

高等学校规划教材

21

# UML基础与应用

王养廷 李磊 宋汉珍 编著

清华大学出版社



高等学校规划教材

# UML基础与应用

王养廷 李磊 宋汉珍 编著

清华大学出版社  
北京

## 内 容 简 介

本书从应用的角度介绍了UML统一建模语言的基本知识和应用技术。与传统的写法不同,本书从UML实际应用出发,从面向对象软件开发的角度组织内容,将知识点融入到实际项目的开发中,注重使用UML来解决实际应用问题。本书通过UML基础知识的学习、课后习题的练习、应用项目AutoWeight系统的开发这三个层次来帮助读者掌握知识点,第一个层次为学习和模仿,第二个层次为练习和深化,第三个层次为应用和提高。

本书浅显易懂,并结合实例和案例进行讲解,有助于读者在较短的时间内学会如何在软件开发过程中应用UML。本书既可以作为高等院校计算机专业的学生学习UML或面向对象程序设计的基础教材,也可作为从事计算机行业的技术人员学习UML的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

UML基础与应用/王养廷,李磊,宋汉珍编著. —北京:清华大学出版社,2006.6

(高等学校规划教材)

ISBN 7-302-12795-6

I. U… II. ①王… ②李… ③宋… III. 面向对象语言,UML—程序设计—高等学校—教材  
IV. TP312

中国版本图书馆CIP数据核字(2006)第033321号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 曾 妍

印 刷 者: 北京四季青印刷厂

装 订 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 15 字数: 308千字

版 次: 2006年6月第1版 2006年6月第1次印刷

书 号: ISBN 7-302-12795-6/TP·8145

印 数: 1~5000

定 价: 22.00元

# 前 言

## UML 基础与应用

UML 是一种定义良好、易于表达、功能强大且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。UML 支持面向对象的分析与设计,可以方便地与具体的软件开发过程相结合。

本书以 UML 应用和实例为主,由浅入深地向读者介绍了 UML 的基础知识以及如何在实际项目中应用 UML 技术。全书共 12 章,分成两个部分。第一部分包括第 1~6 章,主要介绍 UML 的基础知识。第 1 章简要介绍了 UML 的基本概念和软件过程,第 2~6 章详细介绍了 UML 的用例图、类图、对象图、包图、顺序图、合作图、状态图、活动图、构件图和部署图,并以大家熟悉的“学生选课系统”和“图书管理系统”为例,给出了 UML 10 种图的简单应用。第二部分包括第 7~11 章,主要介绍如何在实际的项目中应用 UML,以一个实际的项目“AutoWeight 系统”为例,讲述了 UML 在软件项目的需求分析、系统设计、系统实现、集成和交付这几个阶段的应用。第 12 章主要介绍了 Rational 公司 Rose 工具的使用,以及如何使用 Rose 来描述 AutoWeight 系统实例。在本书的附录 A 给出了常见的 UML 工具,并简单说明了各种工具的主要特点及支持的平台,便于读者今后选择。附录 B 给出了 UML 中一些常见术语的中英文对照表,以便读者在今后查阅英文书籍时使用。附录 C 给出了 UML 中出现过的所有的图符,以方便查询。

本书可作为高等院校计算机专业的学生学习 UML 或面向对象程序设计的基础教材,也可作为从事计算机行业的技术人员学习 UML 的参考书。

本书由王养廷和李磊共同编写,其中第 1、2、7、8、9、10、11 章由王养廷负责编写,第 3、4、5、6、12 章由李磊负责编写,全书由宋汉珍负责统稿和主审。

由于作者的水平有限,加之时间比较仓促,书中难免存在错误之处,恳请广大读者给予谅解。如果读者有问题需要与作者联系,请发送电子邮件到: wang\_y\_t@sohu.com 或 cd\_sywe@sohu.com。

