2008简介 4.1 迁移到Visual Studio 2008 4.2 集成开发环境
4.2.1 起始页 4.2.2 创建项目 4.2.3 多目标 4.2.4 解决方案资源管理器 4.2.5 项目类型 4.
添加引用 4.2.7 Visual Studio中的窗口管理 4.2.8 自动恢复 4.3 类层次结构 4.3.1 “类视图”窗
4.3.2 对象浏览器 4.3.3 类关系图 4.3.4 “错误列表”窗口 4.4 代码编辑器 4.4.1 IntelliSense
4.4.2 外侧代码 4.4.3 字体和颜色格式化 4.4.4 源代码格式设置 4.4.5 修改跟踪 4.5 代码段
插入代码段 4.5.2 默认的代码段 4.5.3 代码段管理器 4.5.4 创建代码段 4.5.5 复制和粘贴
重构 4.7 生成和部署 4.7.1 MSBuild 4.7.2 ClickOnce部署 4.8 数组和集合预览 第5章 数组和
5.1 数组 5.1.1 数组元素 5.1.2 多维数组 5.1.3 交错数组 5.1.4 System.Array 5.1.5 System.A
属性 5.1.6 params关键字 5.1.7 数组转换 5.2 集合 5.2.1 ArrayList集合 5.2.2 BitArray集合
Hashtable集合 5.2.4 Queue集合 5.2.5 SortedList 5.2.6 Stack集合 5.2.7 专用集合 5.3 LINQ
第6章 LINQ简介 6.1 C#扩展 6.1.1 类型推断 6.1.2 对象初始值设定项 6.1.3 匿名类型 6.1.
展方法 6.1.5 lambda表达式 6.1.6 表达式目录树 6.2 LINQ基础 6.2.1 核心要素 6.2.2 转换
符 6.2.3 LINQ查询表达式语法 6.2.4 LINQ在哪里 6.3 LINQ to Objects 6.4 LINQ运算符 6.4.
合运算符 6.4.2 连接运算符 6.4.3 数据类型转换运算符 6.4.4 元素运算符 6.4.5 等于运算符
6.4.6 筛选运算符 6.4.7 生成运算符 6.4.8 分组运算符 6.4.9 联接运算符 6.4.10 分区运算符
6.4.11 量词运算符 6.4.12 集合运算符 6.4.13 排序运算符 6.5 泛型预览 第7章 泛型 7.1 泛型
7.1.1 类型形参 7.1.2 类型实参 7.1.3 构造类型 7.1.4 重载方法 7.2 泛型方法 泛型类型的
引用 7.3 约束 7.3.1 派生约束 7.3.2 接口约束 7.3.3 值类型约束 7.3.4 引用类型约束
认的构造函数约束 7.4 强制类型转换 7.5 继承 7.5.1 重写泛型方法 7.5.2 嵌套类型 7.6 静态
7.6.1 运算符函数 7.6.2 序列化 7.6.3 泛型本质 7.6.4 泛型集合 7.7 枚举器预览 第8章 枚
8.1 可枚举的对象 8.1.1 枚举器 8.1.2 枚举器实例 8.1.3 枚举器实例（版本化集合） 8.1.4
IEnumerator问题 8.2 泛型枚举器 8.2.1 IEnumerable接口 8.2.2 IEnumerator接口 8.2.3 泛型枚举
实例（版本化集合） 8.3 迭代器 8.4 运算符重载预览 第Ⅲ部分 C#语言详解 第9章 运算符重载
数学运算符和逻辑运算符 9.1.1 实现 9.1.2 自增和自减运算符 9.1.3 左移和右移运算符 9.1.4
true和false运算符 9.1.5 成对运算符 9.2 转换运算符 9.3 实例 9.4 运算符重载的本质 9.5 委托
件预览 第10章 委托和事件 10.1 委托 10.1.1 定义委托 10.1.2 创建委托 10.1.3 逆变和协变
10.1.4 激活委托 10.1.5 委托数组 10.1.6 System.MulticastDelegate类 10.1.7 调用列表 10.1.8 泛型
和委托 10.1.9 异步调用 10.1.10 异步委托图 10.1.11 异常 10.2 匿名方法 10.2.1 外部变量
10.2.2 泛型匿名方法 10.2.3 匿名方法的局限性 10.3 事件 10.3.1 发布事件 10.3.2 订阅者
引发事件 10.4 LINQ编程预览 第11章 LINQ编程 11.1 LINQ to XML 11.1.1 XML架构 11.1.2
11.1.3 导航 11.1.4 显式的强制类型转换 11.1.5 XML修改 11.1.6 XML查询表达式 11.2 LINQ
SQL 11.2.1 实体类 11.2.2 LINQ to SQL查询表达式 11.2.3 LINQ to DataSet 11.2.4 关联 11.
LINQ to SQL更新 11.3 异常处理预览 第12章 异常处理 12.1 异常示例 12.2 标准异常模型 12.3
构化异常处理 12.3.1 try语句 12.3.2 catch语句 12.3.3 异常的传播 12.3.4 finally语句 12.3.5

