

<<你必须知道的.NET>>

图书基本信息

书名：<<你必须知道的.NET>>

13位ISBN编号：9787121141287

10位ISBN编号：7121141280

出版时间：2011-8

出版时间：电子工业

作者：王涛

页数：520

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>

<<你必须知道的.NET>>

内容概要

由王涛编著的《你必须知道的.NET》来自于微软MVP的最新技术心得和感悟，将技术问题以生动易懂的语言展开，层层深入，以例说理。

全书主要包括了.NET基础知识及其深度分析，以.NET

Framework和CLR研究为核心展开.NET本质论述，涵盖了.NET基本知识几乎所有的重点内容。

全书分为5个部分，第1部分讲述.NET与面向对象，从底层实现角度分析了.NET如何实现面向对象机制，进一步分析了面向对象设计原则；第2部分论述了.NET类型系统和CLR的内存管理机制，并对IL语言进行了相应介绍；第3部分论述.NET

Framework框架的方方面面，详细分析了.NET框架的所有重点、难点和疑点内容，对框架类库的全貌进行了必要的专题性探讨；第4部分重点介绍了.NET泛型和安全性的相关知识和本质解密；第5部分对.NET

3.0/3.5/4.0 新特性进行了详细的介绍和引导，对于快速入门.NET新特性提供了方便之门。

《你必须知道的.NET》适于对.NET有一定了解的技术学习者、软件工程师和系统架构师阅读，同时也有助于.NET初学者进行快速提高，可作为大中专院校和.NET技术培训机构的参考教材。

<<你必须知道的.NET>>

作者简介

王涛，网名anytao，软件架构师，机械工程硕士，连续三届Microsoft Visual C# MVP，博客园技术专家，著有《你必须知道的.NET》一书，专注于.NET底层架构和.NET平台企业级应用，长期投入于互联网产品开发、微软云计算平台、分布式系统和企业级系统架构的研究与实践。目前，投身于移动互联产业，和几个志同道合的兄弟在技术之路上狂奔。

作者对.NET基础架构和CLR底层运行机制有浓厚的研究兴趣和造诣，熟悉ASP.NET、Windows Azure、SQL Server、WCF、LINQ、Silverlight、IIS、XML、Windows Phone、Facebook相关技术，精通数据库应用系统和大型Web系统的开发流程、技术体系和架构设计，对面向对象、设计模式和软件架构有长期的研究与实践经验。

您可以有三种方式与作者交流：

个人网站：

<http://www.anytao.com/>

技术博客：

<http://anytao.cnblogs.com/>

微博：

<http://weibo.com/anytao>

<<你必须知道的.NET>>

书籍目录

第1部分 渊源——.NET与面向对象

第1章 OO大智慧

1.1 对象的旅行

1.1.1 引言

1.1.2 出生

1.1.3 旅程

1.1.4 插曲

1.1.5 消亡

1.1.6 结论

1.2 什么是继承

1.2.1 引言

1.2.2 基础为上

1.2.3 继承本质论

1.2.4 秘境追踪

1.2.5 规则制胜

1.2.6 结论

1.3 封装的秘密

1.3.1 引言

1.3.2 让ATM告诉你，什么是封装

1.3.3 秘密何处：字段、属性和方法

1.3.4 封装的意义

1.3.5 封装规则

1.3.6 结论

1.4 多态的艺术

1.4.1 引言

1.4.2 问题的抛出

1.4.3 最初的实现

1.4.4 多态，救命的稻草

1.4.5 随需而变的业务

1.4.6 多态的类型、本质和规则

1.4.7 结论

1.5 玩转接口

1.5.1 引言

1.5.2 什么是接口

1.5.3 .NET中的接口

1.5.4 面向接口的编程

1.5.5 接口之规则

1.5.6 结论

参考文献

第2章 OO大原则

2.1 OO原则综述

2.1.1 引言

2.1.2 讲述之前

2.1.3 原则综述

2.1.4 学习建议

<<你必须知道的.NET>>

2.1.5 结论

2.2 单一职责原则

2.2.1 引言

2.2.2 引经据典

2.2.3 应用反思

2.2.4 规则建议

2.2.5 结论

2.3 开放封闭原则

2.3.1 引言

2.3.2 引经据典

2.3.3 应用反思

2.3.4 规则建议

2.3.5 结论

2.4 依赖倒置原则

2.4.1 引言

2.4.2 引经据典

2.4.3 应用反思

2.4.4 规则建议

2.4.5 结论

2.5 接口隔离原则

2.5.1 引言

2.5.2 引经据典

2.5.3 应用反思

2.5.4 规则建议

2.5.5 结论

2.6 Liskov替换原则

2.6.1 引言

2.6.2 引经据典

2.6.3 应用反思

2.6.4 规则建议

2.6.5 结论

参考文献

第3章 OO之美

3.1 设计的分寸

3.1.1 引言

3.1.2 设计由何而来

3.1.3 从此重构

3.1.4 结论

3.2 依赖的哲学

3.2.1 引言

3.2.2 什么是依赖，什么是抽象

3.2.3 重新回到依赖倒置

3.2.4 解构控制反转 (IoC) 和依赖注入

3.2.4 (DI)