作 者

2006 年 5 月

# 目 录

UML 基础与应用

<b>第 1 章 UML 概述</b> .....	1
1.1 什么是 UML .....	1
1.2 UML 发展历史 .....	3
1.3 UML 主要内容 .....	3
1.3.1 UML 模型图 .....	4
1.3.2 UML 在软件开发各个阶段的应用 .....	6
1.3.3 UML 的应用领域 .....	6
1.4 软件过程 .....	7
1.4.1 软件过程概述 .....	7
1.4.2 统一软件过程 .....	9
1.5 UML 工具 .....	10
1.5.1 主要 UML 工具介绍 .....	11
1.5.2 如何选择 UML 工具 .....	11
小结 .....	12
习题 1 .....	12
<b>第 2 章 用例图</b> .....	13
2.1 UML 视图 .....	13
2.2 用例图 .....	14
2.2.1 用例图概述 .....	15
2.2.2 执行者 .....	16
2.2.3 用例 .....	16
2.2.4 执行者与用例之间的关系 .....	17
2.2.5 理解用例 .....	17

2.2.6 用例描述 .....	18
2.3 用例图图符 .....	22
2.4 用例的粒度和范围 .....	24
2.4.1 用例的粒度 .....	24
2.4.2 用例的范围 .....	26
2.5 用例图应用 .....	27
2.5.1 用户需求 .....	27
2.5.2 需求分析 .....	27
2.5.3 需求描述 .....	30
2.6 用例图进一步说明 .....	33
2.6.1 用例使用提示 .....	33
2.6.2 用例常见问题 .....	33
小结 .....	35
习题 2 .....	36
<b>第 3 章 类图、对象图和包图 .....</b>	<b>37</b>
3.1 类图介绍 .....	37
3.1.1 类 .....	37
3.1.2 类图设计 .....	40
3.1.3 类间关系 .....	41
3.1.4 类图深入讨论 .....	45
3.2 类图图符 .....	49
3.3 类图的应用 .....	50
3.3.1 类图的层次 .....	50
3.3.2 需求描述 .....	51
3.3.3 概念类的提取和描述 .....	52
3.3.4 设计阶段类图 .....	53
3.3.5 类图常见问题 .....	54
3.4 对象图 .....	55
3.5 包图 .....	56
3.5.1 包图的图符 .....	57
3.5.2 包中的元素 .....	57
3.5.3 包与包之间的关系 .....	58
3.5.4 包图常见问题 .....	59

小结 .....	59
习题 3 .....	60
<b>第 4 章 行为图 .....</b>	<b>61</b>
4.1 活动图 .....	61
4.1.1 活动图的概念 .....	62
4.1.2 活动图图符 .....	62
4.1.3 泳道 .....	64
4.1.4 分支 .....	66
4.1.5 分叉和汇合 .....	66
4.2 状态图 .....	67
4.2.1 状态图的概念 .....	68
4.2.2 状态图图符 .....	70
4.2.3 复合状态 .....	70
4.2.4 状态图和活动图的区别 .....	71
4.3 行为图应用 .....	72
4.3.1 活动图设计 .....	73
4.3.2 状态图设计 .....	74
小结 .....	76
习题 4 .....	76
<b>第 5 章 交互图 .....</b>	<b>77</b>
5.1 顺序图 .....	77
5.1.1 顺序图的概念 .....	77
5.1.2 顺序图图符 .....	80
5.1.3 分支的表示 .....	81
5.1.4 条件的表示 .....	81
5.1.5 循环的表示 .....	82
5.2 顺序图应用 .....	83
5.2.1 类图 .....	83
5.2.2 顺序图 .....	84
5.3 合作图 .....	86
5.3.1 合作图的概念 .....	86
5.3.2 合作图图符 .....	89

