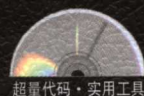


软·件·工·程·师·典·藏



ASP.NET 技术方案宝典

■ 明日科技 张跃廷 王小科 张宏宇 编著



人民邮电出版社
POSTS & TELECOM PRESS

ASP.NET

技术方案宝典

■ 明日科技 张跃廷 王小科 张宏宇 编著



人民邮电出版社
北京

图书在版编目 (CIP) 数据

ASP.NET 技术方案宝典 / 张跃廷, 王小科, 张宏宇编著.
北京: 人民邮电出版社, 2008.2
(软件工程师典藏)
ISBN 978-7-115-17305-8

I. A… II. ①张…②王…③张… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 189957 号

内 容 提 要

本书从 ASP.NET 进行 Web 应用程序开发必须掌握的核心技术入手, 通过实用方案深入介绍各种核心技术在实际开发中的应用。全书分为 9 章, 分别是设计模式与架构选择方案、页面设计方案、数据库操作技术方案、数据检索方案、决策分析方案、水晶报表打印方案、网络通信方案、网站安全与管理方案、网站编译与发布方案等。通过本书, 读者不但可以学习相关技术的各种核心应用, 更能触类旁通、学以致用, 掌握 ASP.NET 进行应用开发的精髓。

本书附有配套光盘。光盘提供了书中所有方案实例的全部源代码, 所有实例都经过精心调试, 在 Windows XP/2003 下全部通过, 保证能够正常运行。

本书内容详实、技术突出, 具有非常强的实用性。适合各级软件开发人员学习使用, 也可供大中专院校师生学习参考。

软件工程师典藏

ASP.NET 技术方案宝典

- ◆ 编 著 明日科技 张跃廷 王小科 张宏宇
责任编辑 屈艳莲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 43.5
字数: 1 183 千字 2008 年 2 月第 1 版
印数: 1-4 000 册 2008 年 2 月河北第 1 次印刷

ISBN 978-7-115-17305-8/TP

定价: 85.00 元 (附光盘)

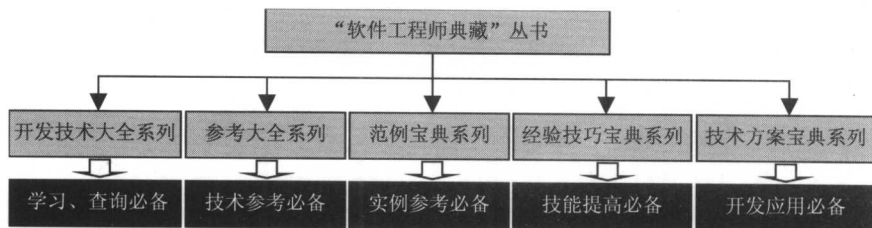
读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

丛书序

随着信息技术的不断发展,软件业已经成为我国经济发展的战略性产业,社会上急需大量的IT人才以满足发展的需求,其中熟练进行软件开发的程序员更得到企业的青睐。但软件需求千变万化,代码实现也是千差万别,要想成为一名合格的开发人员,需要较长时间的学习和历练。缺乏实际开发经验仍是横亘在众多编程人员面前的“鸿沟”。如果能将实际开发经验和技能传授给他们,无疑会对软件开发人员的成才起到催化剂的作用。所以,为软件开发人员量身打造的、专业全面的“软件工程师典藏”丛书就应运而生了。