3.2.5 典型的设计模式

3.2.6 基于契约编程：SOA架构下的

3.2.6 依赖

<<你必须知道的.NET>>

- 3.2.7 对象创建的依赖
- 3.2.8 不规则总结
- 3.2.9 结论
- 3.3 模式的起点
 - 3.3.1 引言
 - 3.3.2 模式的起点
 - 3.3.3 模式的建议
 - 3.3.4 结论
- 13.5.5 结论
- 13.6 江湖一统：WPF、WCF、WF
 - 13.6.1 引言
 - 13.6.2 WPF
 - 13.6.3 WCF
 - 13.6.4 WF
 - 13.6.5 结论
- 参考文献
- 第14章 跟随.NET 4.0脚步
 - 14.1 .NET十年
 - 14.1.1 引言
 - 14.1.2 历史脚步
 - 14.1.3 未来之变
 - 14.1.4 结论
 - 14.2 .NET 4.0，第一眼
 - 14.2.1 引言
 - 14.2.2 第一眼
 - 14.2.3 结论
 - 14.3 动态变革：dynamic
 - 14.3.1 引言
 - 14.3.2 初探
 - 14.3.3 本质：DLR
 - 14.3.4 PK解惑
 - 14.3.5 应用：动态编程
 - 14.3.6 结论
 - 14.4 趋势必行，并行计算
 - 14.4.1 引言
 - 14.4.2 拥抱并行
 - 14.4.3 TPL
 - 14.4.4 PLINQ
 - 14.4.5 并行补遗
 - 14.4.6 结论
 - 14.5 命名参数和可选参数
 - 14.5.1 引言
 - 14.5.2 一览究竟
 - 14.5.3 简单应用
 - 14.5.4 结论
 - 14.6 协变与逆变
 - 14.6.1 引言

<<你必须知道的.NET>>

- 14.6.2 概念解析
- 14.6.3 深入
- 14.6.4 结论
- 14.7 Lazy <T> 点滴
 - 14.7.1 引言
 - 14.7.2 延迟加载
 - 14.7.3 Lazy <T> 登场
 - 14.7.4 Lazy <T> 本质
 - 14.7.5 结论
- 14.8 Tuple一二
 - 14.8.1 引言
 - 14.8.2 Tuple为何物
 - 14.8.3 Tuple Inside
 - 14.8.4 优略之间
 - 14.8.5 结论
- 参考文献
- 后记：我写的不是代码
- 编后记：遇见幸福
- 3.4 面向对象和基于对象
 - 3.4.1 引言
 - 3.4.2 基于对象
 - 3.4.3 二者的差别
 - 3.4.4 结论
- 3.5 也谈.NET闭包
 - 3.5.1 引言
 - 3.5.2 什么是闭包
 - 3.5.3 .NET也有闭包
 - 3.5.4 福利与问题
 - 3.5.5 结论
- 3.6 好代码和坏代码
 - 3.6.1 引言
 - 3.6.2 好代码、坏代码
 - 3.6.3 结论
- 参考文献
- 第2部分 本质——.NET深入浅出
- 第4章 一切从IL开始
 - 4.1 从Hello, world开始认识IL
 - 4.1.1 引言
 - 4.1.2 从Hello, world开始
 - 4.1.3 IL体验中心
 - 4.1.4 结论
 - 4.2 教你认识IL代码——从基础到工具
 - 4.2.1 引言
 - 4.2.2 使用工具
 - 4.2.3 为何而探索
 - 4.2.4 结论
 - 4.3 教你认识IL代码——IL语言基础

