

21 世纪 高等学校计算机应用型本科规划教材精选



Java Web 开发 实践教程



杨俊生 唐琳 张坤 编著



清华大学出版社

21 世纪高等学校计算机应用型本科规划教材精选

Java Web 开发实践教程

杨俊生 唐 琳 张 坤 编著

清华大学出版社
北 京

内 容 简 介

本书通过同一个案例的 4 种完整实现版本系统介绍了 Java Web 在软件开发实践方面的知识, 注重软件的实现过程, 以案例项目的实现贯穿全书, 在案例项目实现过程中, 详细讲解了实现过程。本书内容包括软件需求分析、系统分析设计、系统实现、Struts 框架基础知识、整合 Struts 框架过程、Hibernate 框架基础知识和整合 Hibernate 框架过程。

本书涉及知识面较广, 既包括软件工程中需求分析、面向对象分析和设计方法, 也包括 JSP、Struts 框架和 Hibernate 框架技术, 并从实践的角度展示了知识的综合运用。

本书可以作为高等院校计算机专业本科学生的实践教材或参考书, 也可作为 IT 从业人员的自学读物。

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。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java Web 开发实践教程 / 杨俊生, 唐琳, 张坤编著. —北京: 清华大学出版社, 2010.9
(21 世纪高等学校计算机应用型本科规划教材精选)

ISBN 978-7-302-23260-5

I. ①J… II. ①杨… ②唐… ③张… III. ①JAVA 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 147224 号

责任编辑: 索 梅 顾 冰

责任校对: 梁 毅

责任印制: 李红英

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: 010-62795954, jsjtc@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

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: 19.75 字 数: 480 千字

版 次: 2010 年 9 月第 1 版 印 次: 2010 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 29.00 元

产品编号: 035901-01

前言

FOREWORD

当今 Web 应用已经得到了快速发展和普及, Web 开发技术的掌握几乎是每一位从业者和即将从事本行业的 IT 人员的必备技能。在 Web 开发技术中, Java Web 开发阵营在不断壮大, 开源框架在不断增多, 因此对于从业人员或即将跨入这个行业的人员来说, 有时会觉得无所适从。

本书的写作目的就是帮助读者快速掌握 Java Web 的开发过程和实现技术, 在案例项目实现过程中, 不仅仅是展现实现结果, 而是更加注重实现过程的讲解, 因为大多数读者对软件工程过程和开发技术都有一定的理解和掌握, 但是在项目开发过程中却无从下手, 这实际上是因为缺少实践过程所致。无论做什么事情, 懂怎么做 (Know how) 都是至关重要的, 尤其是对于 IT 行业这样既年轻又发展快速的行业, 对从业人员要求是较高的, 希望本书能够帮助读者快速掌握 Java Web 开发。

1. 本书特点

(1) 按照软件工程过程实现案例项目。

本书案例项目的实现首先从需求分析开始, 讲解如何进行用例建模, 不单是图形表达, 更重要的是建模过程; 其次讲解如何进行面向对象分析和设计, 并通过案例项目展示分析和设计的过程; 最后按照分析和设计结果对系统加以实现, 在实现的过程中讲解常见问题的解决办法和实现技术。

(2) 实现步骤详细。

在案例项目的实现过程中, 详细描述了每个实现步骤, 同时采用大量的图形来帮助读者理解。

(3) 案例项目完整, 配置运行简单。

本书给出了案例项目的 4 个完整实现版本 (JSP/Servlet 实现版本, 整合 Struts 框架后的实现版本, 整合 Hibernate 框架后的实现版本以及同时整合 Struts 框架和 Hibernate 框架的实现版本), 可从清华大学出版社网站 (<http://www.tup.com.cn>) 下载, 读者可根据说明配置数据库, 然后任选一个实现版本运行学习。