一、丛书框架



● 技术大全

介绍知识系统全面、易学易查、实例丰富,是软件开发人员学习和备查的理想参考书。

● 参考大全

深入解析开发中的技术细节,对每一项专题技术都给出了深入的解析和实用的例子。是开发人员不可缺少的技术参考书。

● 范例宝典

精选数百个常用的编程范例,不但解决开发中的常见问题,更可以帮助读者提升编程水平。

● 经验技巧宝典

收录了1000多条经验技巧,这些经验技巧都是开发团队在长期的实践中不断积累的。提供给读者随查随用,帮助读者少走弯路、拓展编程思路。

● 技术方案宝典

剖析实际应用方案、剖析核心技术在开发中的各种实际应用,是软件开发人员必备的技术方案参考。

二、丛书特点

1. 给出语言学习的整体解决方案

本套丛书覆盖了大多数主流语言和工具,可以满足不同读者学习之需。也非常适合专业人士参考。

2. 积淀一线编程高手的经验技巧

丛书作者均具有多年的实际开发经验,他们从程序开发的不同角度为开发人员提供全面、



实用的开发理念，从而保证内容不再是演示性的实例，最大程度地贴近了实战。

3. 完善的答疑服务

丛书作者将通过网站、论坛、热线电话等方式为读者提供疑难解答，随时解决读者遇到的各种技术问题。

感谢参与本丛书写作的每位作者，有他们的不懈努力和辛勤工作，本丛书才能顺利推出。愿本丛书能成为广大软件开发人员的良师益友。

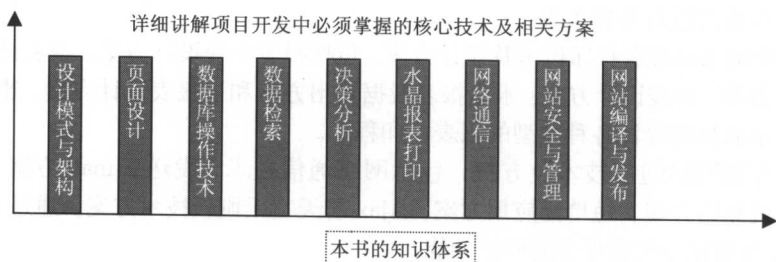
本书编委会
2007年11月



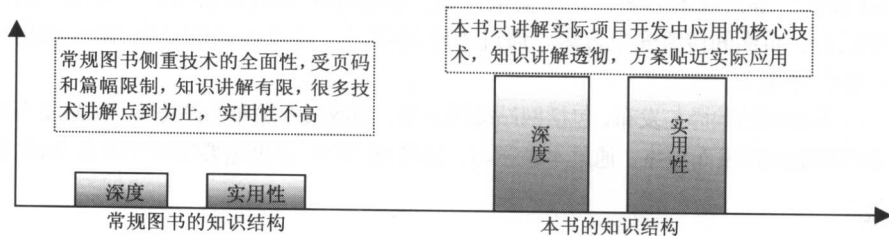
前言

ASP.NET 是 Microsoft 公司推出的 Web 开发平台,也是目前最先进、特征最丰富、功能最强大的 Web 开发平台。ASP.NET 具有方便性、灵活性、性能优、生产效率高、安全性高、完整性强等特性,是目前主流的网络编程环境之一。

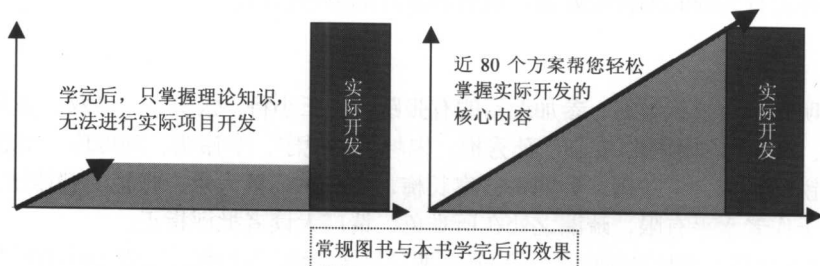
目前,虽然介绍 ASP.NET 的图书如满天星斗,数不胜数,但真正介绍 ASP.NET 核心技术的书籍却不多,尤其是针对实际项目开发提出具体解决方法的书籍。本书从实际开发必须掌握的核心技术出发,将实际开发中与该技术相关的应用整理成方案,通过各种实际开发方案讲透各种技术的具体应用,真正授人以渔,满足实际开发者的需求。读者学完本书后,可以顺利进行实际项目的开发。下图是本书的知识体系。