5.3.3 顺序图与合作图之间的关系 .....	89
5.4 合作图应用 .....	90
5.4.1 对象图 .....	90
5.4.2 合作图 .....	90
小结 .....	91
习题 5 .....	92
<b>第 6 章 实现图 .....</b>	<b>93</b>
6.1 构件图 .....	93
6.1.1 构件图的概念 .....	93
6.1.2 构件图图符 .....	95
6.2 构件图应用 .....	96
6.3 部署图 .....	97
6.3.1 部署图的概念 .....	98
6.3.2 部署图图符 .....	100
6.4 部署图应用 .....	101
小结 .....	103
习题 6 .....	104
<b>第 7 章 UML 在需求分析阶段的应用 .....</b>	<b>105</b>
7.1 UML 在软件开发过程中的应用 .....	105
7.2 AutoWeight 系统简介 .....	106
7.2.1 系统介绍 .....	106
7.2.2 天车称重的工作过程 .....	107
7.3 用户需求 .....	108
7.4 需求分析与描述 .....	110
7.4.1 需求分析 .....	110
7.4.2 用例分析 .....	112
7.4.3 用例模型图 .....	114
7.4.4 用例描述 .....	115
7.4.5 非功能性需求分析 .....	115
7.5 领域模型分析 .....	118
7.5.1 领域概念 .....	118
7.5.2 概念类 .....	118



7.5.3	类间关联	120
7.5.4	增加属性和方法	122
7.5.5	领域模型	123
7.6	工作流程分析	123
7.6.1	称重过程	124
7.6.2	顺序图	124
	小结	125
	实训	126
<b>第 8 章</b>	<b>UML 在设计阶段的应用</b>	<b>128</b>
8.1	对象交互分析	128
8.1.1	系统接口	128
8.1.2	对象设计	133
8.2	数据存储	136
8.2.1	数据保存	136
8.2.2	对象到关系数据库的映射	138
8.3	界面设计	139
8.3.1	开发环境的选择	140
8.3.2	用户界面	141
8.4	其他设计	143
8.4.1	模式应用	143
8.4.2	构件选择	144
8.5	类的设计	144
8.5.1	设计类图	145
8.5.2	设计类	145
8.5.3	设计包	146
	小结	147
	实训	148
<b>第 9 章</b>	<b>UML 在实现阶段的应用</b>	<b>149</b>
9.1	类设计	149
9.1.1	添加属性	149
9.1.2	添加方法	151
9.2	类的实现	153

9.2.1	方法的实现	154
9.2.2	方法的分解	155
9.2.3	类的实现次序	156
9.2.4	代码	157
9.2.5	对象包装	160
小结		162
实训		162
<b>第 10 章</b>	<b>UML 在集成和交付阶段的应用</b>	<b>164</b>
10.1	系统集成	164
10.1.1	系统组成	164
10.1.2	构件图和包图的应用	166
10.2	系统交付	168
10.2.1	系统配置图	168
10.2.2	把构件映射到节点	169
小结		170
实训		171
<b>第 11 章</b>	<b>UML 在测试中的应用</b>	<b>172</b>
11.1	单元测试	173
11.1.1	黑盒测试	173
11.1.2	白盒测试	174
11.1.3	图形界面的测试	175
11.2	集成测试	176
11.2.1	集成测试过程	176
11.2.2	回归测试	177
11.3	系统测试	177
11.3.1	系统测试过程	178
11.3.2	功能测试	178
11.3.3	性能测试	179
11.4	容错程序设计	179
11.4.1	可以预料和防止的错误	180
11.4.2	可以预料但不能防止的错误	180
11.4.3	处理不能预料的错误	180

小结	181
实训	181
<b>第 12 章 AutoWeight 系统 Rose 描述</b>	<b>183</b>
12.1 Rational Rose 简介	183
12.1.1 Rational Rose 安装	183
12.1.2 Rational Rose 主界面	186
12.1.3 Rational Rose 建模	191
12.2 项目需求描述	195
12.2.1 绘制用例图	195
12.2.2 绘制概念类图	200
12.2.3 绘制工作流程活动图	204
12.2.4 绘制顺序图	206
12.3 系统设计描述	207
12.3.1 绘制设计类图	208
12.3.2 绘制交互图	208
12.3.3 绘制状态图	211
12.4 系统部署	212
12.4.1 构件图	212
12.4.2 部署图	214
小结	216
实训	216
<b>附录 A 主要的 UML 工具</b>	<b>217</b>
<b>附录 B UML 术语表</b>	<b>221</b>
<b>附录 C UML 图符总结</b>	<b>226</b>
<b>参考文献</b>	<b>229</b>

