

C#

敏捷开发实践

【英】Gary McLean Hall 著 许顺强 译

拥抱敏捷，编写自适应代码
轻松应对恼人的需求变更



Adaptive Code via C#

Agile coding with design patterns and SOLID principles



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

C# 敏捷开发实践

【英】Gary McLean Hall 著 许顺强 译

Adaptive Code via C#

Agile coding with design patterns and SOLID principles



人民邮电出版社
北 京

图书在版编目 (C I P) 数据

C#敏捷开发实践 / (英) 加里·麦克莱恩·霍尔
(Gary McLean Hall) 著 ; 许顺强译. -- 北京 : 人民邮
电出版社, 2016. 7

(图灵程序设计丛书)

ISBN 978-7-115-42789-2

I. ①C… II. ①加… ②许… III. ①C语言—程序设
计 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第139459号

内 容 提 要

本书共分为敏捷基础、编写 SOLID 代码和自适应实例三大部分, 将理论与实践相结合, 介绍了当前使用 Microsoft .NET Framework 进行 C# 编程的最佳实践, 详尽探讨了 C# 开发人员如何应用 Scrum 等敏捷方案实现高质量、自适应的代码, 并给出大量代码示例, 是 .NET 中高级程序员进阶的实用指南。

本书的读者对象为有一定经验的 .NET 开发人员。

◆ 著 [英] Gary McLean Hall

译 许顺强

责任编辑 朱 巍

执行编辑 杨 琳 赵瑞琳

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 22

字数: 533千字

2016年7月第1版

印数: 1-3 000册

2016年7月北京第1次印刷

著作权合同登记号 图字: 01-2015-2390号

定价: 69.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

版权声明

Authorized translation from the English language edition, entitled *Adaptive Code via C#: Agile coding with design patterns and SOLID principles* by Gary McLean Hall, published by Pearson Education, Inc., publishing as Microsoft Press. Copyright © 2014 by Gary McLean Hall.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2016 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Pearson Education Inc. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

谨以此书献给 Amelia Rose。

译者序

翻译这本书算是圆了我学生时期的一个心愿：在一本纸质书的封面上印上自己的名字。作为一名 IT 技术人员，总是感觉自己的时间不够用，因为技术永无止境，让我从不敢懈怠。以前阅读过许多“大侠”翻译的不少书籍，但对于译者的辛酸和困难都没有太大的感触。直到自己开始实际动手翻译时，才发现整个过程中心情都是忐忑不安的，担心自己的翻译不能清楚传达原书作者的真实意图，害怕自己的译文会让读者觉得乏味，于是，总是会对不满意的翻译片段心生焦虑，也尽了自己最大努力对译文的选词和顺序进行反复的推敲和斟酌。当然，由于自身能力和精力有限，相信大家肯定会找出翻译中需要改进的地方，在此先行感谢大家的热心指正。

原书内容的精彩我就不多赘述。本书主要针对 C# 程序员，基于敏捷方法论，介绍使用 Microsoft .NET Framework 进行 C# 编程的当前最佳实践，其中包括了从敏捷项目过程到代码编写的理论和实践的详细讲解。对于那些需要实操指导的读者来说，几乎全部内容都可以直接应用在实际敏捷项目的管理和编码活动当中。如果你是一名初学者，可以在本书中学习到使用 C# 进行敏捷开发的常见模式和实践，明辨其优劣，让自己走在正确的方向上，为后续的能力提升打下良好的基础。如果你是一名中级开发人员，可以在本书中学习到业界的最佳实践，了解各种实践组合，对 SOLID 原则获得深入的理解，并完全认识到其在实际代码开发中带来的益处。如果你是一名高级开发人员，毫无疑问，你将获益最多。本书提供了大量设计模式、SOLID 原则、单元测试、重构等理论的示例，将理论与实践关联起来，让你可以直接拿来应用于工作之中。