下面列出了常规图书和本套图书在知识结构和学习效果上的比较。



常规图书与本书知识结构的比较



本书内容

本书主要以 Web 开发中必须掌握的核心技术入手,深入探讨实际开发中必须掌握的技术以及技术的各种应用方案,使读者不但可以了解知识的来龙去脉,更能学会如何利用学到的知识进行实际开发。本书共分 9 章,章节内容如下。

第 1 章:介绍软件设计模式、软件架构设计技术方案,主要包括 C# 主要设计模式、二层架构分



析与使用、三层架构分析与使用、MVC 架构分析与使用及架构的选择方案等。学好本章，可以选择合适的设计模式和架构的搭建，从而使程序设计更合理，代码更清晰，更容易维护，团队合作更便捷。

第2章：介绍界面设计技术及相关方案，包括框架布局设计方案、用户控件设计方案、导航条设计方案、广告栏设计方案、多媒体设计方案、母版页设计方案、主题设计方案。通过本章学习，读者可以透彻掌握界面的相关技能，设计界面友好的 Web 应用程序。

第3章：介绍数据库操作技术及方案，包括数据库开发技术、数据库访问方案、ADO.NET 操作数据库方案、文件上传存储方案、数据库维护方案和数据库转换方案。通过本章的学习，读者可以轻松掌握对数据库的各种操作。

第4章：介绍数据查询、显示技术及方案。包括数据查询技术、简单查询方案、中级查询方案、高级查询方案和数据显示方案。通过本章学习，读者将非常容易地处理数据库查询显示问题，并可设计较复杂的查询。

第5章：介绍决策分析技术及方案。包括决策分析技术、交叉表决策分析方案和透视表及透视图决策分析方案、统计图决策分析方案、实时动态决策分析方案。通过本章学习，读者可以轻松对数据进行分析决策。

第6章：介绍水晶报表打印技术及设计方案。包括报表外观设计方案、报表排序分组方案、交叉报表设计方案、图表设计方案、水晶报表数据导出方案和子报表设计方案。通过本章学习，读者可以利用水晶报表设计各种类型的报表打印程序。

第7章：介绍网络通信技术及方案。包括网络通信技术、发送 E-mail 方案、接收 E-mail 方案、发送手机短信方案、短信猫应用方案、Ajax 聊天室无刷新技术方案。通过本章学习，读者可以为 Web 应用程序实现更多的网络功能。

第8章：介绍网站安全与管理方案，包括 Windows 身份验证方案、Forms 验证模式方案、Web.Config 加密与解密技术方案、密码安全方案、数据库中数据加密与解密方案、防止非法用户进入网站后台、页面间传值安全方案。通过本章的学习，读者在以后开发 Web 应用程序时，使 Web 应用程序更安全。

第9章：介绍网站编译与发布。包括网站编译方案、局域网内发布网站方案、网站发布到 Internet 方案和带数据库的网站发布方案。通过本章学习，读者可以轻松根据需要编译和发布 Web 应用程序。

读者对象

本书非常适合目前正从事项目开发实践的软件开发人员开发时参考。对于具备一定语言基础，对程序开发有兴趣的编程爱好者，也有非常好的参考价值。

技术支持

本书由明日科技组织编写，参加编写的有张跃廷、王小科、张宏宇、吕双、房大伟、苏宇、梁水、张言、梁冰、刘彬彬、安剑、孙秀梅、宋坤、刘锐宁、李伟明、刘玲玲、刘欣、梁晓岚、顾彦玲、黄锐、杨丽、王冬雪、孙明娇、寇长梅、张鹏斌、董大永、张艳、郭佳博、乔敏、赛奎春等。由于作者水平有限，疏漏之处在所难免，请广大读者批评指正。

如果读者在使用本书时遇到问题，可以访问明日科技网站，我们将通过明日科技网站为读者提供网上服务和支持。读者使用本书时遇到的问题，我们承诺在 1~6 个工作日给您提供及时回复。