## UML 概 述

面向对象技术出现于 20 世纪 70 年代末,它是软件工程领域中的重要技术。面向对象技术不仅是一种程序设计方法,还是一种对现实世界中问题的抽象方式。它的出现改变了人们对软件的认识和理解,同时人们也开始了相关技术的研究,其中最重要的是面向对象建模技术的研究,主要成果就是统一建模语言 UML。

本章介绍 UML 的基本概念、UML 的历史、UML 的主要内容、软件过程的基本知识和主要的 UML 工具。通过本章的学习可以对 UML 有一个概括的了解,为学习后续的内容做必要准备。

### 1.1 什么是 UML

UML(Unified Modeling Language)统一建模语言是用来设计软件蓝图的可视化建模语言。它支持面向对象系统的分析、设计、实现和交付等各个环节,可以用于系统的理解、设计、浏览、维护和信息控制。UML 是由世界著名的面向对象技术专家 Grady Booch、James Rumbaugh 和 Ivar Jacobson 发起,在著名的 Booch 方法、OMT 方法、OOSE 方法的基础上,广泛征求意见,集众家之长,几经修改而成的。

UML 并不是一种程序设计语言,而是一种描述程序设计思想的工具,不局限于某个开发平台或某种程序设计语言。UML 的特点是使用图符和文档相结合的方式描述现实世界中的问题及解决问题的方案。

#### 1. 模型

简单地说,模型是对现实的简化。模型提供了系统的蓝图,包括了从高层次考虑的总体规划,以及详细的设计和实现。由于人对复杂事物的理解能力有限,通过建立一个恰当的模型可以更好地理解一个复杂的系统。一个好的模型包括那些对系统有重要影响的主要因素,而忽略那些细枝末节。

具体地说,模型主要有以下四个作用:

- (1) 模型可以按照能够理解和接受的方式简单明了地表达一个实际的复杂系统。
- (2) 模型可以帮助深入了解系统的结构和行为。
- (3) 通过对模型的分析 and 理解,可以更好地实现这个系统。
- (4) 模型可以帮助进行决策。

建模是一项经过检验并广为接受的工程技术。这项技术广泛应用于建筑、制造、经济管理等行业,同样在软件项目开发中也需要建立模型。

在实际应用中,每个项目都能从建模中受益。建立简明、准确的表示模型是把握复杂系统的关键。随着时间的推移和软件技术的发展,所有的应用系统都变得越来越复杂,因此建模技术也越来越成为项目成败的关键。

## 2. 面向对象的建模

随着面向对象技术的广泛应用,面向对象的建模受到越来越多的重视。作为 OMG (Object Management Group) 标准的 UML 已经成为主要的面向对象建模语言,受到了计算机界的普遍欢迎,得到了许多科研机构 and 软件公司的支持。许多世界知名的软件公司都成为 UML 修订工作的核心小组成员,这些公司包括: Rational 公司、HP 公司、IBM 公司等。

UML 不仅可以支持面向对象的分析与设计,更重要的是能够有力地支持从需求分析开始的软件开发全过程。

使用 UML 可以对现实问题和需要开发的系统进行可视化描述,以帮助用户和项目组成员理解系统,方便相互之间的交流;使用 UML 还可以描述一个系统的结构和行为;不同的 UML 模型图可以作为项目不同阶段的软件开发文档。

使用 UML 可以方便地进行交流和沟通,减少了不同建模系统之间转换的成本。UML 适合软件开发的各个阶段,从需求描述到系统完成后的交付。

## 3. 为什么使用 UML

软件开发的最终目的是得到可执行的软件,建模是为了更好地理解要实现的软件系统,设计软件系统的蓝图。在什么情况下需要使用 UML? 在软件开发过程中是否需要使用 UML? 可以从以下三个方面来考虑:

- (1) 是否使用面向对象技术来开发软件。
- (2) 软件的规模和复杂程度是否超出了用户的控制能力。
- (3) 是否是一个团队进行软件项目开发。

如果满足以上三个方面,建议使用 UML 进行建模。使用 UML 进行建模可以学习和理解面向对象技术,有助于同领域专家进行交流,更好地理解全局。

采用 UML 可以促进面向对象技术的学习。对于大多数人来说,学习一门面向对象语言并不难,难的是如何进行面向对象程序设计、如何发挥面向对象语言提供的优势。UML 可以帮助理解面向对象技术,进行面向对象的设计。在理解用户需求方面,UML 提供了专

门的用例图,使用用例图可以方便有效地捕获用户需求。在描述系统的静态结构方面,UML 提供了类图和包图。UML 还提供了其他的图来描述系统的行为和构成。模式是重要的面向对象设计技术,使用 UML 还可以有助于表述和学习模式。

采用 UML 可以促进与领域专家进行交流。软件开发面临的最大问题是如何在一个合理的费用前提下构造一个正确的系统,以满足用户的需求。这个问题主要表现在如何同用户进行有效的交流。软件开发人员有自己的专业和术语,用户同样也有自己的专业和术语,为了构建软件系统,软件开发人员需要了解用户的专业和术语,也就是了解用户领域的知识。向该领域的专家学习并同他们进行交流是学习领域知识很好的方法。UML 提供了用例图用于了解用户需求。用例图强调系统的整体性,强调系统外在的功能,强调系统对使用者有什么帮助和好处,强调系统与外部的交互,辅以其他的 UML 模型图,可以很好地对要开发的系统进行描述,为进一步构造正确的系统打下坚实的基础。

采用 UML 可以帮助理解全局。开发大型的软件项目时,经常会犯只见树木不见森林的错误。这时开发人员希望能够从全局来了解和理解一个系统,UML 提供了多种模型图用来从高层描述系统。通过使用这些图,可以方便地了解整个系统功能、结构和行为。

在具体的软件项目开发中,使用 UML 进行建模不一定要用到 UML 中所有的模型图。不同的项目中,每种模型图的详细程度也可能不同。因此在使用 UML 进行建模时,用到哪些模型图,每张图的详细程度如何,这些主要取决于问题的复杂程度和软件开发的需要。

## 1.2 UML 发展历史

从 20 世纪 80 年代初期开始,众多的方法学家都在尝试使用不同的方法进行面向对象的分析和设计,其中比较成功的有 OOA/D、Booch、OMT、CRC 等。到 1994 年 10 月,James Rumbaugh 和 Grady Booch 把各自独立发展的 Booch 和 OMT 方法合并起来,并于 1995 年成为“统一方法”(Unified Method)0.8 版本。随后 Ivar Jacobson 加入,并引入了用例(Use Case)思想,于 1996 年形成了“统一建模语言”0.9 版本。1997 年 1 月,UML 1.0 版本被提交到 OMG 组织,作为软件建模语言标准化的候选。1997 年 11 月 7 日 OMG 正式采纳,并把 UML 作为业界标准。1998 年 OMG 接管了 UML 标准的维护工作,推出了 UML 的 1.3 版、1.4 版、1.5 版。后来又推出了 2.0 版。关于这些版本的详细文档可以登陆 OMG 的网站查阅,网址是 <http://www.uml.org>。

## 1.3 UML 主要内容

UML 融合了 Booch、OMT 和 OOSE 方法中的基本概念,并扩展了原有方法的应用范围。作为一种建模语言,UML 的定义包括 UML 语义和 UML 表示法两个部分。UML 语义