<<你必须知道的.NET>>

- 4.3.1 引言
 - 4.3.2 变量的声明
 - 4.3.3 基本类型
 - 4.3.4 基本运算
 - 4.3.5 数据加载与保存
 - 4.3.6 流程控制
 - 4.3.7 结论
 - 4.4 管窥元数据和IL
 - 4.4.1 引言
 - 4.4.2 初次接触
 - 4.4.3 继续深入
 - 4.4.4 元数据是什么
 - 4.4.5 IL是什么
 - 4.4.6 元数据和IL在JIT编译时
 - 4.4.7 结论
 - 4.5 经典指令解析之实例创建
 - 4.5.1 引言
 - 4.5.2 newobj和initobj
 - 4.5.3 ldstr
 - 4.5.4 newarr
 - 4.5.5 结论
 - 4.6 经典指令解析之方法调度
 - 4.6.1 引言
 - 4.6.2 方法调度简论：call、callvirt和calli
 - 4.6.3 直接调度
 - 4.6.4 间接调度
 - 4.6.5 动态调度
 - 4.6.6 结论
- 参考文献
- 第5章 品味类型
- 5.1 品味类型——从通用类型系统开始
 - 5.1.1 引言
 - 5.1.2 基本概念
 - 5.1.3 位置与关系
 - 5.1.4 通用规则
 - 5.1.5 结论
 - 5.2 品味类型——值类型与引用类型
 - 5.2.1 引言
 - 5.2.2 内存有理
 - 5.2.3 通用规则与比较
 - 5.2.4 对症下药——应用场合与注意事项
 - 5.2.5 再论类型判等
 - 5.2.6 再论类型转换
 - 5.2.7 以代码剖析
 - 5.2.8 结论

<<你必须知道的.NET>>

5.3 参数之惑——传递的艺术

5.3.1 引言

5.3.2 参数基础论

5.3.3 传递的基础

5.3.4 深入讨论，传递的艺术

5.3.5 结论

5.4 皆有可能——装箱与拆箱

5.4.1 引言

5.4.2 品读概念

5.4.3 原理分拆

5.4.4 还是性能

5.4.5 重在应用

5.4.6 结论

参考文献

第6章 内存天下

6.1 内存管理概要

6.1.1 引言

6.1.2 内存管理概观要论

6.1.3 结论

6.2 对象创建始末

6.2.1 引言

6.2.2 内存分配

6.2.3 结论

6.3 垃圾回收

6.3.1 引言

6.3.2 垃圾回收

6.3.3 非托管资源清理

6.3.4 结论

6.4 性能优化的多方探讨

6.4.1 引言

6.4.2 性能条款

6.4.3 结论

参考文献

第3部分 格局——.NET面面俱到

第7章 深入浅出——关键字的秘密

7.1 把new说透

7.1.1 引言

7.1.2 基本概念

7.1.3 深入浅出

7.1.4 结论

7.2 base和this

7.2.1 引言

7.2.2 基本概念

7.2.3 深入浅出

7.2.4 通用规则

7.2.5 结论

7.3 using的多重身份

<<你必须知道的.NET>>

- 7.3.1 引言
- 7.3.2 引入命名空间
- 7.3.3 创建别名
- 7.3.4 强制资源清理
- 7.3.5 结论
- 7.4 认识全面的null
 - 7.4.1 引言
 - 7.4.2 从什么是null开始
 - 7.4.3 Nullable<T> (可空类型)
 - 7.4.4 ??运算符
 - 7.4.5 Null Object模式
 - 7.4.6 结论
- 7.5 转换关键字
 - 7.5.1 引言
 - 7.5.2 自定义类型转换探讨
 - 7.5.3 本质分析
 - 7.5.4 结论
- 7.6 预处理指令关键字
 - 7.6.1 引言
 - 7.6.2 预处理指令简述
 - 7.6.3 #if、#else、#elif、#endif
 - 7.6.4 #define、#undef
 - 7.6.5 #warning、#error
 - 7.6.6 #line
 - 7.6.7 结论
- 7.7 非主流关键字
 - 7.7.1 引言
 - 7.7.2 checked/unchecked
 - 7.7.3 yield
 - 7.7.4 lock
 - 7.7.5 unsafe
 - 7.7.6 sealed
 - 7.7.7 结论

参考文献

第8章 巅峰对决——走出误区

- 8.1 什么才是不变：const和readonly
 - 8.1.1 引言
 - 8.1.2 从基础到本质
 - 8.1.3 比较，还是规则
 - 8.1.4 进一步的探讨
 - 8.1.5 结论
- 8.2 后来居上：class和struct
 - 8.2.1 引言
 - 8.2.2 基本概念
 - 8.2.3 相同点和不同点
 - 8.2.4 经典示例
 - 8.2.5 结论