服务网站：www.mingrisoft.com 服务信箱：mingrisoft@mingrisoft.com

服务电话：0431-84978981 84978982

明日科技
2007 年 11 月



目 录

第1章 设计模式与架构选择方案.....1

1.1 设计模式与架构.....2	
1.1.1 什么是设计模式.....2	
1.1.2 设计模式与架构的关系.....2	
1.1.3 设计模式基本要素.....2	
1.1.4 为什么使用设计模式.....3	
1.1.5 避免过度使用设计模式.....3	
1.1.6 什么是架构.....4	
1.1.7 为什么要使用架构.....4	
1.1.8 二层架构开发.....4	
1.1.9 三层架构开发.....5	
1.1.10 MVC 架构开发.....8	
1.2 主要设计模式.....11	
1.2.1 工厂模式.....11	
1.2.2 单例模式.....17	
1.2.3 建造者模式.....19	
1.2.4 原型模式.....22	
1.2.5 适配器 (Adapter) 模式.....25	
1.2.6 合成 (Composite) 模式.....27	
1.2.7 装饰 (Decorator) 模式.....30	
1.2.8 代理 (Proxy) 模式.....33	
1.2.9 策略模式 (Strategy Pattern)35	
1.3 ASP.NET 架构开发方案.....37	
1.3.1 二层架构开发设计与应用.....38	
1.3.2 三层架构开发设计与应用.....49	
1.3.3 MVC 架构设计与使用.....67	

第2章 页面设计方案.....73

2.1 页面设计技术概述.....74	
2.1.1 页面框架技术.....74	
2.1.2 用户控件技术.....76	
2.1.3 导航条技术.....77	
2.1.4 广告栏技术.....78	
2.1.5 多媒体技术.....78	
2.1.6 母版页技术.....81	

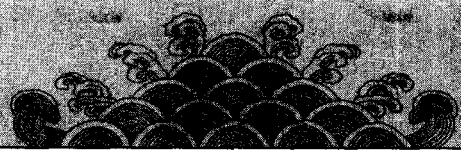


2.1.7 主题应用技术	85
2.2 框架布局设计方案	87
2.2.1 Frameset 框架设计方案	88
2.2.2 Iframe 框架设计方案	90
2.3 用户控件设计方案	92
2.3.1 普通用户控件设计方案	92
2.3.2 高级用户控件设计方案	100
2.4 导航条设计方案	103
2.4.1 树型导航条设计方案	103
2.4.2 水平导航条设计方案	110
2.4.3 下拉菜单导航条设计方案	114
2.5 广告栏设计方案	117
2.5.1 Banner 广告栏设计方案	117
2.5.2 浮动广告栏设计方案	119
2.6 多媒体设计方案	121
2.6.1 循环播放图片设计方案	121
2.6.2 音频嵌入设计方案	126
2.6.3 视频嵌入设计方案	127
2.7 母版页设计方案	129
2.7.1 简单母版页设计方案	129
2.7.2 嵌套母版页设计方案	133
2.7.3 访问复杂母版页的方案	138
2.7.4 母版页的高级应用方案	143
2.8 主题设计方案	148
2.8.1 构建主题的方案	148
2.8.2 简单主题的使用方案	154
2.8.3 主题的高级应用方案	157
2.8.4 网站主题的切换设计方案	159

第3章 数据库操作技术方案 163

3.1 数据库操作技术	164
3.1.1 SQL 数据库技术	164
3.1.2 文件上传技术	168
3.1.3 ADO.NET 技术	169
3.1.4 数据库维护技术	171
3.2 数据库访问方案	174
3.2.1 访问 Access 数据库	174
3.2.2 访问 SQL Server 2000 数据库	179
3.2.3 访问 SQL Server 2005 数据库	183
3.3 ADO.NET 操作数据库方案	184
3.3.1 单条记录操作	184
3.3.2 批量记录操作	190





3.3.3 通过存储过程操作数据	199
3.3.4 通过触发器操作数据	208
3.3.5 通过事务操作记录	219
3.4 文件上传存储方案	225
3.4.1 存储文件路径	225
3.4.2 以二进制形式存储	234
3.5 数据库维护方案	240
3.5.1 数据库的备份	240
3.5.2 数据库的恢复(还原)	246
3.5.3 数据库的附加与分离	250
3.6 数据库转换方案	255
3.6.1 将 Access 转换为其他数据库	255
3.6.2 数据库表转换其他数据库表	260
3.6.3 将 Excel 转换为 SQL Server 数据库	268
3.6.4 XML 文件转换技术	270