描述基于 UML 的精确元模型定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明,使开发者能在语义上取得一致,消除了因人而异的最佳表达方法所造成的影响。此外,UML 还支持对元模型的扩展定义。UML 表示法定义了 UML 符号的表示法,为开发者或开发工具使用这些图形符号和文本语法提供了标准,也为开发者对系统进行建模提供了统一的标准。这些图形符号和文字所表达的是应用级的模型,在语义上它是 UML 元模型的实例。

### 1.3.1 UML 模型图

UML 中定义了 5 类共计 10 种模型图。下面在介绍每种模型图时,给出一个简单的例子来加以说明。

#### 1. 用例图

用例图用来描述用户的需求,它从用户的角度描述系统的功能,并指出各功能的执行者,强调谁在使用系统,系统为执行者完成哪些功能。图 1.1 是一个学生使用计算机辅助教学系统进行自我测试的用例图。

#### 2. 静态图

静态图包括类图、对象图和包图。其中类图用于定义系统中的类,包括描述类的内部结构和类之间的关系,类图主要用于描述系统的静态结构。对象图是类图的一个实例,它描述了系统在某一个具体的时间点上所包含的对象以及各对象之间的关系。包图是由包和类组成,主要表示包与包、包与类之间的关系。包图用于描述系统的分层结构。图 1.2(a)是一个“学生”类的图符,图 1.2(b)是“学生”类的具体对象“张三”的对象图图符。

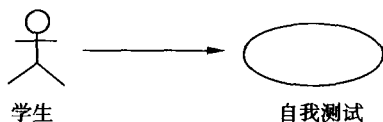


图 1.1 用例图

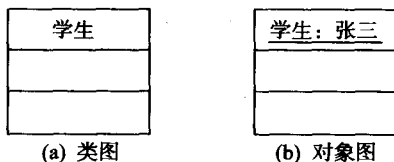


图 1.2 类图和对象图

#### 3. 行为图

行为图主要用来描述系统的动态模型和对象之间的交互关系,包括状态图和活动图。状态图用来描述类的对象所有可能的状态以及事件发生时状态的转移条件。通常,状态图是对类图的补充。活动图用来描述满足用例要求所要进行的活动以及活动间的约束关系,使用活动图有利于识别系统中的并行活动。图 1.3(a)是一个网卡启用状态图,图 1.3(b)是启用网卡的活动图。

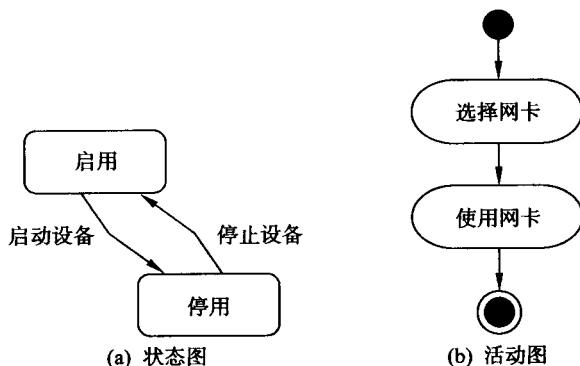


图 1.3 状态图和活动图

#### 4. 交互图

交互图主要用来描述对象之间的交互关系,包括顺序图和合作图。顺序图用来描述对象之间的交互顺序,着重体现对象间消息传递的时间顺序,强调对象之间消息的发送顺序,同时也显示对象之间的交互过程。合作图描述对象之间的合作关系,更侧重于说明哪些对象之间有消息的传递。两个图之间可以相互转化。例如,图 1.4(a)是张三解答试卷并查看结果的顺序图,图 1.4(b)是张三解答试卷并查看结果的合作图。

