Addison Wesley

软件工程系列教材

Java与UML面向对象程序设计

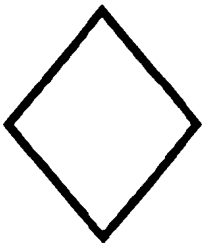
The Essence of Object-Oriented Programming with Java™ and UML

Bruce E.Wampler 著

王海鹏 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

软件工程系列教材



Java与UML面向对象程序设计

Bruce E. Wampler 著

王海鹏 译

人民邮电出版社

图书在版编目(CIP)数据

Java 与 UML 面向对象程序设计/（美）万普勒（Wampler，B.D.）著;王海鹏译.—北京：人民邮电出版社，2002.10
软件工程系列教材
ISBN 7-115-10603-7

I. J… II. ①万… ②王… III. ①JAVA 语言—程序设计—教材②面向对象语言，UML—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 069679 号

软件工程系列教材

Java 与 UML 面向对象程序设计

◆ 著 Bruce E .Wampler
译 王海鹏
责任编辑 俞 彬

◆ 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
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092 1/16
印张：14
字数：329 千字 2002 年 10 月第 1 版
印数：1-4 000 册 2002 年 10 月北京第 1 次印刷

著作权合同登记 图字：01－2002－3745 号
ISBN 7-115-10603-7/TP・3068
定价：28.00 元

内容提要

本书旨在介绍使用Java和UML开发面向对象的软件所必需的知识，并将伴您走上使用Java进行真正面向对象软件开发的道路。

本书共分12章。本书的前4章介绍了面向对象的精髓，第1章是关于对象的简介。第2章介绍了面向对象的基本概念以及UML的一些基本知识。第3章阐述了如何使用Java来编写面向对象程序。第4章介绍面向对象的分析和设计。第5章使用这些面向对象观点分析了图形用户界面（GUI）和Java Swing类库。第6章以一个小规模的Java应用程序为实例，将前5章的内容结合到一起。

接下来的章节让您对面向对象编程的实践方面有较好的了解。第7章介绍了设计模式。第8章介绍软件重构。第9章简要介绍针对大规模和小规模的面向对象软件项目以及当前的一些主要开发方法。第10章介绍当前一些面向对象软件开发的工具软件。第11章给出了作者为开发更好的软件而提出一些指导意见。最后，第12章提供了关于面向对象软件和Java方面更多的学习资源。

本书强调理论和设计相结合，重视对软件开发方法学有指导作用的重要概念。本书可作为高等学校计算机科学系及软件学院高年级学生和研究生的教科书，也可作为从事软件开发的管理者、系统分析员、程序员在学习面向对象程序设计时的参考书。

前　言

为何要读这本书

本书旨在介绍使用 Java 和 UML 开发面向对象的软件所必需的知识。读完本书后，您将能够较好地理解面向对象的软件开发，并能够回答下列问题：

- 什么是面向对象?
- 什么是 UML?
- 什么是面向对象的分析和设计（OOAD）?
- 如何进行 OOAD?
- 什么是面向对象开发方法学?
- 如何使用 Java 来编写真正的面向对象程序?
- 什么是 Swing? 如何使用它来写面向对象的图形用户界面?
- 什么是设计模式?
- 什么是重构?
- 使用哪些工具来编写面向对象的程序?
- 书写优质代码有哪些原则?
- 想进一步学习面向对象，需要继续阅读哪些资料?

本书针对的读者

本书针对那些知道一些 Java 编程的基本知识，希望理解面向对象软件开发基础的程序员。如果您是一个编程新手，上过一些 Java 的课程，您可能已开始觉得 Java 不错了。现在您准备获得真正的 Java 面象对象编程的好处，这本书将帮助您实现这一目标。

如果您是一位有经验的程序员，希望从使用老式的过程编程语言转为用 Java 开发面向对象的系统，本书也是为您而写。本书将伴您走上真正面向对象软件开发的道路。如果您手边有 Java 手册作为快速参考，您将能够通过本书提供的例子学习 Java 最重要的一些方面。

然而，本书不该是您读的最后一本关于面向对象、UML 或 Java 的书。本书使您对对象有了基本理解，让您能够进一步阅读关于这个主题更高级的、更细节化的书籍，以达到更高的目标。

各章简介

第 1 章是关于对象的简介，以及面向对象软件开发的好处。

第 2 章介绍了面向对象的基本概念。面向对象有很多的重要概念，当然，也有它自己专有的词汇。理解主要的概念并熟悉那些特别的词汇是很重要的。即使您已熟悉一些面向对象的概念，也应该通过本章复习一下。本章还介绍了 UML 的一些基本知识。