第4章 数据检索方案 273

4.1 数据查询技术	274
4.1.1 数据查询的环境和方法	274
4.1.2 SQL 查询技术	280
4.1.3 解决查询过程中的问题	296
4.1.4 对应不同的数据库系统	298
4.2 普通查询方案	300
4.2.1 单字段精确查询的实现	300
4.2.2 多字段精确查询的实现	305
4.2.3 简单模糊查询的实现	309
4.2.4 查询排序的实现	314
4.3 高级查询方案	316
4.3.1 利用视图进行查询	316
4.3.2 利用存储过程进行查询	319
4.4 查询结果显示方案	324
4.4.1 表格显示方式	324
4.4.2 主细节页显示方式	328
4.4.3 缩略图显示方式	334
4.4.4 分页显示方式	336
4.5 其他相关查询	341
4.5.1 字段显示设置	341
4.5.2 查询内容描红	344
4.5.3 在结果中查询	350

第5章 决策分析方案 355

5.1 决策分析技术	356
------------	-----



5.1.1	决策分析技术概述	356
5.1.2	交叉表分析技术	356
5.1.3	透视表分析技术	361
5.1.4	透视图分析技术	365
5.1.5	统计图分析技术	365
5.1.6	实时动态决策分析技术	369
5.2	交叉表决策分析方案	371
5.2.1	利用静态交叉表进行决策分析	371
5.2.2	利用动态交叉表进行决策分析	379
5.3	透视表及透视图决策分析方案	385
5.3.1	软件控制生成透视表方案	385
5.3.2	用户控制生成透视表方案	389
5.3.3	透视图决策分析方案	394
5.4	统计图决策分析方案	398
5.4.1	利用普通统计图进行决策分析	398
5.4.2	利用组合统计图进行决策分析	410
5.5	实时动态决策分析方案	426
5.5.1	实时刷新页面统计数据方案	426
5.5.2	实时局部刷新模式统计数据方案	429

第6章 水晶报表打印方案 **433**

6.1	水晶报表技术	434
6.1.1	水晶报表及其数据源概述	434
6.1.2	水晶报表外观设计技术	438
6.1.3	水晶报表排序、分组技术	441
6.1.4	交叉报表设计技术	443
6.1.5	带图表的水晶报表设计技术	444
6.1.6	水晶报表导出数据技术	447
6.1.7	子报表设计技术	451
6.2	报表数据源方案	452
6.2.1	Excel 数据源设计	452
6.2.2	Access 数据源设计	456
6.2.3	SQL Server 数据源设计	463
6.3	报表外观设计方案	469
6.3.1	格式化各类对象设计	469
6.3.2	特殊字段、图片和线设计方案	473
6.3.3	自定义格式化方案	476
6.4	排序、分组设计方案	479
6.4.1	使用查询来排序报表数据	480
6.4.2	报表排序设计	481
6.4.3	单级与多级分组设计	485
6.4.4	层次分组设计方案	488

6.5 交叉表分析设计方案	490
6.5.1 交叉表设计方案	490
6.5.2 多行多列交叉表设计方案	493
6.6 图表分析设计方案	495
6.6.1 使用图表专家设计图表方案	496
6.6.2 自定义图表配置方案	498
6.7 报表导出设计方案	505
6.7.1 CrystalReportviewer 控件方案	506
6.7.2 ReportDocument 对象方案	508
6.8 子报表设计方案	514
6.8.1 未链接子报表	514
6.8.2 链接子报表	516

第7章 网络通信方案 521

7.1 网络通信技术	522
7.1.1 Jmail 组件发送和接收 E-mail 技术	522
7.1.2 发送手机短信技术	527
7.1.3 Ajax 无刷新技术	529
7.1.4 SMTP 服务的安装与配置	533
7.2 发送 E-mail 方案	536
7.2.1 普通 E-mail 发送	536
7.2.2 群发 E-mail	541
7.3 接收 E-mail 方案	545
7.3.1 普通 E-mail 接收	545
7.3.2 接收带附件的 E-mail	553
7.4 发送手机短信方案	557
7.4.1 通过 Web Service 发送手机短信	557
7.4.2 通过短信猫发送手机短信	559
7.5 短信猫应用方案	563
7.5.1 短信采集销售数据	563
7.5.2 节目评比短信信息互动平台	565
7.6 Ajax 聊天室无刷新技术方案	568

