PROGRAMMER TO PROGRAMMER™

Professional Refactoring in C# & ASP.NET

代码重构

（C# & ASP.NET版）

(美) Danijel Arsenovski 著
潘立福 权乐 译

wrox™

清华大学出版社

代码重构

(C# & ASP.NET 版)

(美) Danijel Arsenovski 著

潘立福 权乐 译

清华大学出版社

北 京

Danijel Arsenovski
Professional Refactoring in C# & ASP.NET
EISBN: 978-0-470-43452-9

北京市版权局著作权合同登记号　图字：01-2010-1675

图书在版编目(CIP)数据

代码重构(C# & ASP.NET 版)/(美)阿瑟诺维斯基(Arsenovski, D.) 著；潘立福　权乐 译.
—北京：清华大学出版社，2011.6
书名原文：Professional Refactoring in C# & ASP.NET
ISBN 978-7-302-25555-0
Ⅰ. 代… Ⅱ. ①阿… ②潘… ③权… Ⅲ. ①C 语言—程序设计 ②网页制作工具—程序设计
Ⅳ. ①TP312②TP393.092
中国版本图书馆 CIP 数据核字(2011)第 082547 号

责任编辑：王　军　于　平
装帧设计：孔祥丰
责任校对：胡雁翎
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京密云胶印厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：32.25　**字　数**：785 千字
版　　次：2011 年 6 月第 1 版　　**印　　次**：2011 年 6 月第1 次印刷
印　　数：1～4000
定　　价：68.00 元

产品编号：035684-01

作 者 简 介

Danijel Arsenovski 是一位作家、软件架构师，也是一位敏捷指导员。他目前担任Excelsys S.A.公司的产品和解决方案架构师，主要负责为当地的大量客户设计 Web 2.0银行解决方案。在整改大型银行系统时，他开始尝试重构，并始终对重构保持着浓厚的兴趣。他因提倡在.NET 平台上运用重构而声名远扬。Arsenovski 是 *Visual Studio* Magazine、.NET Developers Journal 和 Visual Systems Journal 的撰稿人，拥有 Microsoft Certified Solution Developer(MCSD)证书，并被评为 2005 年的 Microsoft MVP。可以通过 danijel.arsenovski@empoweragile.com 与他联系，也可以访问其博客(http://blog.refactoring.com)。

前　言

感谢您选择本书，欢迎进入神奇的重构世界。笔者希望，您在进行日常编程工作时，在与同行讨论不同的设计方案时，在准备处理一些晦涩的遗留代码时，或甚至晚上在头脑中琢磨代码行时，能发现本书的用途。如果这是您第一次接触到重构，那么希望本书能深刻改变您编程和思考代码的方式。然而，完成这项任务并不是一件简单的事情，最终还需要由您来评价本书的价值。

笔者在阅读完 Martin Fowler 编著的 *Refactoring: Improving the Design of Existing Code*(Addison-Wesley，1999)一书之后，以系统化的方式采纳了重构。事实证明，该书是一本十分实用的著作，它介绍了一些可立刻运用于实际项目的必要技术。该书没有建立在复杂理论的基础之上，也没有包含任何复杂的数学公式。

该书通俗易懂，任何编写代码的人都能够立刻理解。在阅读完该书之后，笔者注意到自己在编程方式上的很多变化：

- 能够发现更多确定的问题代码和设计缺陷。
- 能够为这些问题思考解决方案并通过重构有效地解决它们。
- 在与同事讨论时，能够简洁明了地阐述自己的决策。

笔者不再将代码视为一成不变的结构，而将其看做一个塑性的、可模化的形式，可以根据自己的需求和爱好进行定制。这使得处理代码的方式发生了根本的改变。笔者意识到存在一种方法，通过该方法可以采用一种高效、可预知的方式来修改代码并改进其设计。

很快，关于重构的言论在我们的团队中流传开来，笔者看到越来越多的同事从书架上取下这本书仔细阅读，甚至有人还自费购买了该书。笔者也因此可以使用重构的术语来与同事们交谈，并将重构作为软件构建过程中不可缺少的部分介绍给大家。这甚至证明了管理层在这一方面是有远见的。

一部分是因为对于学习不同的语言和技术感兴趣，另一部分是因为工作上的需要，所以笔者开始与不同的团队一起工作并用不同的语言来编程。在与用 C#编程的团队一起工作时，笔者试图像对 Java 团队做的那样传播重构知识。事实证明这一过程并非是一帆风顺。笔者很快意识到，对于.NET 的编程人员而言，几乎没有什么关于重构的信息可供使用。虽然在任何面向对象的编程语言中，可以以类似的方式使用大部分重构，但是仍存在一些微妙的区别。在编程人员看到以其自己偏好的语言编写的代码示例之后，他不学习重构的理由将不复存在。