2. 本书组织结构

本书共分为 3 个部分。

第 1 部分 案例分析与实现: 包括本书的前 4 章。第 1 章讲解需求分析, 利用用例建模法对案例项目进行需求建模; 第 2 章讲解系统设计, 利用面向对象的分析和设计方法对系统进行建模; 第 3 章讲解开发环境搭建, 主要介绍常见开源软件 (MySQL、JDK、Tomcat、NetBean) 的下载, 安装及配置; 第 4 章讲解系统实现, 利用最基本的组件 (如 JSP、Servlet、JavaBean), 按照 MVC 模式进行系统实现, 除了实现基本功能外, 还将讲解一些



开发技巧。

第 2 部分 整合 Struts：包括本书的第 5~7 章。第 5 章是 Struts 框架的入门，通过一个入门应用讲解 Struts 框架的使用过程；第 6 章讲解 Struts 基础知识，并结合一些练习讲解各个知识点的应用；第 7 章讲解如何在第 1 部分实现的系统中运用 Struts 框架来实现视图层和控制层，这部分可帮助读者进一步体会 Struts 在 Web 开发中的地位和作用。

第 3 部分 整合 Hibernate：包括本书的第 8~10 章。第 8 章是 Hibernate 框架的入门，通过一个入门应用讲解 Hibernate 框架的使用过程；第 9 章讲解 Hibernate 基础知识，并结合一些练习讲解各个知识点的应用；第 10 章讲解如何在第 1 部分实现的系统中运用 Hibernate 框架实现数据库访问层，这部分可帮助读者进一步体会 Hibernate 在 Web 开发中的地位和作用。

在本书的写作过程中得到了多方面的支持和帮助。首先感谢清华大学出版社给予的机会；其次感谢参与本书编写的张坤（负责编写本书第 5 章）、唐琳（负责编写本书第 6 章和第 7 章），感谢她们在编写过程中提出的许多宝贵建议；最后要感谢我的家人，感谢他们在我编写书稿的过程中给予的支持和鼓励。

鉴于作者才学有限，书中不足之处欢迎同行、专家及读者批评指正。

杨俊生

2010 年 5 月

目 录

CONTENTS

第1部分 案例分析与实现

第1章 需求分析	2
1.1 基本概念	2
1.1.1 需求定义	2
1.1.2 获取需求	3
1.1.3 表达需求	5
1.2 通讯录管理系统需求建模	10
1.3 本章小结	16
习题1	16
第2章 系统设计	17
2.1 系统设计基本概念	17
2.1.1 软件体系结构	17
2.1.2 MVC 模式	18
2.1.3 面向对象分析	24
2.1.4 面向对象设计	25
2.2 通讯录管理系统建模	31
2.2.1 架构设计	31
2.2.2 系统分析	32
2.2.3 功能设计	33
2.2.4 共通功能设计	70
2.2.5 数据模型设计	73
2.2.6 页面设计	74
2.3 本章小结	76
习题2	77
第3章 开发环境搭建	78
3.1 安装 MySQL	78
3.1.1 安装 MySQL 服务器	78
3.1.2 下载 MySQL 驱动	82



3.1.3 安装图形管理工具 SQLyog	83
3.2 安装 JDK	85
3.3 安装 Tomcat	86
3.4 安装 NetBeans	89
3.5 数据库环境准备	92
3.5.1 创建数据库和数据库用户	92
3.5.2 创建数据库表	94
3.5.3 配置 Tomcat 数据库连接池	97
3.6 创建项目工程	97
3.7 本章小结	102
习题 3	102
第 4 章 系统实现	103
4.1 系统基础功能实现	103
4.1.1 数据库工厂类的实现	103
4.1.2 数据分页显示的实现	103
4.1.3 表单组件中选择类组件显示信息的获取	106
4.1.4 页面信息国际化的实现	110
4.1.5 网页访问安全的实现	115
4.1.6 字符集转换实现	117
4.2 通讯录管理系统功能实现	118
4.2.1 注册功能实现	118
4.2.2 登录功能实现	127
4.2.3 注销功能实现	130
4.2.4 修改基本信息功能实现	131
4.2.5 修改密码功能实现	135
4.2.6 增加联系人功能实现	138
4.2.7 查询联系人功能实现	144
4.2.8 修改联系人功能实现	148
4.2.9 删除联系人功能实现	152
4.3 本章小结	155
习题 4	156