第8章 网站安全与管理方案 579

8.1 网站安全技术	580
8.1.1 身份验证技术	580
8.1.2 Web.Config 加密与解密技术	580
8.1.3 密码安全技术	581
8.1.4 数据库中数据加密与解密技术	582
8.1.5 防止非法用户进入网站后台验证技术	588
8.1.6 页面间传值安全技术	588
8.2 Windows 身份验证方案	590





8.2.1	Windows 身份验证方案分析	590
8.2.2	基本身份验证方案	590
8.2.3	摘要身份验证方案	593
8.2.4	集成的 Windows 身份验证方案	594
8.3	Forms 验证模式方案	596
8.3.1	Forms 身份验证方案	596
8.3.2	普通登录方案	597
8.3.3	验证码技术登录方案	601
8.3.4	Login 登录控件登录方案	606
8.4	Web.Config 加密与解密技术方案	610
8.4.1	使用 SectionInformation 类解决方案	610
8.4.2	命令行工具 aspnet_regiis.exe 解决方案	612
8.5	密码安全方案	615
8.5.1	创建安全密码方案	615
8.5.2	密码加密方案	620
8.6	数据库中数据加密与解密方案	622
8.6.1	对称加密解决方案	622
8.6.2	非对称加密解决方案	628
8.6.3	对称解密方案	631
8.6.4	非对称解密方案	636
8.7	防止非法用户进入网站后台	638
8.7.1	通过 Session 判断登录用户方案	638
8.7.2	防止 SQL 注入式攻击登录方案	639
8.8	页面间传值安全方案	641
8.8.1	Querystring 使用方案	641
8.8.2	Session 使用方案	642
8.8.3	Server.Transfer 使用方案	643

第 9 章 网站编译与发布方案 647

9.1	网站编译方案	648
9.1.1	网站编译技术	648
9.1.2	网站编译方案	649
9.2	局域网内发布网站方案	654
9.2.1	使用同一 IP 地址、不同端口号架设多个 Web 网站	654
9.2.2	使用多个 IP 地址架设多个 Web 网站	656
9.2.3	使用 DNS 服务器发布带域名的网站	660
9.3	网站发布到 Internet 方案	666
9.3.1	Visual Studio 2005 发布网站到 Internet	666
9.3.2	使用 FTP 工具发布网站到 Internet	672
9.4	带数据库的网站发布方案	680
9.4.1	使用 Access 数据库的网站发布	680
9.4.2	使用 SQL Server 数据库的网站发布	681



第 1 章

设计模式与架构选择方案

- 设计模式与架构
- 主要设计模式
- ASP.NET 架构开发方案





通过学习本章可以使读者掌握 C# 主要设计模式的合理使用、各种架构设计方案的正确选择与使用,并且每一个技术方案都配以案例实施过程。

1.1 设计模式与架构

本节主要概述了设计模式、架构设计的概念,对设计模式与架构设计的关系进行区别,以及在相应的情况下合理地使用设计模式和设计架构。

1.1.1 什么是设计模式

在生活和工作中当中的各个方面,不断地重复一些事物和事情的方法,可以看作是一种设计模式,例如:

① 小王家的茶几和小李家的茶几很像,但小王家的茶几用门代替了抽屉。

② 张三家的花园和李四家的花园很像,但李四家的花园种的是菊花。

那么在程序设计中也会有同样的事情,例如,开发一个程序可以通过很多技巧来完成程序功能。程序设计模式没有一个统一的定义,是开发者在开发中不断积累、总结的一种可复制的方案,下面是该方面几位专家给出的理解。