信息表 12.3.6 嵌套的try块 12.4 System.Exception 12.4.1 System.Exception函数 12.4.2 System.Exception属性 12.4.3 应用程序异常 12.4.4 异常转换 12.4.5 COM互操作性异常 12.5 远异常 12.6 未处理的异常 12.6.1 Application.ThreadException 12.6.2 AppDomain.Unhandled-Exception 12.7 在Visual Studio中管理异常 12.7.1 异常助手 12.7.2 “ 异常 ”对话框 12.8 元数据和反射预览 第IV部分 调试 第13章 元数据和反射 13.1 元数据 13.1.1 元数据记 13.1.2 元数据堆 13.1.3 流 13.1.4 元数据验证 13.1.5 ILDASM 13.2 反射 13.2.1 获得象 13.2.2 加载程序集 13.2.3 浏览类型信息 13.2.4 动态调用 13.2.5 类型的创建 13.2.6 晚定委托 13.2.7 函数调用性能 13.3 反射和泛型 13.3.1 IsGeneric和IsGenericTypeDefinition 13.3.2 typeof 13.3.3 GetType 13.3.4 GetGenericTypeDefinition 13.3.5 GetGenericArguments 13.3.6 创建型类型 13.3.7 反射的安全性 13.3.8 属性 13.3.9 程序员定义的自定义属性 13.3.10 属性和反射 13.4 MSIL预览 第14章 MSIL编程 14.1 “ Hello World ”应用程序 14.2 计算堆栈 14.3 MSIL详解 14.4 复杂任务 14.5 分支 14.6 数组 14.7 算术指令 14.8 转换运算 14.9 异常处理 14.10 其他 14.11 进程执行 14.12 用Visual Studio 2008进行调试预览 第15章 用Visual Studio 2008进行调试 15.1 调试概述 15.1.1 调试Windows窗体项目 15.1.2 附加到运行进程 15.1.3 调试控制台应用程序项目 15.1.4 调试类库项目 15.2 调试配置 15.2.1 Debug和Release配置 15.2.2 配置管理器 15.3 调试设 15.3.1 Visual Studio环境调试设置 15.3.2 解决方案的调试设置 15.3.3 项目的调试设置 15.4 断点 15.4.1 函数断点 15.4.2 “ 断点 ”窗口 15.4.3 跟踪点 15.5 代码逐步调试 15.5.1 逐步调试命令 15.5.2 “ 设置下一条语句 ”实例 15.6 “ 调试 ”工具栏 15.7 数据提示 15.8 调试窗口 15.8.1 “ 断 ”窗口 15.8.2 “ 输出 ”窗口 15.8.3 “ 监视 ”窗口和其他变量窗口 15.8.4 自动窗口 15.8.5 “ 部变量 ”窗口 15.8.6 “ 即时 ”窗口 15.8.7 “ 调用堆栈 ”窗口 15.8.8 “ 线程 ”窗口 15.8.9 “ 块 ”窗口 15.8.10 “ 进程 ”窗口 15.8.11 “ 内存 ”窗口 15.8.12 “ 反汇编 ”窗口 15.8.13 “ 寄存器 ”窗口 15.9 跟踪 15.9.1 跟踪实例 15.9.2 配置文件 15.9.3 使用配置文件进行跟踪的示例 15.10 DebuggerDisplayAttribute 15.11 DebuggerBrowsableAttribute 15.12 DebuggerTypeProxyAttribute 15.13 转储文件 15.14 高级调试预览 第16章 高级调试 16.1 DebuggableAttribute属性 16.2 调试器 16.3 托管调试器（MDbg） 16.3.1 MDbg实例 16.3.2 MDbg命令 16.4 WinDbg 16.5 Son of Strike （SOS） 16.5.1 SOS实例——第1部分 16.5.2 SOS实例——第2部分 16.6 转储文件 16.7 内存管理 16.7.1 对象图 16.7.2 代 16.7.3 终结进程 16.7.4 可靠性和性能监视器 16.8 线程 16.9 异常 16.10 符号 16.10.1 Symsrv符号服务器 16.10.2 应用程序符号 16.11 内存管理预览 第V部分 高级特性 第17章 内存管理 17.1 非托管资源 17.1.1 垃圾回收概述 17.1.2 GC风格 17.2 终结器 17.3 IDisposable.Dispose 17.3.1 Disposable模式 17.3.2 Disposable模式的考虑因素 17.4 弱引用 17.4.1 引用内部机理 17.4.2 WeakReference类 17.4.3 可靠的代码 17.4.4 管理非托管资源 17.5 GC类 17.6 不安全的代码预览 第18章 不安全的代码 18.1 unsafe关键字 18.1.1 指针 18.1.2 指针参数和针返回值 18.1.3 P/Invoke 18.2 小结 索引