<<你必须知道的.NET>>

- 8.3 历史纠葛：特性和属性
 - 8.3.1 引言
 - 8.3.2 概念引入
 - 8.3.3 通用规则
 - 8.3.4 特性的应用
 - 8.3.5 示例
 - 8.3.6 结论
- 8.4 面向抽象编程：接口和抽象类
 - 8.4.1 引言
 - 8.4.2 概念引入
 - 8.4.3 相同点和不同点
 - 8.4.4 经典示例
 - 8.4.5 他山之石
 - 8.4.6 结论
- 8.5 恩怨情仇：is和as
 - 8.5.1 引言
 - 8.5.2 概念引入
 - 8.5.3 原理与示例说明
 - 8.5.4 结论
- 8.6 貌合神离：覆写和重载
 - 8.6.1 引言
 - 8.6.2 认识覆写和重载
 - 8.6.3 在多态中的应用
 - 8.6.4 比较，还是规则
 - 8.6.5 进一步的探讨
 - 8.6.6 结论
- 8.7 有深有浅的克隆：浅拷贝和深拷贝
 - 8.7.1 引言
 - 8.7.2 从对象克隆说起
 - 8.7.3 浅拷贝和深拷贝的实现
 - 8.7.4 结论
- 8.8 动静之间：静态和非静态
 - 8.8.1 引言
 - 8.8.2 一言蔽之
 - 8.8.3 分而治之
 - 8.8.4 结论
- 8.9 集合通论
 - 8.9.1 引言
 - 8.9.2 中心思想——纵论集合
 - 8.9.3 各分秋色——.NET集合类大观
 - 8.9.4 自我成全——实现自定义集合
 - 8.9.5 结论

参考文献

第9章 本来面目——框架诠释

- 9.1 万物归宗：System.Object
 - 9.1.1 引言
 - 9.1.2 初识

<<你必须知道的.NET>>

- 9.1.3 分解
- 9.1.4 插曲：消失的成员
- 9.1.5 意义
- 9.1.6 结论
- 9.2 规则而定：对象判等
 - 9.2.1 引言
 - 9.2.2 本质分析
 - 9.2.3 覆写Equals方法
 - 9.2.4 与GetHashCode方法同步
 - 9.2.5 规则
 - 9.2.6 结论
- 9.3 疑而不惑：interface “继承” 争议
 - 9.3.1 引言
 - 9.3.2 从面向对象寻找答案
 - 9.3.3 以IL探求究竟
 - 9.3.4 System.Object真是“万物之宗”吗
 - 9.3.5 接口的继承争议
 - 9.3.6 结论
- 9.4 给力细节：深入类型构造器
 - 9.4.1 引言：一个故事
 - 9.4.2 认识对象构造器和类型构造器
 - 9.4.3 深入执行过程
 - 9.4.4 回归故事
 - 9.4.5 结论
- 9.5 如此特殊：大话String
 - 9.5.1 引言
 - 9.5.2 问题迷局
 - 9.5.3 什么是string
 - 9.5.4 字符串创建
 - 9.5.5 字符串恒定性
 - 9.5.6 字符串驻留 (String Interning)
 - 9.5.7 字符串操作典籍
 - 9.5.8 补充的礼物：StringBuilder
 - 9.5.9 结论
- 9.6 简易不简单：认识枚举
 - 9.6.1 引言
 - 9.6.2 枚举类型解析
 - 9.6.3 枚举种种
 - 9.6.4 位枚举
 - 9.6.5 规则与意义
 - 9.6.6 结论
- 9.7 一脉相承：委托、匿名方法和Lambda
 - 9.7 表达式
 - 9.7.1 引言
 - 9.7.2 解密委托
 - 9.7.3 委托和事件

<<你必须知道的.NET>>

- 9.7.4 匿名方法
- 9.7.5 Lambda表达式
- 9.7.6 规则
- 9.7.7 结论

9.8 Name这事儿

- 9.8.1 引言
- 9.8.2 畅聊Name
- 9.8.3 回到问题
- 9.8.4 结论

9.9 直面异常

- 9.9.1 引言
- 9.9.2 为何而抛
- 9.9.3 从try/catch/finally说起：解析异常
- 9.9.3 机制
- 9.9.4 .NET系统异常类
- 9.9.5 定义自己的异常类
- 9.9.6 异常法则
- 9.9.7 结论

参考文献

第10章 格局之选——命名空间剖析

10.1 基础——.NET框架概览

- 10.1.1 引言
- 10.1.2 框架概览
- 10.1.3 历史变迁
- 10.1.4 结论

10.2 布局——框架类库研究

- 10.2.1 引言
- 10.2.2 为什么了解
- 10.2.3 框架类库的格局
- 10.2.4 一点补充
- 10.2.5 结论

10.3 根基——System命名空间

- 10.3.1 引言
- 10.3.2 从基础类型说起
- 10.3.3 基本服务
- 10.3.4 结论

10.4 核心——System次级命名空间

- 10.4.1 引言
- 10.4.2 System.IO
- 10.4.3 System.Diagnostics
- 10.4.4 System.Runtime.Serialization和
- 10.4.4 System.Xml.Serialization
- 10.4.5 结论