第 2 部分 整合 Struts

第 5 章 初识 Struts	158
5.1 Struts 定义	158
5.2 Struts 在 Java Web 开发中的作用	158
5.3 Struts 工作流程	160

5.4 第一个 Struts 应用的实现	161
5.4.1 应用实现流程和组件介绍	162
5.4.2 创建 Java Web 项目	163
5.4.3 配置资源文件	166
5.4.4 创建 index.jsp 和 main.jsp	166
5.4.5 创建 UserFormBean	167
5.4.6 创建 LoginAction	169
5.4.7 创建 User.java 和 UserDao.java	172
5.5 本章小结	173
习题 5	173
第 6 章 Struts 基础	174
6.1 标记库	174
6.1.1 HTML 标记库	174
6.1.2 Bean 标记库	182
6.1.3 Logic 标记库	183
6.2 ActionForm 类	184
6.2.1 ActionForm 的执行流程	185
6.2.2 多样化的 ActionForm	186
6.2.3 ActionForm Bean 的配置和使用	186
6.2.4 DynaActionForm Bean 的配置与使用	188
6.3 Action 类	189
6.3.1 多样化的 Action 类	189
6.3.2 Action 的配置和编写	189
6.3.3 LookupDispatchAction 的配置和使用	191
6.4 Struts 配置详解	192
6.4.1 struts-config.xml 文件的主要标记	192
6.4.2 数据源配置	192
6.4.3 ActionForm 配置	193
6.4.4 全局异常配置	193
6.4.5 全局转发配置	194
6.4.6 Action 配置	194
6.4.7 controller 配置	195
6.4.8 资源文件配置	195
6.4.9 plug-in 配置	195
6.5 Struts 国际化	196
6.6 Validator 验证框架	197
6.6.1 验证框架简介	197
6.6.2 Struts 验证框架的引入和配置	197

6.6.3	ValidatorForm Bean 的配置和使用	200
6.6.4	ValidatorActionForm Bean 的配置和使用	202
6.6.5	DynaValidatorForm Bean 的配置和使用	202
6.6.6	DynaValidatorActionForm Bean 的配置和使用	203
6.6.7	验证框架中常用验证规则的使用	204
6.7	本章小结	206
	习题 6	206
第 7 章	应用 Struts 框架	207
7.1	环境准备	207
7.1.1	创建支持 Struts 框架的 Web 项目	207
7.1.2	复制已实现的组件	209
7.2	功能实现	210
7.2.1	注册功能实现	210
7.2.2	登录功能实现	216
7.2.3	注销功能实现	218
7.2.4	修改基本信息功能实现	219
7.2.5	修改密码功能实现	222
7.2.6	增加联系人功能实现	225
7.2.7	查询联系人功能实现	229
7.2.8	修改联系人功能实现	231
7.2.9	删除联系人功能实现	235
7.3	本章小结	237
	习题 7	237

第 3 部分 整合 Hibernate

第 8 章	初识 Hibernate	240
8.1	数据持久化技术	240
8.1.1	数据存储形式	240
8.1.2	数据持久化	240
8.1.3	数据持久化的实现技术	241
8.2	Hibernate 简介	241
8.3	Hibernate 工作流程	242
8.4	第一个 Hibernate 应用的实现	243
8.4.1	创建 Java 项目	243
8.4.2	创建数据库和表	244
8.4.3	创建 Hibernate 配置文件 hibernate.cfg.xml	245

