

软件工程  
技术丛书

# VISUAL BASIC

# 设计模式

VB 6.0 和 VB.NET



VB 6.0 and VB.NET

Visual Basic Design Patterns

【美】 James W. Cooper 著  
赵会群 等译



清华大学出版社

软件工程技术丛书

# VISUAL BASIC 设计模式

VB6.0 和 VB.NET

[美] James W.Cooper 著

赵会群 等译

清华大学出版社

北京

## 内 容 简 介

这是一本非常实用的 VB 设计模式专著。书中从使用 VB 进行面向对象编程的角度，讲述如何在设计模式思想指导下用 VB.NET 和 VB6 编写应用程序。

本书首先概述了 VB 面向对象编程的概念和方法，然后讨论了 23 种设计模式，每种模式都通过至少一个完整的 VB 程序来说明，以帮助读者建立设计模式的思想。这种方式使得设计模式的概念容易掌握，也更容易理解设计模式的本质及目的。读过本书，VB 程序员可以迅速提高编程水平，并从设计模式中受益。

本书可以为设计比较复杂的 VB 程序提供指导，既可以作为深入学习 VB 编程的教科书，也可作为探讨软件设计模式研究领域人员的参考书。

EISBN: 0-201-70265-7

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED AND TSINGHUA UNIVERSITY PRESS.

Original English language title: Visual Basic Design Patterns — VB 6.0 and VB.NET by James W.Cooper, Copyright © 2002.

All rights reserved.

Published by arrangement with the original publisher, Prentice Hall, a Pearson Education Company.

This edition is authorized for sale only in People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权清华大学出版社在中华人民共和国境内 (香港、澳门特别行政区除外) 出版、发行。

未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

**本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签，无标签者不得销售。**

北京市版权局著作权合同登记号 图字：01-2003-1108

**版权所有，盗版必究。**

书 名：VISUAL BASIC 设计模式——VB6.0 和 VB.NET

作 者：James W.Cooper

译 者：赵会群等

出版者：清华大学出版社 (北京清华大学学研大厦，邮编 100084)

<http://www.tup.com.cn>

印刷者：北京市耀华印刷有限公司

发行者：新华书店总店北京发行所

开 本：异 16 印张：25.625 字数：559 千字

版 次：2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

印 数：0001~4000

盘 号：ISBN 7-89494-071-2

定 价：47.00 元 (1CD)

# 译者序

设计模式一词是 20 世纪 70 年代一个叫 *Christopher Alexander* 的建筑工程师首先提出来的。在他编写的两本书中首先提到设计模式的概念，一本书叫《模式语言》，另一本书叫《永恒的建筑风格》。在这两本书中用许多实例阐述了文档模式的机理。

计算机领域中有关模式的概念在 1987 年 OOPSLA 会议上首次正式提出。此后设计模式的概念在一些知名计算机专家（如 Grady Booch, Richard Helm, Erich Gamma 和 Kent Beck）的论文和报告中频频出现。可以说在 1995 年前，设计模式概念还只停留在理论研究阶段，到 1995 年 Erich Gamma, Richard Helm, Ralph Johnson 和 John Vlissides 出版了 *design patterns: software reuse basics* 一书之后，设计模式概念才真正被广泛地应用。

设计模式是一种建议，它提供了程序共享设计的思路。在编写程序过程中，我们可能遇到许多以前已经遇到过的问题，而且今后还会继续发生，我们经常思考如何解决该问题。文本模式的思想就是你可以重用和共享你已经知道解决问题的最佳途径的信息。软件设计模式就是借鉴了这种思想。