当然，尽信书不如无书，相信有读者会对书中所讲并不完全赞同，但表达意见的前提是首先要理解书中讲解的本意。技术皆有优劣，作为 IT 技术人员，切忌用个人主观情绪来表达对技术观点的不满。我有时会听到周围有人说：“我就是看着不爽！”他们习惯了一刀切，也不明白一句古话：“三人行，必有我师焉。”原书作者 Gary 已经在全书技术理论和实例讲解过程中很好地穿插了优缺点和应用场景的讨论。认真读完一本技术书的收获应该是：明悉技术的概念和原理，了解其优缺点，并且知道其适用场合；如果还能在实践中针对缺点提出改进，那就再好不过了。

许顺强

2016 年 3 月 21 日

前言

本书英文书名中首先提到的一个关键术语是自适应代码 (Adaptive Code)，这一关键术语很好地诠释了应用本书中介绍的基本原则能够达成的效果：无需大量返工，代码即可自动适应后续新的需求和无法预见的场景。本书旨在将使用 Microsoft .NET Framework 进行 C# 编程的当前最佳实践囊括于一卷之中。尽管其他书中也会涵盖本书中的某些内容，但这些书要么偏重于讲解理论，要么就不是特定于 .NET Framework 开发的。

编写代码不能急于求成。与阻碍变化的代码库相比，如果你的代码具有自适应能力，你就能更加快速、轻松地对其进行更改，并且不会引发多少错误。相信大家都知道，对于需求，不变的主题就是变化。因此管理需求变更是软件项目成败的一个关键因素。面对需求变更，开发人员可以有很多种应对方法，这些方法可以归类到下面要讲述的两种方法论。这两种方法论的主旨几乎是截然相反的，特别是在变更处理的连续性上。

第一种是瀑布方法论。这种方法论要求开发人员必须遵循严格的流程。应用这种方法论的项目中，大到要遵循的开发流程，小到要实现的类型设计都很不灵活，甚至几乎与 50 年前用穿孔卡片进行编程开发一样死板。所有瀑布方法论都在竭尽全力确保软件很难被自由更改。软件开发被划分为分析、设计、实现和测试几个显著不同的阶段，而且整个过程是单向的。一旦进入实现阶段，用户就很难去变更需求，或者说至少要付出昂贵的代价才能变更需求。当然，代码也无需为需求变更作任何准备，因为整个瀑布流程几乎不提供任何其他选项。

第二种是敏捷方法论。它不仅仅是另一种选择，而且是对瀑布方法论的一种彻底推翻。敏捷流程的主旨就是拥抱变化，它被看作是客户和开发者之间的一个必要的联系纽带。如果客户想要对自己付费的产品进行某些改变，就应该把时间和资金的代价与需求的变更关联起来，而不是直接把变更加入流程中正在进行的阶段。软件工程的基础是源代码，相对于物理工程，它具有更好的可塑性。建造一座房子的过程就是使用水泥把砖逐块粘合在一起，所以改变房子设计的代价就很自然地与房子的建造完成度直接相关了。假设工程尚未开始，只有设计蓝图，那么变更设计的代价相对来说就会很低。如果已经装好了窗户，布好了电线、管线，此时再想把楼上的浴室改到楼下的厨房旁，那代价就会非常高了。软件产品的源代码具有良好的可塑性，因此移动特性和修改用户界面的导航看起来不应该有很高的代价，但不幸的是，事实并不总是如此。单单时间成本就经常不允许在软件产品中进行这样自由的变更。在我看来，这主要就是因为代码缺乏对需求变更的自适应能力。

本书将通过一些实际的例子，为大家演示和讲解敏捷流程以及如何编写自适应代码。

本书面向的读者

本书的意图是要把理论与实践关联起来。如果你是经验丰富的高级开发人员，想要找一些设计模式、SOLID 原则、单元测试、重构等理论的示例，那么这本书就是为你而作。

如果你是具有一定经验和能力的中级开发人员，想要学习业界的最佳实践，了解它们是如何配合使用的，或者你对现在的业界最佳实践组合有疑问，都可以从这本书中获益，因为现实的项目开发中很难找到简单且容易理解的实例或者理论。开发人员对大多数 SOLID 原则已经有所了解，但是对于其中比较复杂的开放与封闭原则（第 6 章）和 Liskov 替换原则（第 7 章）的理解还不够充分。即使是经验丰富的开发人员，有时也无法完全认识到依赖注入（第 9 章）给代码开发带来的好处。与此类似，接口（第 3 章）能给代码带来的适配灵活性也经常被忽视。

