第一图书网, tushu007.com

<<.NET 2.0模式开发实战>>

图书基本信息

书名：<<.NET 2.0模式开发实战>>

13位ISBN编号：9787115153852

10位ISBN编号：711515385X

出版时间：2007-5

出版时间：人民邮电出版社

作者：戈洛斯

页数：305

译者：张凯峰

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

内容概要

《.NET2.0模式开发实战》阐述了应用于.NET 2.0框架的设计模式，重点以C#语言来演示应用各种模式。
书中的主要内容包括面向对象编程的实质、模块化和异常、测试驱动开发、基本设计模式、应用于架构策略的设计模式、序列化和持久化等，同时针对测试、模式和重构，阐述了相关联的面向对象编程。
《.NET2.0模式开发实战》适合高等学校计算机相关专业师生，以及从事.NET程序设计的程序员阅读。

作者简介

Christian Gross是一位经验丰富、多才多艺的软件顾问，客户包括微软、戴姆勒－奔驰等大型公司。
他精通各种微软客户端/服务器技术。
除本书外，他还撰写了畅销书Ajax Patterns and Best Practices，并与人合作撰写了许多ASP、SQL Server、Windows编程方面的技术图书。

书籍目录

第1章　面向对象编程的实质1.1　理解类型1.2　模块化与可重用代码1.3　利用作用域控制访问1.4　理解继承1.4.1　class与struct在继承上的区别1.4.2　简单的继承1.4.3　利用虚函数继承1.5　软件工程中的继承1.6　编写泛型代码1.6.1　泛型的实例1.6.2　约束1.6.3　一些思考1.7　组合1.8　异步代码1.9　最后的思考第2章　日志、错误与测试驱动开发2.1　日志管理2.1.1　简单的log4net例子2.1.2　利用ILog接口生成消息2.1.3　管理配置文件2.1.4　建立一个现实的配置2.1.5　实现ToString2.2　实现异常处理2.2.1　错误与异常分类2.2.2　实现错误策略2.2.3　实现异常策略2.3　使用NUnit来做测试驱动开发2.3.1　理解测试驱动开发2.3.2　在应用程序中使用NUnit2.3.3　运行NUnit测试2.3.4　更多详细的NUnit测试2.4　最后的思考第3章　定义根基3.1　定义应用程序的基础3.1.1　定义意图3.1.2　测试驱动开发3.2　实现Bridge模式3.2.1　使用接口时留有选择余地3.2.2　过犹不及3.2.3　关于.NET1.x3.3　Bridge模式实现变种3.3.1　实现应用程序逻辑3.3.2　控制器接口3.3.3　实现默认的基类3.3.4　接口和类的设计决策3.4　使用Factory模式实例化类型3.4.1　需要帮助类型3.4.2　实现插件架构3.4.3　根据计划创建对象3.4.4　克隆对象3.5　最后的思考第4章　应用程序的架构4.1　使应用程序正常运行4.1.1　可扩展性和可维护性4.1.2　使用黑盒4.2　PipesandFilters模式4.2.1　例子：从Amazon.com购买电视机4.2.2　电视机选择系统的架构4.2.3　电视机选择系统的实现4.2.4　关于PipesandFilters模式的几点思考4.3　Client-Dispatcher-Server模式4.3.1　定义Client-Dispatcher-Server模式的架构4.3.2　静态分配器架构4.3.3　动态分配器架构4.3.4　架构Client-Dispatcher-Server模式4.3.5　实现程序集目录解析器4.3.6　实现Web服务解析器4.4　Micro-Kernel模式4.4.1　微内核的架构4.4.2　隐藏微内核的细节4.4.3　设计微内核4.4.4　微内核实现细节4.4.5　构建简单的银行应用程序4.4.6　关于Micro-Kernel模式的思考4.5　最后的思考第5章　实现成组的组件5.1　两个传统的面向对象错误5.1.1　属性和烤箱温度控制5.1.2　继承和基类的脆弱性问题5.2　示例应用：翻译程序5.2.1　快速编写一个简陋的应用程序5.2.2　重构代码5.2.3　重构并且实现Bridge和Factory5.2.4　实现Mediator模式5.2.5　实现Template模式5.2.6　实现Adapter模式5.2.7　关于翻译程序的最后思考5.3　为应用程序添加多语言支持5.3.1　想想看：Decorator还是Composite5.3.2　实现ChainofResponsibility模式5.3.3　实现Command模式5.3.4　实现Composite模式5.3.5　实现Decorator模式5.3.6　实现State模式5.3.7　实现Strategy模式5.3.8　实现翻译语言的动态选择5.4　最后的思考第6章　编写算法6.1　不做修改的功能模仿6.1.1　实现Proxy模式6.1.2　使用仿函数增强类型6.1.3　为集合创建泛型仿函数架构6.2　构建电影票应用程序6.2.1　从基础开始6.2.2　计算票的销售额6.2.3　读取销售额数据6.2.4　使用null的问题6.2.5　更为简单的买票方法：使用Facade模式6.3　使用多态管理扩展6.3.1　实现StaticExtension模式6.3.2　实现DynamicExtension模式6.3.3　扩展、类型转换以及它们的意义6.4　使用Iterator模式遍历数据6.4.1　使用C#2.0实现Iterator模式6.4.2　在迭代器中使用仿函数6.5　最后的思考第7章　高效代码7.1　不可变类是高效的类7.1.1　为什么不可变类具有一致性7.1.2　为什么不可变类是可伸缩的7.1.3　一些使用不可变类的经验7.2　在Flyweight模式中使用不可变类7.2.1　Flyweight模式的例子7.2.2　通用的Flyweight架构7.2.3　使用通用Flyweight架构7.2.4　使用Flyweight的实现7.3　对象池原理7.3.1　对象池和COM+7.3.2　对象池理论7.3.3　在.NET中实现ObjectPool模式7.4　多线程应用程序7.4.1　简单的线程例子7.4.2　实现单例7.4.3　使用生产者-消费者技术管理多线程问题7.5　最后的思考第8章　数据持久化8.1　.NET中的序列化8.1.1　.NET中的二进制对象序列化8.1.2　.NET中的XML对象序列化8.1.3　序列化的问题8.2　调整并完善Serializer模式8.2.1　问外部状态：Visitor模式8.2.2　访问内部状态：Memento模式8.3　使用NHibernate进行对象/关系数据映射8.3.1　简单的NHibernate示例8.3.2　映射一对多关系8.3.3　其他类型的关联8.3.4　使用HQL8.4　最后的思考第9章　通过重构实现模式9.1　测试驱动开发与重构9.1.1　编写第一行代码9.1.2　在第一部分代码后9.1.3　重构的种类9.2　类，方法--一切都太大了9.2.1　重构Stream类9.2.2　重构Stream类的问题9.2.3　重构类而不是基类型9.3　我不理解代码9.3.1　处理未知代码9.3.2　跟踪代码9.3.3　中断代码9.4　代码似同实异9.4.1　为什么复制和粘贴代码有效9.4.2　使用Template方法重构重复的代码9.4.3　可以接受的重复9.5　时不我待9.6　我希望移除代码9.7　最后的思考

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