8.4.4 创建持久化类和映射文件	247
8.4.5 创建获取 Session 的实用类 HibernateUtil	252
8.4.6 编写实现数据库基本操作的 Dao 类 UserinfoDao.java	253
8.4.7 编写测试程序并运行	255
8.5 本章小结	257
习题 8	257
第 9 章 Hibernate 基础	258
9.1 配置数据库连接	258
9.1.1 使用默认的数据库连接池	258
9.1.2 配置第三方连接池产品	259
9.1.3 使用 Web 服务器配置的数据源	260
9.2 对象-关系映射	260
9.2.1 类映射	260
9.2.2 对象标识符映射	261
9.2.3 属性映射	265
9.2.4 多对一映射	265
9.2.5 一对多映射	266
9.3 操作持久化对象	267
9.3.1 持久化类对象的状态	267
9.3.2 Session 操作数据方法	267
9.4 检索数据	269
9.4.1 HQL 查询	269
9.4.2 QBC 查询	271
9.4.3 SQL 查询	272
9.5 使用 ThreadLocal 管理 Session	272
9.5.1 ThreadLocal	272
9.5.2 使用 ThreadLocal 管理 Session	273
9.6 使用二级缓存	274
9.6.1 Hibernate 缓存介绍	274
9.6.2 配置二级缓存	275
9.6.3 配置类缓存和集合缓存	277
9.6.4 配置查询缓存	278
9.7 本章小结	279
习题 9	279
第 10 章 应用 Hibernate 框架	280
10.1 环境准备	280
10.1.1 复制工程	280



10.1.2 增加 Hibernate 库支持	280
10.1.3 创建 Hibernate 配置文件 hibernate.cfg.xml	281
10.1.4 创建持久化类和映射文件	282
10.1.5 创建获取 Session 的实用类 HibernateUtil	290
10.2 改造 Dao 类实现	291
10.3 系统优化	297
10.3.1 增加过滤器实现 Session 的集中获取和关闭	297
10.3.2 改进持久化类和映射文件	297
10.3.3 增加二级缓存配置	301
10.3.4 改进 Dao 类实现	304
10.4 本章小结	304
习题 10	304
参考文献	305

第1部分

案例分析与实现

第1章

需求分析

需求关注的是“做什么”（do what），它是软件要实现的目标和方向。目标不明确、不清晰则容易导致走弯路，甚至走错路。

本章要点

- 软件需求概念、软件需求的问题；
- 软件需求的获取和表达。

1.1 基本概念

1.1.1 需求定义

1. 软件需求

IEEE 软件工程标准词汇表（1997 年）中定义需求如下：

- （1）用户解决问题或达到目标所需的条件或能力（capability）。
- （2）系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或能力。一种反映上面（1）或（2）所描述的条件或能力的文档说明。

从以上的定义中，依然无法得到有关需求的清晰概念，我们在进行软件项目开发的过程中一直都非常强调“需求”，那么，什么是软件需求呢？软件需求就是系统必须要完成的事或者说是必须具备的某种品质，具体来说一般的软件需求包括功能需求、非功能需求及设计约束三方面的内容。

- 功能需求：是指系统必须完成的那些事，即系统要完成用户提出的各种功能要求。
- 非功能需求：是指软件必须具备的品质或属性，如可靠性、性能、系统响应时间、容错和系统可扩展性等。
- 设计约束：一般是指客户提出的一些补充约束说明，如系统必须基于 SOA 设计模式、必须采用 Oracle 数据库、必须采用商用服务器、必须做应用服务器的负载均衡以及必须采用 UNIX 服务器等技术要求。

需求定义了系统必须具有的能力，一个软件项目的成功与否往往取决于它是否符合需求。需求是否正确有效、是否管理规范对于项目的成败至关重要。

2. 软件需求特点

软件需求的最大特点就是“不稳定”，软件系统天生就是抽象的，无形的，可以任意改变的，也正是它的易变性在不断推动着软件业的快速发展，新的软件技术、新的开发方法以及新的管理模式不断推陈出新，目的只有一个，就是想方设法满足易变的客户需求。需求的易变对于技术人员来说就是要设计开发出更灵活的软件。

案例分析：网站背景问题

曾经有这样一个客户，某软件公司开发人员按照客户的要求完成了软件开发，验收时客户看着网站提了一个听起来不合情理的要求：要求网站的背景能够随着季节的变化而显示不同的背景色。开发人员经过与客户协商，使客户最终放弃了这一要求。

从这一案例中能得到如下启示：

(1) 客户需求管理问题，在调研客户需求时客户没有提出这一明确需求，而后客户在验收时提出需求，很明显是由于对客户需求缺乏有效管理，没有流程约束才导致这一问题的出现。