软件设计模式作为软件重用的一种方法，对提高程序设计与开发的效率有直接的作用，学习和研究软件设计模式十分重要。软件设计模式作为一个新的研究领域，对它的研究与学习还刚刚起步，有关的理论和方法尚未完全形成。本书作为目前仅有的几本有关软件设计模式的专著，对读者学习和进一步研究软件设计模式的理论和方法有一定的作用。本书的最大特点是结合具体设计示例介绍软件设计模式的基本思想和方法，这对读者理解软件设计模式相关概念和方法十分有利。另外，书中采用比较常用的 VB6 和功能强大的 VB.NET 作为软件设计模式的实现工具，使得读者能够更容易接受软件设计模式的实现方法。

由于有关软件设计模式的专著在国内还刚刚引入，我们对这方面的学习和研究还不够深入，所以在翻译过程中可能有许多地方都没有把原著的精华体现出来，如有不当之处还请读者多多指教。本书主要的翻译工作和统稿工作由赵会群完成。此外，吴洁明、胡景德、吕春和刘娟也参加了翻译工作，15 章到 18 章由吴洁明翻译，19 章到 21 章由胡景德翻译，22 章到 26 章由吕春翻译，27 章到 32 章由刘娟翻译。

# 前 言

这是一本非常实用的 VB 设计模式专著，讲述如何在设计模式思想指导下用 VB.NET 和 VB6 编写程序，也可以作为 VB.NET 的编程导论。本书以一系列篇幅较短小的章节来讨论设计模式，每种模式一章，并给出一到两个已经用程序实现的例子，并在每一章中用 UML 图来解释类之间的相互作用。

该书不是众所周知的由“四人帮”（Gang of Four）编写的 *Design Patterns* 一书的姊妹篇，而是为那些希望深入学习和应用设计模式的人编写的导学书。在阅读本书之前没有必要一定读过 *Design Patterns*，但在阅读之后你可能急切地想去读或更细致地重读原著 *Design Patterns*。

在本书中，你将了解到设计模式是程序中经常用于组织对象的有效方法，可以使程序设计更简单并且更容易修改。你也将会发现，在逐渐地了解设计模式后，将会增加许多如何进行程序设计的新词汇。

人们开始从纯理论到纯实践的各个方面去认同设计模式，到最终发现它的巨大作用的时候，感叹将脱口而出。此刻也就意味着你的脑海里出现了如何使用设计模式来帮助你工作的设想。

在本书中，我们试图通过各种不同的方法，帮助你建立设计模式的思想。本书由引言、VB.NET 简介、设计模式一般性描述、生成模式、结构模式和行为模式六个部分构成。

对每一个设计模式，我们首先简单介绍所讨论的设计模式，然后设计简单的程序实例。每一个实例都是基于图形用户界面的程序，你可以运行这些程序，以便更具体地了解设计

模式的含义。书中的所有示例程序，以及这些示例的改进程序都放在配套光盘中，你可以从光盘中运行、修改这些程序，并且根据实际工作调整程序中的内容。

我们首先向你展示如何在 VB6 中有效地使用设计模式，然后再在 VB.NET (也被称为 VB7) 中使用同样的模式。因为每一个程序中都有许多类文件，所以我们为每一个示例建立了一个 VB 工程文件，并分别存放在各自的子目录中，从而避免混乱，我们把 VB.NET 示例分开存放于该模式的子目录中。本书基于 Beta-2 版的 VB.NET 讨论编程，它与最终的发行版不会有太大的变化。你可以从 Addison-Wesley 的网址获得书中改进的示例程序。

翻开此书，你将会看到我们制作的一些用于解释示例的程序屏幕截图，它提供另一种巩固设计模式学习的方法。另外，你也可以看到另一种用于解释类之间交互的 UML 图。这是一种用框和线来解释类之间的联系和层次结构的示意图。图中的箭头指向父类，而点线箭头指向接口。如果你不是十分了解 UML，可以仔细阅读我们在第 II 部分对 UML 的介绍。