如果你是刚刚入门的初级开发人员，读完这本书，也会有所收获。你可以学习到常见的模式和实践，并且知道哪些方面是好的，哪些方面从长期来看是不好的。我见到的软件开发实习生所写的代码有很多共同点，到处都可以看到随从反模式（第 2 章）和服务定位器反模式（第 9 章）的代码。通常，实习生已经具备了很多方面的软件开发技能，要把他变成一个重量级的开发人员，只需要在正确的方向上推他们一把即可。本书也提供了多种可选的实践方案，并对其优缺点进行了详细解释。

阅读本书的前提条件

要阅读这本书，你应该具备一些在语法上与 C# 类似的编程语言（比如 Java 或 C++）的实战经验，也应该精通条件分支、循环和表达式等核心过程编程概念。此外，还应该有使用类进行面向对象开发的经验，并且对接口的概念有所了解。

本书不适合哪些人

如果你刚刚开始学习编程开发，那么本书并不适合你，因为书中涉及一些高级开发话题，需要你对基本的编程开发概念有深入的理解。

本书结构

本书共分为三个部分，每一部分都以上个部分为基础。尽管如此，也可以从任何一个部分开始阅读本书。每个章节都详细讲解了一个完整的主题，并在适当的地方包括了指向其他章节的交叉索引。

第一部分 敏捷基础

这个部分会讲解如何以自适应方式开发软件的基础概念，其中包括业界有名的敏捷流程 Scrum，该流程要求代码具有自适应变更的能力。这一部分的所有章节都围绕接口、设计模式、

重构和单元测试进行详细的讲解。

□ 第1章 Scrum 介绍

这一章是本书的开篇，首先介绍一种业界知名的敏捷项目管理方法论 Scrum，然后详细介绍 Scrum 项目中相关的工作、角色、度量标准和阶段的概念，最后为大家展示如何在敏捷环境下组织资源和代码。

□ 第2章 依赖和分层

这一章将引领你一起探索依赖和架构分层。代码要做到自适应，前提是解决方案的结构允许这样做。首先讲解三种不同类型的依赖：第一方、第三方和框架；然后讲解如何从反模式（应该避免的）到模式（应该使用的）来管理和组织所有的依赖关系；最后会介绍一些高级主题供你进一步阅读，比如面向切面编程和非对称分层等。

□ 第3章 接口和设计模式

在现代.NET 应用的开发中，接口几乎无处不在，但是它们也经常被滥用、误解和错用。这一章将首先通过一些常见和实用的设计模式来展示接口的多种用途；然后阐明除了使用接口进行简单的抽象外，还能够以多种不同的方式使用接口来解决同一个问题。如果能够流利地使用开发者武器库中的混合类型、鸭子类型和流接口，将更能体会到接口的强大用途。

□ 第4章 单元测试和重构

单元测试和重构正在成为开发人员必备的两个实战技能，只有同时应用这两个技能，才能编写出自适应代码。没有完备的单元测试，重构动作肯定会造成很多错误；没有重构，代码则会变得臃肿、僵化且难以理解。这一章会从一个十分简单的单元测试示例开始讲解，然后扩展讲解更高级、实用的模式和实践，比如流断言、测试驱动开发和模拟。本章还提供了真实的重构示例来讲解如何改善源代码的可读性和可维护性。

第二部分 编写 SOLID 代码

这一部分以第一部分为基础，每一章都会专门讲解 SOLID 中的一个原则。这些章节不会单单讲解为什么要使用这些原则，还有详细的实战示例来讲解如何在编码实现中使用这些原则。每一章都会提供一个实际项目中的例子来展示 SOLID 原则的作用。

