

## <<框架设计>>

### 图书基本信息

书名：<<框架设计>>

13位ISBN编号：9787564124144

10位ISBN编号：7564124148

出版时间：2010-10

出版时间：东南大学出版社

作者：瑞奇特

页数：873

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>

## <<框架设计>>

### 内容概要

深度挖掘并掌握公共语言运行时、C#和.NET开发的错综复杂之处。

作为编程专家和微软.NET团队的长期顾问，Jeffrey Richter将带领你获得务实的见解来创建健壮、可靠且反应迅速的应用程序与组件。

更新.NET Framework 4.0及多核编程部分之后，这本完美的经典指南可帮助开发者建立任何类型的应用程序，包括MicrosoftSilverlight、ASP.NET、Windows、Presentation Foundation、Web services和控制台应用。

从本书理解以下内容： 建立、部署基于版本的应用程序、组件和共享程序集； 在理解原语、数值和引用类型行为的基础上来定义并更高效地使用它们； 使用泛型和接口定义可重用算法； 有效地使用特殊的CLR类型——委托、枚举、自定义属性、数组、字符串； 理解垃圾收集器如何管理资源； 使用线程池、任务、取消模式、定时器和异步I/O操作来设计灵活、可靠、可扩展的解决方案； 使用异常处理以协助状态管理； 使用CLR托管、AppDomain、程序集加载、反射和C#动态类型来构建动态可扩展的应用程序。

## <<框架设计>>

### 作者简介

Jeffrey Richter是Wintellect的共同创办人之一([www.wintellect.com](http://www.wintellect.com))，这是一家培训、咨询和调试相关的企业，专门致力于帮助其他公司更快构建更好的软件。

他于1999年10月开始为微软的.NET框架团队提供咨询，并已促成了微软的数款产品。

除了本书之前的版本，他还写过一些其他流行的主题，包括《Windows viaC/C++》和《Advanced Windows》。

## <<框架设计>>

### 书籍目录

ForewardIntroductionPart CLR Basics 1 The CLR's Execution Model Compiling Source Code into Managed Modules Combining Managed Modules into Assemblies Loading the Common Language Runtime Executing Your Assembly's Code IL and Verification Unsafe Code The Native Code Generator Tool: NGen.exe The Framework Class Library The Common Type System The Common Language Specification Interoperability with Unmanaged Code2 Building, Packaging, Deploying, and Administering Applications and Types3 Shared Assemblies and Strongly Named AssembliesPart Designing Types4 Type Fundamentals5 Primitive, Reference, and Value Types6 Type and Member Basics7 Constants and Fields8 Methods9 Parameters10 Properties11 Events12 Generics13 InterfacesPart Essential Types14 Chars, Strings, and Working with Text15 Enumerated Types and Bit Flags16 Arrays17 Delegates18 Custom Attributes19 Nullable Value TypesPart Core Facilities20 Exceptions and State Management21 Automatic Memory Management (Garbage Collection)22 CLR Hosting and AppDomains23 Assembly Loading and Reflection24 Runtime SerializationPart Threading25 Thread Basics26 Compute-Bound Asynchronous Operations27 I/O-Bound Asynchronous Operations28 Primitive Thread Synchronization Constructs29 Hybrid Thread Synchronization ConstructsIndex

## &lt;&lt;框架设计&gt;&gt;

## 章节摘录

插图：In fact, at runtime, the CLR has no idea which programming language the developer used for the source code. This means that you should choose whatever programming language allows you to express your intentions most easily. You can develop your code in any programming language you desire as long as the compiler you use to compile your code targets the CLR. So, if what I say is true, what is the advantage of using one programming language over another? Well, I think of compilers as syntax checkers and “correct code” analyzers. They examine your source code, ensure that whatever you've written makes some sense, and then output code that describes your intention. Different programming languages allow you to develop using different syntax. Don't underestimate the value of this choice. For mathematical or financial applications, expressing your intentions by using APL syntax can save many days of development time when compared to expressing the same intention by using Perl syntax, for example.

## <<框架设计>>

### 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>