- 设计模式是对程序设计人员经常遇到的设计问题的可再现的解决方案 (The Smalltalk Companion)。
- 设计模式建立了一系列描述如何完成软件开发领域中的特定任务的规则 (Pree 1995)。
- 设计模式更关注与复用可重复出现的结构设计方案,而框架注重于具体设计和实现 (Coplien and Schmidt 1995)。
- 设计模式提出了一个发生在特定设计环境中的可重复出现的设计问题,并提供解决方案 (Buschmann et al.1996)。
- 设计模式识别并确定类和实例层次上或组件层次上的抽象关系 (Gamma, Johnson, and Vlissides, 1993)。

总之,设计模式在设计者中是一种流行的思考设计问题的方法,是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了使代码具有可重用性、让代码更容易被他人理解和保证代码可靠性。设计模式使代码编制为工程化,设计模式是软件工程的基石。

1.1.2 设计模式与架构的关系

现在,使用面向对象软件系统一般划分为三大类:应用程序、工具箱和框架 (Framework)。应用程序是指开发工具,如 Visual Studio 2005、Visual Basic、delphi 等;工具箱是开发程序时使用的工具,如服务器控件、控件、组件等;而框架是构成一类特定软件可复用设计的一组相互协作的类库,如 .NET Framework。

框架通常定义了应用体系的整体结构、类和对象的关系等设计参数,以便于具体应用实现者能集中精力于应用本身的特定细节。框架主要记录软件应用中共同的设计决策,框架要求设计重复使用,因此框架设计中必然要使用设计模式。

另外,设计模式有助于对框架结构的理解,成熟的框架通常使用了多种设计模式,如果程序人员熟悉这些设计模式,毫无疑问,将迅速掌握框架的结构,一般开发者如果突然接触框架,会觉得特别难学、难掌握,那么先掌握设计模式,无疑是剖析框架的基础。

1.1.3 设计模式基本要素

设计模式使人们可以更加简单方便地复用成功的设计和体系结构,将已证实的技术表述成





设计模式也会使新加入的系统开发者更加容易理解其设计思路。设计模式的基本要素为模式名称、问题、解决方案和效果。

(1) 模式名称

一个助记名称，用来描述设计模式、解决方案和效果。设计模式允许在较高的抽象层次上进行设计。基于一个模式词汇表，开发团队之间可以讨论模式并在编写文档时使用它们。模式名称可以帮助我们思考，便于团队成员交流设计思想及设计结果。找到合适的模式名称也是设计模式编目工作的难点之一。

(2) 问题

问题主要描述在何时使用设计模式。它解释了设计问题和存在的问题的前因后果、特定的设计问题和怎样用对象表示算法等。通常情况下，模式必须满足的一系列先决条件是问题。

(3) 解决方案

解决方案描述了设计的组成成分、它们之间的相互关系及各自的职责和协作方式。因为模式就像一个模板，可应用于多种不同场合，所以解决方案并不描述一个特定具体的设计或实现，而是提供设计问题的抽象描述和怎样用一个具有一般意义的元素组合（类或对象组合）来解决这个问题。

(4) 效果

描述了模式应用的效果及使用模式权衡的问题。尽管描述设计决策时，并不是总提到模式效果，但它们对于评价设计选择和理解使用模式的代价及优势具有重要意义。软件效果大多关注对时间和空间的衡量，它们也表述了语言和实现问题。因为复用是面向对象设计的要素之一，所以模式效果包括它对系统灵活性、扩充性或可移植性的影响，显式地列出这些效果对理解和评价这些模式很有帮助。

1.1.4 为什么使用设计模式

开发人员不会对一个预先设计应用原则和模式。相反，这些原则和模式被应用在一次次的迭代中，力图使代码设计流畅。

为什么使用设计模式，笔者的体会如下。

- 设计模式是为了使设计适应变化。
- 设计模式是重构的工具。
- 设计一开始就要保持流畅、简单，并具有持续性。
- 不能过度使用设计模式。

使用设计模式的目的是为了适应未来的变化，变化之所以存在是因为一切的事物都具有不可预见性，如果具有可预见性，则不能称为变化。如何判断哪些需求可能变化，哪些需求可能不变，并且在最大程度上保持设计的流畅、简单，这些是工艺问题，而不是工程问题。既然是工艺问题，那么就只能给出原则，不能给出标准。使用设计模式的原则是：对未来极有可能发生变化的问题进行简单的修改、降低成本。