这启发了笔者首先编写了一本关于 Visual Basic .NET 中重构的书，并最终编著了这本关于 C#和 ASP.NET 中重构的书。笔者认为本书确有存在的必要，C#编程人员将发现该书是非常实用的。

本书读者对象

本书面向经验丰富的中高级 C#开发人员，可引领他们进入重构世界中。为了能够利用本书收到最圆满的学习效果，读者应该较好地掌握 C#编程语言，尤其是面向对象编程原则。

如果您还只是一个编程新手，那么无法将本书用作初级入门书籍。本书的目的不是传授 C#编程的基本知识。然而，在职业生涯中不尽早地熟悉重构是没有任何理由的。随着学会编写第一个类，您就可以使用本书来学习如何正确地设计并纠正可能引入到设计中的任何错误。

如果您是从其他编程语言(如 C++、Java 或 VB.NET)转为学习 C#的，也会发现本书很有用。对于过渡中的编程人员来说，新语言的语法通常很容易掌握，但是真正领会编程语言的精神则要难得多。本书可以帮助您“以 C#的精神”来编程。

本书没有对应用程序的类型或领域做出任何假设。所以，您的应用程序可以是通常的 Web 应用程序、Web 服务、架构、组件、购物车应用程序、新的 Facebook 小部件或射击游戏。但是不管是什么应用程序，只要其中包含 C#代码，您都会发现本书所阐释的技术是很有价值的。

此外，本书还讨论了一些与 ASP.NET 相关的重构知识。如果您正在使用.NET 创建 Web 应用程序并编写 HTML 和 ASP.NET 代码，那么这些内容会最让人感兴趣。本书并未仅局限于 C#代码范围，也处理了诸如 HTML、XHTML、CSS 和 HTTP 的代码和技术。虽然任何 ASP.NET 开发人员都能读懂这些讨论 ASP.NET 的章节，但是 ASP.NET 代码的示例是使用 C#语言编写的。

本书所讨论的大部分重构是适用于所有完全面向对象语言的标准重构。这意味着如果使用其他某种面向对象的语言(如 C++)进行编程，那么只要熟悉 C#语法且能够阅读代码示例，就可以理解和运用本书所提供的信息。

本书主要内容

本书将详尽介绍 C#中的重构和 ASP.NET 中的一些重构概念。

本书包括如下基本的重构概念：

- 代码味道
- 重构代码转换
- 一些基本的面向对象原则
- 使用重构工具自动完成重构过程

本书只使用了一个示例(对于一本书而言，该示例的规模相对较大)来演示重构在较大基本代码上的实际应用。

除了适用于任何面向对象语言的标准重构之外，本书还介绍了一些 C#专有的重构，以及重构的一些特殊用法。例如，如何将经典的 C#代码转换成 LINQ，以及如何使用扩展方法。此外，还包括了设计模式的主题。

本书包含了大量重要的味道和重构，然而，本书并没有给出完整的重构目录。由于时

间和篇幅的限制，一些重要的重构并没有介绍。例如，本书并没有讨论诸如 Simplify Conditional 或 Reverse Conditional 之类的重构，而这些重构已经在 Developer Express 的 Refactor！for ASP 插件中实现自动化。

此外，本书也没有讨论很多“逆向”重构，如 Inline Class。本书的首要目标是介绍重构。笔者的经验表明编程人员刚开始接触重构时首先需要处理的问题是结构化很差的代码问题，而诸如 Extract Method 或 Extract Class 之类的重构将帮助解决这一问题。与其相对的，Inline Method 和 Inline Class 重构将帮助处理过分结构化的代码。这些重构将帮助消除不再需要的构造(如方法或类)。这种情况常常在对基本代码运用了广泛的重构之后才发生。

然而，这并不意味着“逆向”重构不重要。在了解重构的过程中，对这些重构的需要可能就在掌握了基本的重构之后出现。从中也可让人认识到学无止境。例如，随着每个新版本的 Refactor!插件发布，将有新的重构添加到支持的重构菜单中。

人们不断地发明和修改重构。随着逐渐熟悉该技术，读者将会发明自己的重构并可能最终决定与其他人共享这些重构。因此，笔者建议读者跳出本书的范围；不要局限于本书中所罗列的重构。寻找和发明新的重构，这样将能真正掌握持续代码改进的艺术。