□ 第5章 单一职责原则

这一章会为开发人员展示如何使用修饰器和适配器模式来实践单一职责原则。应用这个原则后，类的数目会增加，但是每个类的规模会变小。相对于那些功能繁多的大型类，小型类可集中解决一个大型问题中的一小部分。聚合使用这些小型类会比使用单个大型类更强大、更灵活。

□ 第6章 开放与封闭原则

这一章要讲解的是开放与封闭原则，它的概念非常简单，但是它对代码有着举足轻重的影响。它负责确保那些遵守所有 SOLID 原则的代码不会被改动，而只是添加新代码。这

一章还会讨论跟开放与封闭原则相关的可预测变化的概念，并且会探讨它如何能够识别后续自适应扩展的切入点。

□ 第7章 Liskov 替换原则

这一章会展示在代码中应用 Liskov 替换原则所带来的好处，特别要提到的是，Liskov 替换原则有助于确保开放与封闭原则的应用，并避免代码改动所带来的副作用。这一章还会介绍代码契约，它包括前置条件、后置条件和数据不变式三个要素，可以通过代码契约工具来确保代码满足它们。此外，这一章还描述了一些子类型原则，比如协变、逆变和不变性原则，并介绍了违背这些原则会带来的不良影响。

□ 第8章 接口分离原则

在实际的代码中，接口和类的问题类似，它们的规模通常太过庞大。接口分离原则是一个经常被忽视却简单有效的实践原则。这一章会展示将接口规模限制到足够小，以及配合使用这些小型接口能够带来的好处。此外还会探究可能会促使接口分离的各种不同的原因，比如客户端需求和架构需求。

□ 第9章 依赖注入原则

这一章的核心是依赖注入，它能够将本书中讲解到的所有特性结合为一个整体。依赖注入真的非常重要，没有它，很多其他原则都无法起作用。这一章会详细介绍依赖注入并比较实现依赖注入的不同方法。这一章还会讨论如何使用控制反转容器来管理对象的生命周期，以避免服务定位器等反模式；此外，还会讨论如何识别组合根和解析根。

第三部分 自适应实例

这一部分以一个示例应用为主线，将本书的剩余内容组织到一起。尽管这些章节中有很多代码，但我也提供了很丰富的注释和讲解。因为本书的讲解背景是敏捷环境，所以下面这些章节会按照 Scrum 的冲刺进行组织，并且所有工作都是由积压工作项和客户变更请求而来。

□ 第10章 自适应实例简介

这个部分要实际开发的示例应用是一个基于 ASP.NET MVC 5 开发的在线聊天应用。这一章会先详细描述这个应用，并给出一个简要的设计作为后续架构的指导，最后还给出了产品积压工作上所有特性的解释。

□ 第11章 自适应实例冲刺 1

这一章介绍如何使用测试驱动开发方法来开发示例应用的首要特性，包括查看和创建聊天室和消息。

□ 第12章 自适应实例冲刺 2

这一章讲解客户对应用提出需求变更，以及整个开发团队如何通过自适应代码来适应这些必然会发生的变更。

附录 自适应工具

附录简要介绍如何使用 Git 源代码控制从 GitHub 上下载代码, 以及如何使用 Microsoft Visual Studio 2013 编译下载的代码。请注意, 附录不是完整的 Git 使用说明, 你可以在网上找到很多非常详尽的资料, 比如 Git 的官方教程: <http://git-scm.com/docs/gittutorial>。