章节摘录

插图：第Ⅰ部分 核心语言第1章 Visual C#编程简介Visual Studio 2008在开发期间称为Orcas，它是Visual Studio 2005的后继者。
最新版Visual Studi0与Visual C#2008、NET Framework 3．5和ASPNET 3．5一起发行。
Microsoft继续推动着Visual Studio从单一的综合性开发工具向软件生命周期解决方案转变。
visual Stuio包括的集成开发环境（IDE）、组件（测试工具和代码分析等）和工具，都是软件设计、开发、测试、质量保证和调试不可或缺的。
Anders Hejlsberg是Microsoft的技术院士（Technical Fellow）兼C#语言的主要设计师，他曾在很多场合谈到Visual C#2008。
他强调指出，语言集成查询（Language Integrated Query，LINQ）及相关的增强功能是Visual C群2008主要的新特色。
LrNQ是集成到C#语言中的一个面向对象的统一查询模型。
C#的诸如lambda表达式、扩展方法、表达式目录树、隐式类型的局部变量和对象及其他新特征，单独使用时都很有用，而且还扩展了C#语言对LINQ的支持。
这些改变以及其他方面的改变，确保了C#是NET的首要开发语言。
由于集成了LINQ语言，Visual C#2008改变了开发者与数据之间的关系。
LINQ是一种使用独立于数据源的查询语言来访问特定数据的解决方案。
LINQ也是面向对象的，而且是可扩展的。
LINQ在一定程度上使C≠}更接近于函数式编程，它使开发人员的关注焦点从数据（状态）的具体细节转移到信息（对象）行为，并且它还为数据查询提供了一个统一的模型，不再依赖于某种特定语言或技术的难以预测的变化。
通过LINQ，可以使用相同的统一查询模型访问SQL（Structured Query Language，结构化查询语言）数据库和XML（Extensible Markup Language，可扩展标记语言）文件，甚至还可以用它来访问数组。

编辑推荐

《Visual C#2008核心编程》特点为：C#专家经典力作全新出击，全面、深入地剖析C#2008新特性，内容全面，实例丰富，操作性强。
《Visual C#2008核心编程》全面深入地阐述了掌握Visual C撑2008的核心语言功能所需的理论知识、最佳实践和代码。
《Visual C#2008核心编程》内容与最新的Microsoft．NETFramework 3．5兼容，书中详细论述了LINQ，并阐述了Visual C#语言的复杂特征及其应用。
《Visual C#2008核心编程》围绕软件生命周期详细剖析TC#语言，是掌握Visual C#2008的必备指南。
《Visual C#2008核心编程》要重主题：比较使用类、类型和数组的语法与方法通过泛型和枚举器来增强代码效率和编码技巧了解何时和如何实现异常处理使用LINQ提供程序直接从C撑操纵和访问数据了解MSIL代码——改善编写托管代码的方法使用Microsoft Visual Studio IDE进行调试——或者使用高级调试器扩展在晚期绑定、程序集检查和自生成代码中使用元数据和反射掌握垃圾回收、内存管理和不安全代码的基本原理并运用最佳实践方法

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