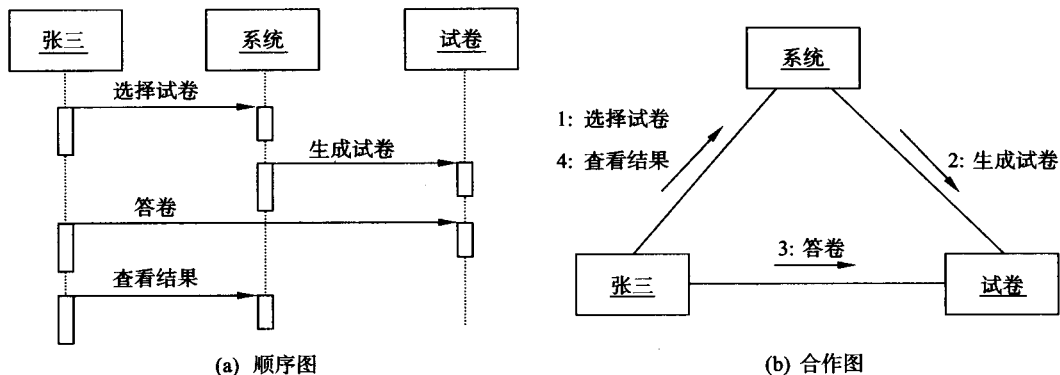


图 1.4 顺序图和合作图

#### 5. 实现图

实现图包括构件图和部署图。构件图用来描述代码构件的物理结构以及各构件之间的依赖关系。一个构件可能是一个资源文件、一个二进制文件或一个可执行文件。部署图定义了系统中硬件的物理体系结构,用来描述实际的物理设备以及它们之间的连接关系。物理设备用节点表示,在节点内部放置可执行的构件和对象,用来显示节点与可执行软件单元



之间的对应关系。

这 5 类 10 种模型图从不同的方面对系统进行了描述,把它们有机地结合到一起就可以分析和构造一个一致的系统。本书从第 2 章开始,用 5 章的内容分别介绍 UML 的 5 类模型图。

### 1.3.2 UML 在软件开发各个阶段的应用

从应用的角度上来看,采用面向对象技术设计软件系统时,使用用例图来描述用户需求;使用类图、对象图、包图、构件图和部署图这 5 种静态图来描述系统的静态结构;使用顺序图、合作图、活动图和状态图这 4 种图来描述系统动态行为。在软件开发的各个阶段使用不同的 UML 图对系统进行描述。

(1) 需求: UML 的用例图可以用来描述客户的需求,通过用例建模,可以找出谁与系统打交道、系统的主要功能、系统和外部是如何进行交互的,并进一步分析这个交互的过程。通过对商业过程和用户要构建的系统进行分析,可以准确把握用户的需求,保证最终的软件系统完全符合用户的需要。

(2) 分析: 分析阶段主要对用户的需求进行进一步的分析,明确要解决问题的细节。使用 UML 的类图来描述系统的静态结构,使用顺序图、合作图、活动图和状态图来描述系统的动态行为。

(3) 设计: 根据对系统的分析提出解决问题的方案。使用 UML 的类图、包图把分析类和包转化成设计类和设计包。对类的接口和实现进行设计。

(4) 实现: 在实现阶段,把设计阶段的类转换成某种面向对象程序设计语言的代码。根据 UML 中详细设计的类图进行实际编码。

(5) 集成与交付: 在系统的集成和交付阶段主要应用 UML 的构件图、包图和部署图来进行系统的集成与部署。

(6) 测试: 根据类图对实现的每个单元进行单元测试,然后对每个类和包的接口进行集成测试,最后对整个软件进行系统测试。单元测试使用类图和类的规格说明书,集成测试使用类图、包图、构件图和合作图。系统测试使用用例图来确认系统的功能和行为。

在软件开发的各个阶段可能会用到相同的模型图来描述,但每个阶段描述的重点和详细程度不同。例如,在软件分析、设计和实现阶段都用到了类图。在分析阶段侧重类的概念描述;在设计阶段侧重类的接口描述;在实现阶段侧重类的实现的描述。很多相同的问题可以用不同的模型图来描述。例如,描述对象之间的关系,可以使用对象图、顺序图、合作图和活动图,但是每个图描述的侧重点不同。

### 1.3.3 UML 的应用领域

UML 是以图形的方式来描述任何类型的系统,具有很宽的应用领域。UML 可以用