读完此书，你将从中受益颇多，并将在日常的 VB 编程工作中使用设计模式。

James W. Cooper  
Nantuchet, MA  
Wilton, CT  
Maui, Hi

# 目 录

## 第 I 部分 VB 面向对象程序设计

<b>第 1 章 设计模式概述</b> .....	3	3.7 类初始化	25
1.1 定义设计模式	4	3.8 类和属性	26
1.2 学习过程	5	3.9 另一个接口示例——伏特计	27
1.3 学习设计模式	6	3.10 一个 vbFile 类	28
1.4 评论面向对象方法	6	3.11 Visual Basic 程序设计风格	30
1.5 VB 设计模式	7	3.12 小结	30
1.6 本书组织	7	<b>第 4 章 面向对象的程序设计</b> .....	31
<b>第 2 章 UML 图</b> .....	8	4.1 构建 VB 对象	32
2.1 继承	9	4.2 产生一个对象实例	32
2.2 接口	10	4.3 一个用 VB 实现的测量程序	32
2.3 组合	11	4.4 对象中的方法	33
2.4 注释	12	4.5 变量	33
2.5 基于 WithClass 的 UML 图	12	4.6 参数传值和传址	34
2.6 Visual Basic 工程文件	13	4.7 面向对象程序设计中的术语	34
<b>第 3 章 在 VB 中使用类和对象</b> .....	14	<b>第 5 章 创建自己的控件</b> .....	36
3.1 一个简单的温度换算程序	14	5.1 一个激活的文本	36
3.2 构建一个温度类	15	5.1.1 调整用户控件大小	37
3.2.1 换算到开氏温标	17	5.2 测试 HiText 控件	38
3.3 在 Temperature 类作决定	17	5.3 在用户控件中增加属性和方法	39
3.4 在类中进行数据的格式化和值转换	18	5.4 编译一个用户控件	40
3.4.1 处理不合理值	20	5.5 小结	40
3.5 一个字符串的分割类	21	5.6 光盘中的程序	40
3.6 类对象	23	<b>第 6 章 继承和接口</b> .....	41
3.6.1 类包含	24	6.1 接口	41
		6.2 一个投资模拟器	42

6.3 编写一个模拟器 .....	43	7.16.1 从长方形中产生一个正方形 .....	68
6.4 用于接口使用的指示器 .....	44	7.17 Public, Private 和 Protected .....	69
6.5 重新使用共同的方法 .....	47	7.18 在导出类中重载方法 .....	69
6.6 隐藏接口 .....	49	7.19 重载与隐蔽 .....	70
6.7 小结 .....	49	7.20 重载窗口控件 .....	72
6.8 光盘中的程序 .....	49	7.21 接口 .....	73
<b>第 7 章 VB.NET 简介 .....</b>	<b>50</b>	7.22 小结 .....	74
7.1 VB.NET 中新的语法 .....	50	7.23 光盘中的程序 .....	74
7.1.1 改进的函数语法 .....	51	<b>第 8 章 VB.NET 中的数组、文件和异常 .....</b>	<b>75</b>
7.2 变量声明和作用域 .....	52	8.1 数组 .....	75
7.2.1 VB.NET 中的对象 .....	53	8.2 集合对象 .....	76
7.3 编译选择 .....	53	8.2.1 数组列表 .....	76
7.3.1 VB.NET 中的数值型变量 .....	54	8.2.2 Hashtable .....	77
7.4 VB6 和 VB.NET 中的属性 .....	54	8.2.3 SortedList .....	78
7.5 快捷等号语法 .....	55	8.3 异常 .....	78
7.6 管理语言和垃圾回收 .....	56	8.4 多重异常 .....	79
7.7 VB.NET 中的类 .....	56	8.5 抛出异常 .....	80
7.8 构建一个 VB7 应用 .....	58	8.6 文件处理 .....	80
7.9 VB.NET 最简单的窗口程序 .....	60	8.6.1 File 对象 .....	80
7.10 继承 .....	61	8.6.2 读一个文本文件 .....	81
7.11 构造函数 .....	63	8.6.3 写一个文本文件 .....	81
7.12 VB.NET 中的图画 .....	65	8.7 在文件处理中使用异常 .....	82
7.13 工具标签和鼠标移动键 .....	65	8.8 测试文件结束 .....	82
7.14 重载 .....	66	8.9 FileInfo 类 .....	83
7.15 继承 .....	66	8.10 vbFile 类 .....	84
7.16 名字空间 .....	67	8.11 光盘中的程序 .....	85