第 3 章介绍如何使用 Java 来编写面向对象程序。它并不是一个 Java 指导手册，而是着重阐述使用 Java 来实现面向对象概念。本章的前半部分介绍一些 Java 的基本概念，后半部分介绍一些更高级的主题，例如对象生命周期、对象拷贝，以及使用类和对象的其他一些重要概念。

第 4 章介绍面向对象的分析和设计（OOAD）。本章并不是针对任何特定的 OOAD 开发方法，而是介绍对所有方法都重要的一些基本要素。

本书的前 4 章介绍了面向对象的精髓，第 5 章使用这些面向对象观点分析了图形用户界面（GUI）和 Java Swing 类库。这种以面向对象方式来介绍 Swing 的方法与其他一些 Swing 指导手册有些不一样。

第 6 章以一个小规模的 Java 应用程序为实例，将所有的东西结合到一起。使用第 4 章介绍的 OOAD 基本概念来设计应用程序，使用第 3 章和第 5 章介绍的 Java 和 Swing 来实现。

本书剩余章节的目的是让您对面向对象编程的一些实践方面有较好的了解。第 7 章介绍了设计模式。设计模式是近年来发展起来的一种方法，通过使用以前逐步积累的软件设计模式，使得新软件的设计变得更容易。第 8 章介绍软件重构，这是一种很好的面向对象方法，目的是修改和增强已有的软件。第 9 章针对大规模和小规模的面向对象软件项目，简要介绍当前的一些主要开发方法。第 10 章介绍当前一些面向对象软件开发的工具软件。第 11 章给出了我本人对开发出更好的软件的一些建议。第 12 章提供了关于面向对象软件及 Java 方面更多的学习资源。

关于作者

我写第一个程序是在 30 年前，从那时起，就一直从事软件开发工作直到现在。我写的大多数软件在 PC 市场上出售，这意味着我的代码必须完成有用的工作，包含尽量少的错误，然后交给其他人继续开发。这就要求我必须有较高的开发效率并且拥有实际开发经验。长期以来，我一直希望与其他程序员分享我的实践经验。

我都有些什么样的经验呢？1979 年当我从犹他大学获得计算机科学博士学位后，就到 Sandia 国家实验室从事安全软件方面的工作。然而，我发现当时出现的个人计算机更令人兴奋，于是我离开了 Sandia 实验室，开办了一个小软件公司，写了一个最早运行在 PC 上的拼写检查软件。我的下一步是编写第一个基于 PC 的语法和写作风格检查软件。

后来我卖掉了我的公司，开始在新墨西哥大学教授计算机科学。这一直持续到 1997 年，其间也有一些时间并非全职。但是我还是离不开 PC 事业。我继续写我的语法检查软件，1985 年和一些朋友一起在旧金山开了一个新的 PC 软件公司，这个名为 Reference Software International 的公司开发并销售 Grammatik 语法检查软件。我是公司的首席科学家，组建了一个相当大的软件开发团队来改进 Grammatik 和开发其他的参考软件产品。1992 年，WordPerfect

收购了 Reference 软件公司，我又回到新墨西哥大学教书，在那儿我第一次想写一本关于面向对象软件开发的书。

同时，我还设计并实现了一个开放源码的 C++ GUI 框架，称为 V。它是一个易于使用的框架，用于在 Windows 和 X 上开发简单的 GUI 应用程序。今天，它已被广泛使用。我还设计了 VIDE 自由软件编辑器和集成开发环境，同样被广泛使用。

这么多年来在我所见证的所有软件开发的进步中，使用 Java 和 C++来进行面向对象的程序设计应该是最重要的进步，因为它使得程序设计任务变得容易许多。虽然面向对象并不能解决所有软件开发的问题，但它使得软件的开发和长期的维护变得更容易，其结果是大大提高了编程的效率，所以花一些力气学习面向对象的软件开发是很有价值的。

本书的目的是向您介绍面向对象的精髓，使您不致被某种面向对象开发方法的所有细节或编程语言的细微差别所淹没。经过多年的程序设计教学工作和软件开发工程，我发现如果您一开始就较好地理解对象和使用对象来设计系统，学习 Java 或其他面向对象程序设计语言就会变得更容易。

我发现，仅仅因为程序员使用一种面向对象的程序设计语言并不能说明他们是在写优质的面向对象程序。没有对面向对象的深刻理解，是不可能利用它所带来的所有好处的，包括最重要的好处——易于开发和维护。

致谢

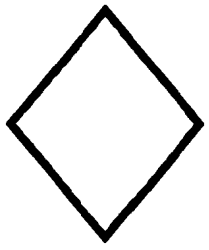
首先，我必须感谢我的家庭。在过去的一年里，当我在办公室里写作本书时，是我的家庭给了我理解和支持。我知道他们宁愿我有更多的时间和他们在一起，但写作本书是我多年来一直想做的事情。特别感谢我的儿子 Van，他为封面创作了一幅了不起的 Kokopelli 程序的图。