(2) 软件设计过程中没有充分考虑到软件的可能变化点，导致软件灵活程度不够，其实客户最后提的这个需求不是满足不了，而是软件设计时没考虑到这个变化的可能性，所有页面的背景都写死在各个页面文件中了，可想而知如满足客户这一需求需要的成本有多高。

3. 需求分析

需求分析就是分析客户的需求是什么（分析原系统功能、存在的问题以及客户对未来系统的期望），全面理解客户的各项要求，并准确表达所接受的客户需求。图 1-1 描述了需求分析的输入输出。

简言之，需求分析是获取需求、表达需求和验证需求的过程，最终形成一个客户和开发人员都遵守的规约：软件需求规格说明书，在软件需求规格说明书中详细记录项目的目标、约束条件、功能需求、非功能需求、接口需求和开发运行环境要求等内容。

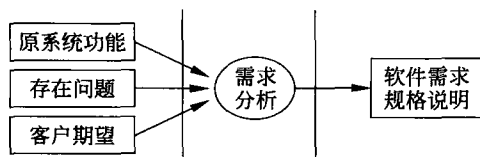


图 1-1 需求分析 IPO 图

1.1.2 节和 1.1.3 节将重点讲述需求分析的前两个过程（获取需求和表达需求），验证需求一般可以通过需求确认会议由客户来验证需求的准确性（通过会议验证需求可以在获取需求过程中一并完成），必要时可以通过开发系统原型来验证。

1.1.2 获取需求

任何一个软件都有需求，而软件系统决定了需求的来源。如果是研发软件产品，需求应由产品经理根据市场情况提出来。如果是为客户定制软件，需求则需要客户提供。这里主要讨论后者。

软件需求的获取，说难也难，说容易也容易。如果碰上对软件系统认识较深的客户，那么不用你问，他也会告诉你他的需求是什么，你只要清楚地记录下来就可以。如果碰上对软件系统一无所知的客户，那就要看调研人员的能力了，除了根据事先设计好的问题提问，还要根据经验不断启发客户。

获取需求大致有以下几种方式：

1. 问卷调查

这是一种通用的方式，事先设计好需要提问的问题，然后形成问卷并将问卷发放给所有和系统相关的客户人员，客户人员根据自己的工作职责，以答卷的形式表达对未来系统的要求。为了能收到更好的效果，要将问卷的内容尽可能细化，避免收集到大量模棱两可的需求。同时问卷调查应注意多提一些开放性问题，给客户足够的表达机会。

问卷调查的确可以收集到大部分客户需求，但对特殊需求仍需单独调研。

2. 客户访谈

客户访谈是最直接、最有效的需求调研方式，通过与客户方的项目相关人员面对面交流，可以直接了解到客户对系统的要求，并在访谈过程中，可以更深层次地挖掘到客户对系统的期望。

访谈的形式可以多样化，可以是正规会议、小组讨论或电话沟通。无论是哪种形式，都应该做好事前准备、事中记录和事后总结，并及时将沟通确认过的内容以文档形式发给客户方。

和客户交流要注意方式、方法，不能盲目约见。下面以事前、事中和事后应注意的事项进行说明。

(1) 事前：充分准备

找什么样的人？在什么样的时间约见？了解什么样的事？这是事前必须考虑清楚的问题。

首先选择合适的访谈对象，选择的标准就是尽可能找有决定权的、精通业务的客户。如果选人不当，往往会造成想了解的事情大都得不到确切答案，随之而来的就是需要就同样的问题访谈不同客户，使访谈的成功率大大下降。

其次选择合适的时间，切忌不要选临近午休或下班前的时间段，这个时间段会让大家感觉紧张，不能平心静气地讨论问题，而使访谈的深度大打折扣，因此最好选择时间宽裕的时间段。

最后就是准备想了解的事，通常访谈前的问题列表准备时间要远远多于访谈的时间，客户在连续和你交谈了 2 个小时之后，就会失去热情和耐心，这是大部分人的共同特点。所以问题列表准备很重要，通过认真研究客户提供的文件和表格等资料，全面了解客户业务，有针对性地提出问题列表，这会使客户感觉你了解业务，能帮助他们想问题，因而在有效的时间内了解尽可能多的事情。