## 第 II 部分 生成模式

<b>第 9 章 简单工厂模式 .....</b>	<b>87</b>	9.2 代码片段 .....	87
9.1 一个简单工厂如何工作 .....	87	9.3 两个导出类 .....	88



9.4 构建简单工厂 .....	89	12.5.2 VB7 实现的单一类中的错误 处理 .....	120
9.4.1 使用工厂 .....	89	12.6 一个 VB.NET 实现的 SpoolDemo 程序 .....	122
9.5 用 VB.NET 编写工厂模式 .....	91	12.7 全局访问点 .....	123
9.6 使用数学计算的工厂模式 .....	92	12.8 单一类模式其他结论 .....	123
9.7 光盘中的程序 .....	93	12.9 光盘上的程序 .....	123
<b>第 10 章 工厂方法模式 .....</b>	<b>94</b>	<b>第 13 章 构造器模式 .....</b>	<b>124</b>
10.1 Swimmer 类 .....	96	13.1 一个投资跟踪程序 .....	124
10.2 Events 类 .....	97	13.2 调用构造器 .....	126
10.3 直接筛选 .....	99	13.3 列表框构造器 .....	128
10.3.1 交叉筛选 .....	99	13.4 复选框构造器 .....	128
10.4 我们的选拔程序 .....	100	13.5 用 VB.NET 实现构造器 .....	129
10.5 其他的类工厂 .....	101	13.5.1 股票类工厂 .....	131
10.6 用 VB7 实现的选拔程序 .....	101	13.5.2 复选框类 .....	132
10.7 什么时候使用工厂方法 .....	103	13.5.3 列表框类 .....	133
10.8 光盘上的程序 .....	104	13.6 在列表框中使用下标集合 .....	134
<b>第 11 章 抽象工厂模式 .....</b>	<b>105</b>	13.6.1 最终选择 .....	135
11.1 一个花卉工厂 .....	105	13.7 小结 .....	137
11.2 用户接口如何工作 .....	108	13.8 光盘中的程序 .....	137
11.3 用 VB7 实现一个抽象工厂 .....	108	<b>第 14 章 原型模式 .....</b>	<b>138</b>
11.3.1 PictureBox 框 .....	110	14.1 VB6 中的克隆 .....	138
11.3.2 处理单选按钮和按钮事件 .....	111	14.2 使用原型 .....	139
11.4 增加更多的类 .....	112	14.3 使用原型模式 .....	142
11.5 抽象工厂评价 .....	113	14.3.1 在子类中增加方法 .....	143
11.6 光盘中的程序 .....	113	14.3.2 具有相同接口的不同类 .....	144
<b>第 12 章 单一类模式 .....</b>	<b>114</b>	14.4 原型管理器 .....	147
12.1 使用静态方法产生单一类 .....	114	14.5 用 VB7 编写原型 .....	147
12.2 捕获错误 .....	116	14.6 小结 .....	150
12.3 提供单一类的全局访问点 .....	116	14.7 光盘中的程序 .....	151
12.4 MSComm 控件作为单一类 .....	117	14.8 生成模式总结 .....	151
12.4.1 可用的串口 .....	118		
12.5 用 VB.NET 实现单一类 .....	119		
12.5.1 使用私有的构造函数 .....	120		