1.1.5 避免过度使用设计模式

易维护的程序首先要容易理解，这很重要。易理解的代码才利于维护。过度的使用设计模式会增加程序的复杂性，让程序不易理解，从而降低了程序的易维护性。

下面举一个实际开发中的例子，讲解过度使用设计模式的弊端。

Switch 语句曾经遭致诟病，许多重构的例子就是拿 Switch 做比较的，Switch 语句是高效的语句，可以写出很优雅、简单的代码。在特殊情况下，直接使用 Switch 语句比把它拆成若干个 Class 更流畅。例如，有一段五百多行的代码负责整个系统的调度，如果未来的变化仅仅是修改





这五百行代码而不会大量添加代码，那么把这五百多行代码集中在一个方法里面，比将它拆分成十个 Class 更加容易维护。

1.1.6 什么是架构

在本文中，“架构”与“软件架构”是相同的含义。软件架构（Software Architecture）是一系列相关的抽象模式，用于指导大型软件系统各个方面的设计。软件架构是一个系统的草图，软件架构描述的对象是直接构成系统的抽象组件，各个组件之间的连接则明确和相对细致地描述组件之间的通信。在实现阶段，这些抽象组件被细化为实际的组件，例如具体某个类或者对象。在面向对象领域中，组件之间的连接通常用接口来实现。

软件体系结构是构建计算机软件实践的基础，与建筑师设定建筑项目的设计原则和目标，作为绘图人员画图的基础一样，一个软件架构师或者系统架构师陈述软件构架以作为满足不同客户的需求需要，并作为实际系统设计方案的基础。

一般而言，软件系统的架构有两个要素如下。

- 它是一个软件系统从整体到部分的最高层次的划分。一个系统通常是由元件组成的，而这些元件如何形成、相互之间如何发生作用，则是关于这个系统本身结构的重要信息。详细地说，就是要包括架构组件（Architecture Component）、联结器（Connector）和任务流（Task-flow），所谓架构组件，也就是组成系统的核心“砖瓦”，而联结器则描述这些元件之间通信路径、通信机制和通信的预期结果，任务流则描述系统如何使用这些元件和联结器完成某一项需求。
- 建造一个系统所做出的最高层次的、以后难以更改的、商业的和技术的决定。在建造一个系统之前会有很多的重要决定需要事先做出，而一旦系统开始进行详细设计甚至建造，这些决定就很难更改甚至无法更改。显然，这样的决定必定是有关系统设计成败的最重要决定，必须经过非常慎重的研究和考察。

1.1.7 为什么要使用架构

在研发软件项目前，必须把软件架构搭建出来，这样做有以下优点。

- 可靠性。软件系统对于用户的商业经营和管理来说非常重要，因此软件系统必须非常可靠。
- 安全性。软件系统所承担的交易的商业价值极高，系统的安全性非常重要。
- 可扩展性。软件必须能够在用户的使用率、用户的数目增加很快的情况下，保持合理的性能。只有这样，才能适应用户市场扩展的可能性。
- 可定制化。同样的一套软件，可以根据客户群的不同和市场需求的变化进行调整。
- 可扩展性。在新技术出现的时候，一个软件系统应当允许导入新技术，从而对现有系统进行功能和性能的扩展。
- 可维护性。软件系统的维护包括两方面，一是排除现有的错误，二是将新的软件需求反映到现有系统中去。一个易于维护的系统可以有效地降低技术支持的花费。
- 客户体验。软件系统必须易于使用。
- 市场时机。软件用户要面临同业竞争，软件提供商也要面临同业竞争。以最快的速度争夺市场先机非常重要。

1.1.8 二层架构开发

1. 什么是二层架构

这里所说的二层架构既不是 B/S 也不是 C/S 架构，所谓二层架构就是将系统的整个业务应