通过快速 Web 搜索可以找到其他来源。

附录还会简要介绍其他的一些开发工具, 比如持续集成和开发环境。

本书约定

本书中有若干反复出现的约定用法, 你可以在微软出版社的出版物中找到标准的解释, 我也在这里先给出一些简要的解释。

代码清单

书中代码清单会在适当的地方出现, 相关的代码会在同样的背景块中。比如下面的代码清单 I-1。

代码清单 I-1 这是一个代码清单, 在书中会经常出现

```
public void MyService : IService
{
}
}
```

只要看到代码清单, 你都应该关注其中特定的一部分代码。比如, 当对上一份示例代码进行改动时, 与改动相关的代码就会加粗显示。

阅读辅助和补充信息

阅读辅助主要提供一些与主题相关的边栏, 比如注意或者警告, 而补充信息则是提供一些更进一步扩展主题的信息。下面是一些示例。

注意 这是阅读辅助, 其中包括与主要内容相关的小信息, 不过具有额外的重要性。

这是补充信息

尽管已经尽量缩短篇幅, 但是补充信息通常包含与主要话题不太相关的较长讨论。

图片

有时候,无论文字解释有多么形象,还是不足以表达确切的含义。这时候,就必须提供图片了。书中所有使用 Microsoft Visio 2013 创建的图表都是黑白色的,目的是为用户提供清晰的说明。同样,截图也是在高对比度主题下获取的。

系统要求

为了使用书中提供的代码示例,需要以下硬件和软件。

- ❑ 安装了以下任意一个操作系统: Windows XP SP3 (Starter Edition 除外)、Windows Vista SP2 (Starter Edition 除外)、Windows 7、Windows Server 2003 SP2、Windows Server 2003 R2、Windows Server 2008 SP2 以及 Windows Server 2008 R2。
- ❑ 安装了 Visual Studio 2013 的任意版本 (如果你使用的是 Express Edition 系列产品,可能需要下载几个安装包)。
- ❑ 安装了 Microsoft SQL Server 2008 Express Edition 或者更高版本 (2008 或 R2 版本), 以及 SQL Server Management Studio 2008 Express 或更高版本 (Visual Studio 安装包中已经包括, Express Edition 则需要单独下载)。
- ❑ 处理器频率不低于 1.6 GHz (推荐 2 GHz)。
- ❑ 内存大小不低于 1 GB (32 位操作系统) 或 2 GB (64 位操作系统)。如果运行虚拟机系统或者使用 SQL Server Express 版本,则另外需要 512 MB 内存,更高的 SQL Server 版本需要更多内存。
- ❑ 硬盘可用空间不低于 3.5 GB。
- ❑ 硬盘转速不低于每秒 5400 转。
- ❑ 显卡必须支持 DirectX 9, 并且支持 1024×768 或者更高分辨率的显示。
- ❑ DVD 光驱 (如果需要从 DVD 安装 Visual Studio)。
- ❑ 能够连接互联网以便下载软件或者代码示例。

基于不同的 Windows 配置,你可能需要本地管理员权限才能安装或配置 Visual Studio 2013 和 SQL Server 2008 等产品。

下载示例代码

我会尽力保证书中的代码片段都是一个可独立运行的应用或者单元测试中的一部分。我使用 MSTest 写了很多简单的单元测试,因此无需使用其他额外的测试器,另外我也使用 NUnit 写了一些更复杂的单元测试。我使用 Visual Studio 2013 Ultimate 编写了书中所有的代码。尽管一些代码是用 Visual Studio 2013 Ultimate 预览版编写的,但是它们都能够使用正式版成功编译和测试。我尽量不使用 Visual Studio 2013 Express 版本之外的特性,但是有些主题的代码必须使用,因此你可能需要安装一个付费版本来运行部分代码。

代码可以从 GitHub 下载，网址是：http://aka.ms/AdaptiveCode_CodeSamples。

附录包含了 Git 的使用方法简介。

我很乐意看到你的留言，下面是我的 WordPress 博客网址：<http://garymcleanhall.wordpress.com>。

致谢

本书的作者署名只有我自己，这肯定是不准确的。因为如果没有大家给我提供各方面的帮助，我根本无法完成这本书。

感谢我的妻子 Victoria，她是这本书成为可能的关键。这不是空话，这就是事实。

感谢我的女儿 Amelia，她每天的表现都十分完美。

感谢我的母亲 Pam，她帮助我校对，感谢她对我毫无保留的鼓励。

感谢我的父亲 Les，感谢他的所有付出。

感谢我的兄弟 Darryn，感谢他给我源源不断的指导。

感谢来自 Online Training Solution 公司的 Kathy Krause，她的努力让这本书读起来舒服流畅。

感谢 Devon Musgrave 的耐心帮助。

勘误、更新和相关支持

我们已经尽了最大的努力来确保书中内容以及相关材料是正确的。你可以从下面的网址得到本书的更新，其中包括完整的勘误表：<http://aka.ms/Adaptive/errata>。