本书的组织结构

因为本书是第一本专门介绍 C#和 ASP.NET 中重构的书籍，也可能是读者阅读的第一本关于重构的书，笔者希望本书完成如下多个目标：

- 详尽地介绍 C#中的重构。
- 详尽地介绍日常编程会议中可能会咨询到的重构技术和代码味道。
- 通过一个示例演示如何在现实情形中运用重构技术。这个示例就是 Rent-a-Wheels 应用程序，在本书的每 1 章最后都将分析和修改该应用程序。

为了达到上述目标，本书主体部分的叙述与其他的技术书籍是类似的。在介绍和阐述新的概念时，也是从逻辑上由基本到复杂逐步推进。对于每个概念，本书给出了用于举例的代码示例。通过这种方法，可以从头到尾按照一种逻辑顺序来阅读本书。在得到本书并首次打开之后，读者通常会这样浏览一两遍。

除了主要的叙述部分之外，还将看到本书中散布着很多味道、重构和面向对象设计原则的所谓的“定义框”。这些定义的目的是提供相应主题的综述。例如，对于味道而言，其定义包含了关于如何发现该味道的试探法。对于重构的情况而言，有一个“技巧”部分。它包含了特定的方法，指明了为高效地执行特定重构而必须完成的步骤。在日常工作期间，您可以通过参考这些内容来提醒自己如何完成重构、如何发现某种味道以及使用什么重构来消除该味道等。

大部分章节的末尾都将讨论其章节中谈到的重构、味道以及原则是如何反映到本书所给的示例中的。读者可以下载每一章中的代码并查看当前一章中的示例发生了何种变化。示例分析的目的是展示重构的实际应用。在很多技术性的书籍中常常会发现，为了论证观点，对选中的代码示例进行了不太实际的简化。虽然这样的处理让示例更加清楚且更易于编者进行演示，但是这常常意味着读者将在现实世界中碰到更复杂的情形，而且如果可以

在生产代码上运用特定的技术，则可能会产生无法预计的障碍。本书试图通过Rent-a-Wheels 示例来向读者呈现一个更为贴近实际的情形。

本书分成了五个部分，这些部分将引导读者按照逻辑顺序从较为基本的概念深入到较为复杂的概念中。

- 第 1 章~第 4 章为本书奠定了基础。例如，在第 1 章中，概括性地讨论了有关重构的内容。该章还澄清了一些关于重构的错误理解。第 2 章提供了在实践中重构的初次尝试。随后，第 3 章介绍了为自动化重构工作而必不可少的工具。第 4 章则提供了整本书中使用的一个示例。
- 第 5 章~第 8 章主要讨论了一些核心的标准重构。该部分介绍了为代码构造选择名称的重要性以及重复代码可能带来的毁灭性的后果。该部分还讨论了诸如 Extract Method 之类的标准重构并详细介绍了方法层次上的代码结构化。
- 第 9 章~第 11 章主要讨论了一些高级重构，这些重构最大限度地利用了编程环境中面向对象的功能。对于本书这部分而言，良好的面向对象的技能是至关重要的。这一部分将介绍如何发现类、创建继承层次结构以及大规模地重新组织代码等。
- 第 12 章~第 13 章将介绍为了达到具体的目标如何成功地运用重构。例如，如何将重构与设计模式配合使用以生成更为完善的设计方案。该部分将展示一些与 C#的 Visual Studio 2008 版本一起装载的新功能以及如何才能使用重构来最大化地利用它们。第 13 章将讨论 LINQ、扩展方法以及其他 C# 3.0 功能的使用。甚至回到了第 2 章中的示例，并从 C# 3.0 的角度来重新审视这个示例应用程序。
- 第 14 章~第 15 章讲述的是 ASP.NET 中的重构。第 14 章为第 15 章的内容奠定了基础。这一章首先讨论了 Visual Studio 插件 Refactor! for ASP.NET，然后解释了与 Web 应用程序和 HTML 相关的很多味道的历史背景。第 15 章主要介绍了如何让代码遵从更新的 Web 标准(如 XHTML 和 CSS)以及如何从 ASP.NET 代码重用机制(如母版页面和用户控件)中获益。

使用本书的要求

为了能充分使用本书，需要如下的软件：

- Visual Studio .NET 2008——为了能利用 Refactor！for ASP 插件，至少需要安装 Professional Edition。在免费的 C# 2008 Express Edition 中，仍然可以调试、执行代码，并手动地完成重构。如果使用的是 Visual Studio 2005，那么也可以利用本书大部分的内容。但第 13 章除外，因为这一章专门讨论了只有 C# 3.0 和 Visual Studio 2008 才提供的功能。
- Developer Express 的 Refactor! for ASP.NET 插件——从 Developer Express 站点上下载免费的 Visual Studio 插件的最新版本，网址为：www.devexpress.com/Products/Visual_Studio_Add-in/RefactorASP/。