我还必须感谢 Addison-Wesley 出版社的编辑 Ross Venables，是他在我的网站上发现了这本书的早期版本并鼓励我把它变成一本完整的书。我也要感谢 Paul Becker，他参与了这个项目并看着它完成。

我要感谢所有帮助我把这本书变得更好的人，包括 Addison-Wesley 出版社的审阅者和编辑们，以及所有对我发表在网站上的早期草稿提供建议和反馈意见的人们。

Bruce E. Wampler
科罗拉多州，Glenwood Springs

目　录



第1章

对象、UML 和 Java

本书主要介绍面向对象（OO）软件开发。编写要交付给真人来使用的真正的面向对象软件不只是敲打出几行 Java（或 C++、Eiffel，或其他任何面向对象的程序设计语言）代码那么简单。说到底，面向对象的软件开发包括了完整的过程——分析问题、设计解决方案、编码实现以及长期的维护。面向对象的开发可以使所有程序变得更好，无论是小的基于 Web 的应用程序还是全尺寸的关键业务软件系统。

面向对象技术具备开发出伟大产品的潜力，但仅当它被用于一个完整的过程时，才具备这种可能性。今天，有一些小而敏捷的开发方法学适用于从两人到十来个人的软件开发团队，也有一些大规模的开发方法学适用于大的项目。这些方法学中的大多都采用了 UML（Unified Modeling Language，统一建模语言），或是从 UML 中吸取了营养。UML 是可以用于辅助设计任何 OO 系统的建模工具。但在您能理解和使用这些方法之前，您必须把您的思维从仅仅让程序能运行转变到面向对象地思考。

有人说，用任何程序设计语言都可以写出面向对象的程序（有人用 C 做到了这一点），但是使用真正面向对象的语言更容易做到这一点。仅仅因为您使用的是一种面向对象的程序设计语言，并不能说明您写的程序就是面向对象的。

如果与面向对象的分析和设计（OOAD）过程一同使用，面向对象的编程工作就能进行得更好。不首先经过分析和设计就试图写一个 OO 程序就好比不分析住宅的需求，也不进行设计和产生一套设计蓝图，就要建造住宅一样。建造完成时你可能会发现，虽说屋顶是比你的头顶要高，但是各个房间杂乱无章地排列着，有些房间可能根本就没有，而且整个建筑可能在第一次暴风雨来临时就砸在你的头上（图 1.1）。一个没经过一点 OOAD 而写成的 OO 程序，不论它是使用哪种程序设计语言编写的，可能看上去是能工作的，但是当您第一次试图进行修改时，极有可能出现的情况是程序中充满了毛病和断点。

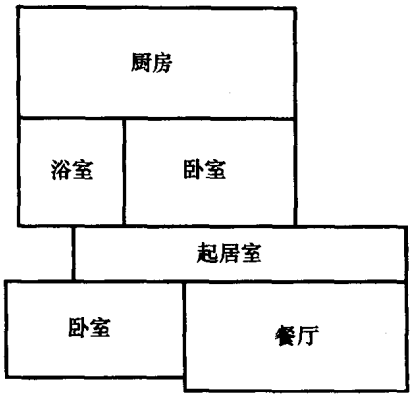


图 1.1 任意规划的住宅

1.1 面向对象技术

面向对象技术的核心是对象。一个对象几乎可以代表在程序中想模型化的任何东西。一个对象可以是一个雇员的模型，或是代表一个传感器，用户界面的一个窗口，一种数据结构，如一个链表，事实上它可以是任何东西。一种方式是把对象想象为一个黑盒子，它有一些按钮和一些指示灯（图 1.2）。它可以是一个电视机、一辆汽车，或是其他东西。要想使用这个对象，就需要知道按钮的功能，即按哪些按钮能让对象做哪些您想做的事，还需要知道那些指示灯说明了对象处在何种状态。那些盒子内部是如何组装的细节，在你使用它时是无需了解的。重要的是那个对象正确地实现了它的功能和职责。软件中的对象也没太大区别。软件中的对象有一组定义良好的方法来与外界交互，它可以提供自身的当前状态信息。在外界看来，它内部的表示方式、算法以及数据结构都被隐藏起来了。

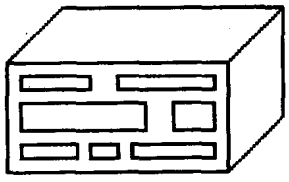


图 1.2 黑盒子

用最简短的术语来说，设计一个 OO 系统包括识别系统包含哪些对象、识别这些对象的行为和职责，以及对象之间是如何交互的。OO 可以产生优雅的和易于理解的设计，从而导致优雅的和易于理解的程序。不同的对象常常可以独立地实现和调试。已有的对象库可以很容易地重用并适应新的设计。最重要的是，一个好的 OO 程序是易于修改的，在程序修改和维护的过程中，能有效防止引入缺陷。