如果你发现了勘误表之外的新错误或者不当之处，也请在上面的网址提交。

如果你需要额外的支持，可以发送电子邮件给微软出版社支持中心：mspinput@microsoft.com。

此外，有关微软软件或者硬件产品的支持不能通过上面的网址获得，请访问以下网址：<http://support.microsoft.com>。

微软出版社的免费电子书

微软出版社有很多免费的电子书，深入讲述了很多技术主题。这些电子书以 PDF、EPUB 和 Mobi（Kindle 电子书阅读器支持的格式）等格式供读者下载：<http://aka.ms/mspressfree>。

常去看看，你会发现很多新的电子书。

期待你的反馈

在微软出版社看来，读者的满意度始终是第一位的，读者的反馈是最有价值的资产。请在下面的网址将你对这本书的看法提交给我们：<http://aka.ms/tellpress>。

我们知道大家都很忙，所以尽量只设计了少量的简单问题。大家的回答会被直接发送给微软出版社的编辑（不需要个人信息）。谢谢大家的热心反馈。

保持联系

让我们保持联系，下面是我们的 Twitter 网址：<http://twitter.com/MicrosoftPress>。

电子书

扫描如下二维码，即可购买本书电子版。



目 录

第一部分 敏捷基础

第 1 章 Scrum 介绍	3
1.1 Scrum 与瀑布	4
1.2 角色和职责	6
1.2.1 产品负责人	7
1.2.2 Scrum 主管	7
1.2.3 开发团队	8
1.2.4 “猪”和“鸡”	8
1.3 工件	9
1.3.1 Scrum 面板	9
1.3.2 图表和度量标准	20
1.3.3 积压工作	24
1.4 冲刺	25
1.4.1 发布计划会议	26
1.4.2 冲刺计划会议	26
1.4.3 每日站立会议	28
1.4.4 冲刺演示会议	29
1.4.5 冲刺回顾会议	30
1.4.6 Scrum 日历	31
1.5 Scrum 和敏捷的问题	32
1.6 总结	36
第 2 章 依赖和分层	37
2.1 依赖的定义	38
2.1.1 一个简单的例子	38
2.1.2 使用有向图对依赖建模	44
2.2 依赖管理	48
2.2.1 实现与接口	48
2.2.2 new 代码味道	49

2.2.3 对象构造的替代方法	52
2.2.4 随从反模式	54
2.2.5 阶梯模式	56
2.2.6 依赖解析	57
2.2.7 使用 NuGet 管理依赖	67
2.3 分层	70
2.3.1 常见的模式	71
2.3.2 纵切关注点	76
2.3.3 非对称分层	77
2.4 总结	79
第 3 章 接口和设计模式	80
3.1 接口是什么	80
3.1.1 语法	80
3.1.2 显式实现	83
3.1.3 多态	87
3.2 自适应设计模式	88
3.2.1 空对象模式	88
3.2.2 适配器模式	94
3.2.3 策略模式	96
3.3 更多形式	98
3.3.1 鸭子类型	98
3.3.2 混合类型	102
3.3.3 流接口	106
3.4 总结	108
第 4 章 单元测试和重构	109
4.1 单元测试	109
4.1.1 布置、动作和断言	110
4.1.2 测试驱动开发	113
4.1.3 更复杂的测试	118

4.2 重构	131
4.2.1 更改已有代码	131
4.2.2 一个新的账户类型	139
4.3 总结	143

第二部分 编写 SOLID 代码

第 5 章 单一职责原则

5.1 问题描述	147
5.1.1 重构清晰度	150
5.1.2 重构抽象	153
5.2 单一职责原则和修饰器模式	160
5.2.1 复合模式	162
5.2.2 谓词修饰器	165
5.2.3 分支修饰器	168
5.2.4 延迟修饰器	169
5.2.5 日志记录修饰器	170
5.2.6 性能修饰器	172
5.2.7 异步修饰器	175
5.2.8 修饰属性和事件	177
5.3 用策略模式替代 switch 语句	178
5.4 总结	180