## 第III部分 结构模式

<b>第 15 章 适配器模式</b> .....	<b>153</b>	17.3 员工类 .....	179
15.1 在列表中移动数据.....	153	17.4 下标类 .....	182
15.2 使用 MSFlexGrid .....	154	17.5 Boss 类.....	183
15.3 使用 TreeView .....	157	17.6 构建员工树.....	185
15.3.1 对象适配器.....	157	17.7 自提升 .....	187
15.4 在 VB7 中使用适配器 .....	158	17.8 双向链表.....	187
15.5 VB.NET 的 TreeView 适配器.....	160	17.9 小结 .....	188
15.6 采用 DataGrid 控件.....	161	17.10 一个简单组合.....	189
15.7 类适配器 .....	163	17.11 VB 中的组合 .....	189
15.8 两路适配器 .....	164	17.12 VB.NET 中的组合 .....	189
15.9 在 VB.NET 中实现对象和类 适配器 .....	164	17.12.1 枚举器.....	191
15.10 可插入的适配器 .....	164	17.12.2 Boss 构造函数多态 .....	191
15.11 在 VB 中的适配器 .....	164	17.13 其他实现条款.....	192
15.12 光盘中的程序 .....	165	17.14 光盘中的程序.....	193
<b>第 16 章 桥模式</b> .....	<b>166</b>	<b>第 18 章 修饰模式</b> .....	<b>194</b>
16.1 visList 类 .....	169	18.1 CoolButton 按钮的修饰 .....	194
16.2 类关系图 .....	169	18.2 使用 Decorator.....	197
16.3 桥模式的扩展 .....	169	18.3 将 ActiveX 控件作为 Decorator 使用.....	200
16.4 ActiveX 控件作为桥 .....	172	18.4 VB.NET 中的 Decorator.....	200
16.5 用 VB.NET 实现桥模式 .....	172	18.5 不可见的 Decorator.....	202
16.5.1 ListBox 的 visList 类.....	173	18.6 修饰、适配和组合模式.....	202
16.5.2 Grid 的 visList 类.....	174	18.7 小结 .....	203
16.5.3 导入数据.....	174	18.8 光盘中的程序.....	203
16.6 改变数据格式 .....	175	<b>第 19 章 伪模式</b> .....	<b>204</b>
16.7 小结 .....	177	19.1 数据库是什么.....	204
16.8 光盘中的程序 .....	177	19.2 从数据库中获得数据.....	205
<b>第 17 章 组合模式</b> .....	<b>178</b>	19.3 数据库的种类.....	206
17.1 一个组合的实现 .....	179	19.4 ODBC .....	206
17.2 计算工资 .....	179	19.5 微软的数据库连接策略.....	207

19.6 数据库的结构 .....	207	19.19 装载数据库表 .....	234
19.6.1 DBase 类 .....	207	19.20 最终的应用程序 .....	235
19.7 建立 Façade 类 .....	210	19.21 Façade 的组成 .....	236
19.7.1 Stores 类 .....	212	19.22 小结 .....	236
19.8 建立 Stores 和 Foods 表 .....	214	19.23 光盘中的程序 .....	237
19.9 建立 Price 表 .....	215	<b>第 20 章 轻量模式 .....</b>	<b>238</b>
19.9.1 建立价格查询 .....	216	20.1 讨论 .....	239
19.10 小结 .....	217	20.2 举例 .....	239
19.11 在 VB6 中使用 ADO 访问 数据库 .....	218	20.2.1 类结构图 .....	242
19.11.1 使用 ADO 连接数据库 .....	218	20.2.2 选择一个文件夹 .....	242
19.11.2 在表中添加或查询记录 .....	219	20.3 用 VB.NET 实现轻量模式文件夹 .....	243
19.11.3 使用 ADO 扩展 .....	220	20.4 VB 中 Flyweight 的使用 .....	247
19.12 ADO 中的 DBase 类 .....	221	20.5 可共享对象 .....	247
19.13 在 VB.NET 中访问数据库 .....	224	20.6 Copy-on-write 对象 .....	248
19.14 使用 ADO.NET .....	224	20.7 光盘中的程序 .....	248
19.14.1 连接数据库 .....	225	<b>第 21 章 代理模式 .....</b>	<b>249</b>
19.14.2 从数据库表中读取数据 .....	225	21.1 示例 .....	249
19.14.3 执行查询 .....	226	21.2 用 VB.NET 实现代理模式 .....	251
19.14.4 删除表中的内容 .....	226	21.3 VB 中的 Proxy .....	253
19.15 使用 ADO.NET 向数据库表中 添加记录 .....	227	21.4 Copy-on-write .....	253
19.16 编写 VB.NET ADO 伪模式 .....	228	21.5 相关模式的比较 .....	253
19.16.1 DBTable 类 .....	229	21.6 光盘中的程序 .....	254
19.17 为每一个表格创建类 .....	230	21.7 结构模式总结 .....	254
19.18 存储价格 .....	232		