对于软件开发来说，面向对象的开发是一个重要的进步。尽管它可能并不是能解决所有软件开发问题的神奇的“银弹”，但它比其他的方法学要好。虽然其他的软件开发方法学，

如结构化的设计和编程，也有很多有效正确之处，其中许多也被OO开发所利用，但是面向对象的设计天生要更容易进行，也更易于长时间的维护。

1.2 面向对象程序设计语言

目前存在着多种面向对象程序设计语言，包括Smalltalk、Eiffel、C++、Objective C、Objective Pascal、Java、Ada，甚至有一种版本的Lisp也是面向对象的。在市场上占主导地位的有两种语言：C++和Java。

今天，Java作为一种脱颖而出的面向对象程序设计语言，已成为很多程序员和软件项目的选择。Java流行的一个原因就是World Wide Web和Java能够让小程序在任何有浏览器的计算机或操作系统上运行的能力。另一个原因是Java是一种优秀的程序设计语言。它是一种小巧、设计精良的语言，不仅可以用来写Web小程序，也可以用来在当今几乎任何一种计算机上开发全尺寸的程序。在早期，Java因为性能问题，推广受到一些阻碍，但如今这已不再成为问题。由于它是如此的优秀，世界各地的大学和院校纷纷采用Java作为教授计算机科学的主要程序设计语言。在计算机科学和编程的整个历史上，同一种程序设计语言既在教学上受到欢迎，又在工业界被广泛采用，这还是第一次。

C++也是一种被广为使用的程序设计语言。它仍是当今大多数计算机上核心应用程序（如电子表格和字处理程序）的首选程序设计语言。C++源自于C，因此就传承了在真实系统中完成真实任务的能力，而且对业已存在的C代码具有兼容性。然而C++的一个问题在于，它已经成长为一种巨大而且复杂的语言，难以全面掌握并达成相当的水平。

本书的绝大部分是关于面向对象程序设计的。也就是说，本书将重点放在适用于任何程序设计语言的面向对象程序设计通用原则上。但本书也将展示如何用Java将面向对象的设计变成真实的程序，重点将放在如何运用Java语言的能力来实现OO设计。这并不是一个关于Java的使用指南。我们假定您已经掌握了Java的基础知识，现在准备学习对象以及如何用Java来写出更好的程序。

1.3 面向对象的设计和UML

目前使用的面向对象开发方法学有许多种，每种都有其长处和弱处。老一点的、更为传统的方法学常被称为“重量级的”方法学，对于涉及成百上千人，开发期达数年的大的软件项目，这些方法学最有效。新一点的方法学，称为“轻量级的”或“敏捷的”方法学，更适合于小一些的项目。这些方法学中的大部分都还很新，正在经历标准化过程。

设计和开发方法学总是需要一个图形化的符号表示法来表达设计。过去，存在的一个主要问题就是每种主要的方法学都有自己的一套图形化符号表示法。随着UML作为一种标准符号表示法的出现，这种情况将得到改善。

UML源于20世纪90年代中期James Rumbaugh、Ivar Jacobson和Grady Booch（The Three Amigos，三个朋友）的工作。Object Management Group（www.omg.org）整理并形成了一个

UML 标准规范。OMG 是一个业界赞助的组织，致力于为面向对象开发社区提供与供应商无关的标准支持。UML 已经成为面向对象符号表示法事实上的标准。

设计 UML 的目的是将其用于讨论面向对象的设计。它的表现对象和对象关系的能力特别有用，将被用在贯穿本书的例子中。UML 的多种特征将根据需要予以介绍。

1.4 对象的回报

在软件开发活动中，面向对象技术可以导致巨大的回报。面向对象的设计通常比较简洁并且易于理解。一旦完成设计，常常可以分开来实现和测试单独的对象。一旦完成，每个对象会比较健壮并且无错。当修改系统时，现存的对象仍能继续工作。当改进现存的对象时，它们对外界的接口保持不变，所以整个系统能继续工作。正是这种易于改变的特性和健壮性，使得 OO 软件开发真正不同于其他开发方式，值得我们朝这个方向努力。

1.5 本章小结

- 面向对象技术是一种软件开发方式，能够得到设计精良的系统，既健壮又易于维护；
- UML 是一种图形化的表示手段，适用于设计和理解面向对象系统；
- Java 是一种优秀的面向对象程序设计语言，既适用于 web 小程序，也适用于非 web 的应用程序。