参考文献

第4部分 拾遗——.NET也有春天

第11章 接触泛型

11.1 追溯泛型

<<你必须知道的.NET>>

- 11.1.1 引言
- 11.1.2 推进思维, 为什么泛型
- 11.1.3 解析泛型——运行时本质
- 11.1.4 结论
- 11.2 了解泛型
 - 11.2.1 引言
 - 11.2.2 领略泛型——基础概要
 - 11.2.3 典型.NET泛型类
 - 11.2.4 基础规则
 - 11.2.5 结论
- 11.3 深入泛型
 - 11.3.1 引言
 - 11.3.2 泛型方法
 - 11.3.3 泛型接口
 - 11.3.4 泛型委托
 - 11.3.5 结论
- 11.4 实践泛型
 - 11.4.1 引言
 - 11.4.2 最佳实践
 - 11.4.3 结论
- 参考文献
- 第12章 如此安全性
 - 12.1 怎么样才算是安全
 - 12.1.1 引言
 - 12.1.2 怎么样才算安全
 - 12.1.3 .NET安全模型
 - 12.1.4 结论
 - 12.2 代码访问安全
 - 12.2.1 引言
 - 12.2.2 证据 (Evidence)
 - 12.2.3 权限 (Permission) 和权限集
 - 12.2.4 代码组 (Code Group)
 - 12.2.5 安全策略 (Security Policy)
 - 12.2.6 规则总结
 - 12.2.7 结论
 - 12.3 基于角色的安全
 - 12.3.1 引言
 - 12.3.2 Principal (主体)
 - 12.3.3 Identity (标识)
 - 12.3.4 PrincipalPermission
 - 12.3.5 应用示例
 - 12.3.6 结论
- 参考文献
- 第5部分 未来——.NET技术展望
- 第13章 走向.NET 3.0/3.5变革
 - 13.1 品读新特性
 - 13.1.1 引言

<<你必须知道的.NET>>

- 13.1.2 .NET新纪元
- 13.1.3 程序语言新特性
- 13.1.4 WPF、WCF、WF
- 13.1.5 Visual Studio 2008体验
- 13.1.6 其他
- 13.1.7 结论
- 13.2 赏析C# 3.0
 - 13.2.1 引言
 - 13.2.2 对象初始化器
(Object Initializers)
 - 13.2.3 集合初始化器
(Collection Initializers)
 - 13.2.4 自动属性
(Automatic Properties)
 - 13.2.5 隐式类型变量 (Implicitly Typed
Local Variables) 和 隐式类型数组
(Implicitly Typed Array)
 - 13.2.6 匿名类型 (Anonymous Type)
 - 13.2.7 扩展方法
(Extension Methods)
 - 13.2.8 查询表达式
(Query Expressions)
 - 13.2.9 结论
- 13.3 LINQ体验
 - 13.3.1 引言
 - 13.3.2 LINQ概览
 - 13.3.3 查询操作符
 - 13.3.4 LINQ to XML示例
 - 13.3.5 规则
 - 13.3.6 结论
- 13.4 LINQ江湖
 - 13.4.1 引言
 - 13.4.2 演义
 - 13.4.3 基于LINQ的零代码数据访问
层实现
 - 13.4.4 LINQ to Provider
 - 13.4.5 结论
- 13.5 抢鲜Visual Studio 2008
 - 13.5.1 引言
 - 13.5.2 Visual Studio 2008概览
 - 13.5.3 新特性简介
 - 13.5.4 开发示例

<<你必须知道的.NET>>

编辑推荐

由王涛编著的《你必须知道的.NET》以“你必须知道”而自诩，唯有意图达到以实例为基点，以归纳为方法的技术论述特点：对于技术的论述和分析，力求做到深入浅出、娓娓道来；对于晦涩艰深的问题以故事性的分析来引导；对于典型的问题以对比的角度来揭密；对于知识性的内容以归纳总结形成纲要。

作者对每个技术要点的论述，均结合浅显易懂的实例来展开，将复杂的技术问题化解在循序渐进的思考中。

让你的“悟”道，快乐而轻松。

全书主要包括了.NET基础知识及其深度分析，以.NET Framework和CLR研究为核心展开.NET本质论述，涵盖了.NET基本知识几乎所有的重点内容。

<<你必须知道的.NET>>

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>