## 第IV部分 行为模式

<b>第 22 章 响应链 .....</b>	<b>256</b>	22.4.1 获得帮助命令 .....	263
22.1 适用范围 .....	256	22.5 链还是树 .....	264
22.2 代码示例 .....	257	22.6 用 VB.NET 实现响应链 .....	266
22.3 列表框 .....	260	22.7 请求的种类 .....	268
22.4 实现帮助系统 .....	262	22.8 VB 中的示例 .....	268

22.9	小结 .....	269	25.4	用 VB.NET 实现迭代 .....	309
22.10	光盘中的程序 .....	269	25.5	小结 .....	311
<b>第 23 章</b>	<b>命令模式 .....</b>	<b>270</b>	25.6	光盘中的程序 .....	311
23.1	目的 .....	270	<b>第 26 章</b>	<b>协调模式 .....</b>	<b>312</b>
23.2	命令对象 .....	271	26.1	一个示例 .....	312
23.3	建立 Command 对象 .....	271	26.2	控件间的交互 .....	313
23.4	命令数组 .....	272	26.3	代码示例 .....	314
23.5	命令模式小结 .....	275	26.3.1	系统的初始化 .....	317
23.6	提供撤消 .....	275	26.4	协调和命令对象 .....	317
23.7	VB.NET 中的命令模式 .....	280	26.5	用 VB.NET 实现协调模式 .....	318
23.8	CommandHolder 接口 .....	282	26.5.1	初始化 .....	320
23.9	在 VB.NET 中处理撤消命令 .....	284	26.5.2	处理新控件的事件 .....	320
23.10	VB 中的命令模式 .....	288	26.6	小结 .....	321
23.11	光盘中的程序 .....	288	26.7	单接口协调类 .....	322
<b>第 24 章</b>	<b>解释模式 .....</b>	<b>289</b>	26.8	用法的讨论 .....	322
24.1	目的 .....	289	26.9	光盘中的程序 .....	322
24.2	适用范围 .....	289	<b>第 27 章</b>	<b>记事模式 .....</b>	<b>323</b>
24.3	一个简单的报告示例 .....	290	27.1	目的 .....	323
24.4	解释这种语言 .....	290	27.2	实现 .....	324
24.5	解析时使用的对象 .....	291	27.3	示例 .....	324
24.6	减少分析栈 .....	295	27.3.1	注意事项 .....	330
24.7	实现解释模式 .....	296	27.4	用户接口中的命令对象 .....	330
24.7.1	语法树 .....	296	27.5	处理鼠标和画图事件 .....	331
24.8	用 VB6 实现解释器 .....	300	27.6	用 VB.NET 编写 Memento .....	332
24.9	解析对象 .....	302	27.7	小结 .....	334
24.10	小结 .....	303	27.8	光盘中的程序 .....	335
24.11	光盘上的程序 .....	303	<b>第 28 章</b>	<b>观察模式 .....</b>	<b>336</b>
<b>第 25 章</b>	<b>迭代模式 .....</b>	<b>304</b>	28.1	查看颜色的变化 .....	337
25.1	目的 .....	304	28.2	用 VB.NET 实现观察模式 .....	339
25.2	VB6 示例代码 .....	305	28.3	传递的消息 .....	341
25.2.1	使用迭代 .....	306	28.4	小结 .....	342
25.3	过滤迭代 .....	307	28.5	光盘中的程序 .....	342
25.3.1	过滤迭代器 .....	307			