- Microsoft SQL Server——为了运行本书中包含的示例，需要安装 Microsoft SQL Server。可以安装 MS SQL Server 2000、2005 或 2008，对于本书介绍的应用程序而言，Express 版本已经足够。
- 操作系统：可以使用任何可以运行上述软件的操作系统，如 Windows XP、Windows Server 2003 或 2008，或者 Windows Vista。

味道、重构以及面向对象设计原则的方框

在本书中，将主要碰到 3 种位于方框中的文本：味道、重构以及面向对象设计原则。

味道

这些方框包含了关于代码味道的概括性定义。代码味道是一个非常重要的重构概念，每个方框包含如下四个部分。

- **简略图**　虽然是松散地建立在 UML 静态类图的基础上，但是从 UML 语言的意义上讲，这些图无论如何都是不严格的。这些图应该让重构在代码上执行的转换更加形象。不是所有的重构都有这样的简略图；一些小规模的方法层次上的重构并不适合这种图。在创建这些图时，笔者的灵感来自 Sven Gorts 在其“Refactoring Thumbnails”一文中的工作。更多关于他的图的信息，请访问该文网址 www.refactoring.be/thumbnails.html。
- **检测该味道**　介绍一些简单的试探法来检测代码中的代码味道。
- **相关的重构**　列出为消除该味道可以使用的重构。
- **基本原理**　以更理论化的术语来解释代码味道的负面影响。

重构

这些方框介绍了重构的基本内容。每个方框包括如下部分：

- **动机**　解释该重构带给代码的好处以及如何改进应用程序的设计。
- **相关的味道**　列出该重构可以帮助消除的味道。
- **技巧**　提供执行该重构的方法步骤。

面向对象的设计原则

在这些方框中，定义了一些重要的面向对象设计原则并常常使用一些简短的代码示例来阐释它们。

源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源

代码文件。本书使用的所有源代码都可以从本书合作站点 http://www.wrox.com/ 或 http://www.tupwk.com.cn/downpage 上下载。登录到站点 http://www.wrox.com/，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有源代码。

提示：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-43452-9。

下载代码后，只需要用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 http://www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者更方便地学习，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们将会检查您的反馈信息。如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 http://www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传达感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 http://p2p.wrox.com 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

(1) 进入 p2p.wrox.com，单击 Register 链接。

(2) 阅读使用协议，并单击 Agree 按钮。

(3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。

(4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目　录

第 1 章

重构的全面介绍

如果查看当今任何主要集成开发环境(IDE)，必然会在某些地方发现“重构”选项。如果继续跟随编程社区的开发行为，那么肯定会发现很多与该主题相关的文章和书籍。对于某些编程人员来说，自设计模式开始，重构就是编程过程中最重要的开发技术。

与其他时尚的东西不同，因为重构可以帮助编程人员和编码员更好且更有成效地工作，所以他们都很乐于接受并热心传播这项技术。毫无疑问，不管使用的是什么工具、编程语言，也无论正在开发什么类型的程序，重构的应用已经成为编程人员日常工作中非常重要的部分。C#就是其中之一；目前，大量的开发人员使用 C#语言来完成重构，同时很多优秀且成熟的工具可用于自动完成重构过程。

本章包含以下内容：

- 重构的含义以及它的重要性
- 重构带来的好处
- 有关重构的一些常见错误看法
- 有关 C#编程语言的一些具体情况，以及重构是如何与 C#语言密切相关的

下面首先介绍一些与重构相关的背景。

1.1 重构的快速浏览

在处理某些编程任务时，可以使用很多方法来完成。虽然只是从一个概念开始的，但是随着进一步了解细节信息，难免遇到这些问题：“是否应该把这种方法放到类中或可能的其他类中？是使用类表示数据类型，还是直接使用基本类型？是否要把这个类分成多个部分？两个类之间是否存在继承关系或是否应该只是使用组合？”

如果与其他同事交流这些想法，那么可能会听到更多关于设计系统的方法。然而，一旦决定使用某种方法，那么后面如果要更改最初的决定将付出高昂的代价。重构将教会读者如何高效地修改代码，这样将把因为这些更改而产生的影响保持在最低水平。此外，重构还将帮助把设计视为在项目任何阶段都可以予以处理的问题，而不是从一开始就一成不变的。事实上，可以通过十分灵活的方式处理设计问题。