第 6 章 开放与封闭原则

6.1 开放与封闭原则介绍	181
6.1.1 Meyer 的定义	181
6.1.2 Martin 的定义	181
6.1.3 缺陷修复	182
6.1.4 客户端感知	182
6.2 扩展点	183
6.2.1 没有扩展点的代码	183
6.2.2 虚方法	184
6.2.3 抽象方法	184
6.2.4 接口继承	185
6.2.5 “为继承设计或禁止继承”	186
6.3 防止变异	186
6.3.1 可预见的变化	187
6.3.2 一个稳定的接口	187
6.3.3 足够的自适应能力	187
6.4 总结	188

第 7 章 Liskov 替换原则

7.1 Liskov 替换原则介绍	189
7.1.1 正式定义	189
7.1.2 Liskov 替换原则的规则	190
7.2 契约	190
7.2.1 前置条件	192
7.2.2 后置条件	193
7.2.3 数据不变式	194
7.2.4 Liskov 契约规则	195
7.2.5 代码契约	201
7.3 协变和逆变	208
7.3.1 定义	208
7.3.2 Liskov 类型系统规则	213
7.4 总结	216

第 8 章 接口分离原则

8.1 一个分离接口的示例	217
8.1.1 一个简单的 CRUD 接口	217
8.1.2 缓存	223
8.1.3 多重接口修饰	226
8.2 客户端构建	228
8.2.1 多实现、多实例	229
8.2.2 单实现、单实例	231
8.2.3 超级接口反模式	232
8.3 接口分离	233
8.3.1 客户端需要	233
8.3.2 架构需要	239
8.3.3 单方法接口	243
8.4 总结	244

第 9 章 依赖注入原则

9.1 简单的开始	245
9.1.1 任务列表应用	248
9.1.2 对象图的构建	250
9.1.3 控制反转	254
9.2 比较复杂的注入	267
9.2.1 服务定位器反模式	267
9.2.2 非法注入	270
9.2.3 组合根	272
9.2.4 约定优于配置	277

9.3 总结	280
--------------	-----

第三部分 自适应实例

第 10 章 自适应实例简介	284
----------------------	-----

10.1 Trey Research 公司	284
10.1.1 团队	284
10.1.2 产品	286
10.2 最初的产品积压工作	287
10.2.1 从描述中挖掘故事	287
10.2.2 故事点估算	288
10.3 总结	292

第 11 章 自适应实例冲刺 1	293
------------------------	-----

11.1 计划会议	293
11.2 “我想创建多个房间以对会话进行分类”	295
11.2.1 控制器	295
11.2.2 房间存储库	299
11.3 “我想查看代表会话的房子的列表”	303
11.4 “我想查看发送到一个房间内的消息”	307
11.5 “我想给房间内的其他成员发送纯文本消息”	309
11.6 演示会议	311

11.7 回顾会议	311
11.7.1 什么做得比较好	312
11.7.2 什么做得不太好	312
11.7.3 什么需要改变	313
11.7.4 什么需要保持	314
11.7.5 遇到了什么意料之外的事情	314
11.8 总结	315

第 12 章 自适应实例冲刺 2	316
------------------------	-----

12.1 计划会议	316
12.2 “我想发送正确格式化的标记”	317
12.3 “我想过滤消息内容以确保它是适合发表的”	321
12.4 “我想同时服务数百个用户”	323
12.5 演示会议	325
12.6 回顾会议	326
12.6.1 什么做得比较好	326
12.6.2 什么做得不太好	327
12.6.3 什么需要改变	327
12.6.4 什么需要保持	327
12.6.5 遇到了什么意料之外的事情	327
12.7 总结	328

附录 自适应工具	329
----------------	-----