<b>第 29 章 状态模式</b> .....	<b>343</b>	<b>第 31 章 模板方法模式</b> .....	<b>374</b>
29.1 示例代码.....	343	31.1 目的.....	374
29.2 状态间的转换.....	348	31.2 Template 类中的方法.....	375
29.3 Mediator 和 StateManager 之间的 交互.....	349	31.3 示例.....	376
29.4 处理 Fill 状态.....	352	31.3.1 画一个标准的三角形.....	377
29.5 处理撤消列表.....	352	31.3.2 画一个等腰三角形.....	378
29.5.1 用 VB6 实现填充圆形.....	355	31.4 画三角形程序.....	379
29.6 在 VB.NET 中实现状态模式.....	355	31.5 模板和回调.....	380
29.7 Mediator 和 God 类.....	360	31.6 小结.....	381
29.8 小结.....	361	31.7 光盘中的程序.....	381
29.9 状态转换.....	361	<b>第 32 章 访问者模式</b> .....	<b>382</b>
29.10 光盘中的程序.....	362	32.1 目的.....	382
<b>第 30 章 策略模式</b> .....	<b>363</b>	32.2 使用访问者模式的时机.....	383
30.1 目的.....	363	32.3 示例.....	384
30.2 示例代码.....	364	32.4 访问类.....	385
30.3 Context 类.....	365	32.5 访问多个类.....	386
30.4 程序命令.....	365	32.6 经理也是员工.....	388
30.5 线形图和柱形图策略.....	366	32.7 访问者的全部捕获操作.....	388
30.6 用 VB 绘图.....	367	32.8 两次分派.....	389
30.7 用 VB.NET 实现策略模式.....	369	32.9 为什么要这样做.....	389
30.8 小结.....	372	32.10 遍历一系列的类.....	390
30.9 光盘中的程序.....	373	32.11 在 VB6 中编写访问者.....	390
		32.12 小结.....	393
		32.13 光盘中的程序.....	393
		<b>参考文献</b> .....	<b>394</b>

# 第 I 部分

---

## VB 面向对象程序设计

本书的第 I 部分首先介绍面向对象程序设计的基本概念, 以及用 VB 进行面向对象程序设计的方法。内容包括如何用 UML 图表示程序及其类对象, 如何用 VB6 构建自己的控件; 还将介绍对象的继承和接口概念, 以及 VB.NET 的基本编程方法。在此基础上, 还将进一步学习如何用 VB6 和 VB.NET 这些强大的语言工具进行简化程序设计的技术。

在学完上述导引性内容之后, 你就可以深入地学习和使用本书介绍的各种设计模式。



# 第 1 章 设计模式概述

当坐在计算机前开始设计一个新程序的时候，你直观地知道要实现的功能，用什么对象和什么数据结构来实现这些功能，但如何选择更好的、更通用的编程方法仍然不是十分明确。

事实上，在想清楚程序中包括哪些代码模块以及这些代码模块的接口之前，你可能不会编写一条语句。越是全局考虑，编写的程序越有效。如果在编写程序代码之前，你还没有形成全局观点，那么最好还是暂时离开计算机，放眼窗外，即使是程序设计的方法比较简单，你也要如此。

在某种意义上，越优雅的程序的可复用性与可维护性也会越好。但在你放心地设计完一个相对优雅的解决方案时，却有可能存在着很多未经显露的内在问题。

设计模式开始引起计算机学者重视的原因之一，它是实现简单、精巧和可重用解决方案的基础。术语“设计模式”对从未听过的人来说有点太正规，甚至容易引起困惑。但事实上，它是一种方便的重用方法，无论是工程之间，还是程序员之间都是如此。设计模式的思想并不复杂，简单地说，设计模式就是典型的对象之间交互的定义和分类。

早期的关于程序框架的文献中经常引用的模式是 Smalltalk (Krasner and Pope 1988) 中提出的“模型-视图-控制器”框架 (Model-View-Controller)。在该模式下，把用户接口设计问题分成 3 个部分，如图 1-1 所示。这 3 个部分分别是数据模型 (Data Model)，用于计算；视图 (View)，表现用户接口；控制器 (Controller)，实现用户与视图的交互。

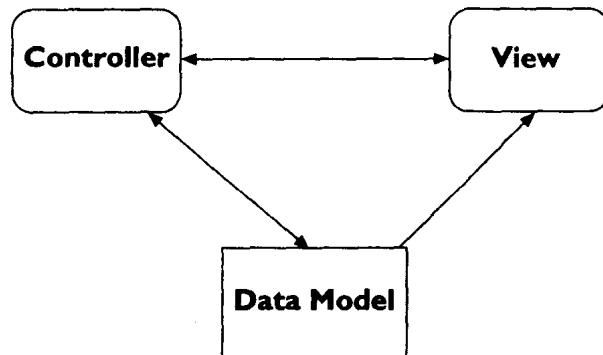


图 1-1 模型-视图-控制器框架

这 3 个部分构成 3 个不同的对象，它们有其各自的数据处理规则。用户、用户界面和



数据之间的通信应被细心地组织，而且它们的功能被有效地实现，它们采用一定的连接约束集进行交互，这就是一个典型的设计模式实例。

另一方面，从面向对象程序设计的角度来看，设计模式描述了如何使得不同对象的方法和数据方法在对象通信中保持相互独立，这正是面向对象程序设计的目标。如果你一直是这样做的，那么你可能正在使用设计模式的基本思想。

设计模式是 20 世纪 90 年代初期由 Erich Gamma (1992 年) 正式提出的，他结合 GUI 应用框架 ET++，对设计模式给出描述。之后设计模式引起广泛关注，组织了许多有关的讨论和会议，但最能反映学术水平的是由 Gamma、Helm、Johnson 和 Vlissides 于 1995 年撰写的 *Design Patterns — Elements of Reusable Software* (《设计模式——可复用面向对象软件的基础》)。该书对设计模式的研究产生了重要的影响，深受读者的好评，读者也风趣的把这本书称为“四人帮”(Gang of Four, GoF)。这本书中介绍了常见的 23 种设计模式，讲述了何时使用和如何使用这些模式。在本书中，我们提到这本里程碑式的巨著时将简称它为 *Design Patterns*。

目前已出版了一些有关设计模式的专著，比较相关的一本书是 Alpert、Brown 和 Woolf 于 1998 年撰写的 *The Design Patterns Smalltalk Companion*，该书包括 23 个与 *Design Patterns* 相同的基于 Smalltalk 的设计模式，所以我们把该书称为 Smalltalk 实现。最近出版的一本关于设计模式的书是 *Java Design Patterns: A Tutorial*，书中完全用 Java 来解释设计模式思想。

## 1.1 定义设计模式

在日常生活中，我们经常可以发现许多事情的处理是按一定的模式进行的。同样，在编写程序时也是如此，并完全接受对象之间以相对独立的方式进行通信的这种有效的编程模式。

下面是在一些文献中出现的设计模式定义。

- “设计模式是对常见设计问题的可重复使用的解决方案” (The Smalltalk Companion)。
- “设计模式构成一个规则集合，这些规则描述如何实现软件开发工作中的特定任务” (Pree 1994)。
- “设计模式更加注重体系结构设计的重用，而框架关心软件设计和实现的细节” (Coplien and Schmidt 1995)。
- “模式阐述了在特定设计环境下出现的设计问题，并给出它的解决方案” (Buschmann et al. 1996)。
- “模式是对类、实例或组件的更高层次抽象的标识和说明” (Gamma et al. 1993)。