(2) 事中：启发客户、详细记录

通过提问引导客户说出他们的需求是非常关键的。有一些人通常会问这样的问题“你们的工作流程是什么样的？”这种问题是非常经典的无效问题。当你向客户提出问题时，

你可以先进行换位思考，如果有人问你这个问题你该怎么回答，是不是很好回答？如果连你也觉得这个问题并不好回答时，就需要考虑换个问法了。通常人们在谈论自己很熟悉的东西时，都会有说不完的话。因此问恰当的问题，问能让客户打开话匣子的问题，你就胜利了。

如果是在需求会议上，就尽可能抛出全局性的问题（与大家都相关的问题）进行交流，因为需求会议上主要都是利益相关人，大家能聚在一起的机会很难得，因此对问题的讨论可以更深入、更彻底。

同时，要详细记录访谈内容，这时可以借助一些工具来帮助记录，比如录音笔等。因为客户通常不会愿意就同样的问题进行重复回答，这就要求我们不能遗漏客户访谈的内容，尤其是一些关键的业务内容。如果访谈过程中由于客户回答比较快或非常重要而你又没有完全记录下来，那么可以适当打断客户并重复刚才客户讲过的内容让客户确认，这种办法一般不会引起客户反感，反而会让客户觉得你工作认真、细致，所以客户会很耐心地讲解。

（3）事后：总结，及时反馈

通过认真研究访谈记录，整理出一份资料，记录访谈过程中已经明确的问题和待定问题，并尽可能在最短的时间内反馈给客户，让客户确认访谈内容。

如果条件允许，可考虑通过原型的方式来提高客户对软件的认知程度。原型能使客户对软件有一个直观的认识，面对原型，他们可以更好地提出他们的想法和意见，尤其是那些对软件缺乏认识的客户。对原型修改，再确认，最后得到稳定的原型，这时的需求会更加稳定，可以减少实施过程中的反复修改或者返工。

1.1.3 表达需求

软件需求主要包括功能需求、非功能需求和约束条件 3 个方面，可以采用自然语言的方式来描述，例如（一个包含注册功能的网站）：

（1）功能需求部分

注册功能：客户访问网站首页，单击注册链接，之后打开注册页面，客户输入注册相关信息（用户名、密码、联系方式……），然后系统保存信息。

××功能：……

（2）非功能需求部分

- 性能要求：网页单击响应时间在 5 秒内。
- 可靠性：要求系统 7×24 小时连续运行。

⋮

（3）约束条件

- 进度要求：从签订合同起 3 个月内开始试运行。
- 运行环境：Web 服务器：Tomcat 6.0；数据库：MySQL 5.0。

⋮

需求表达本身并没有好坏之分，只要能正确有效表达即可。需求表达的重点在于需求文档要易于交流，这包括与客户交流、与项目团队交流，尤其是与客户交流这一环节，自然语言表达方式不太直观，在表达功能需求时侧重于系统的角度，而不是从用户的角

度。

20 世纪 60 年代后期, Ivar Jacobson 提出了“用例”的概念。20 世纪 80 年代, 他将用例引入了面向对象编程领域, 填补了需求分析过程的一个空白。“用例”的另一种叫法是“用况”, 它们都是译过来的词汇, 其原文为 Use Case, 强调的是用户使用系统的情况, 是一种使用场景, 所以用例法更利于用户理解。接下来将介绍如何基于用例法表达系统的功能需求。

1. 什么是用例

用例描述了系统完成动作的序列, 这一序列动作对执行者产生一个有价值的可见结果。以下进一步分析概念中的元素:

- 动作序列: 动作序列描述了为产生结果而进行的一系列动作, 当执行者提供信息启动该用例时, 就激活了这些动作。
- 系统完成: 系统为执行者服务。
- 有价值的可见结果: 这是最重要的部分, 执行者使用系统一定是为了达成某种目标, 用例必须对执行者有价值。
- 执行者: 可以是一个启动用例的个体或设备。

2. 用例建模

用例建模是统一建模语言 (Unified Modeling Language, UML) 建模的一部分, 也是 UML 最基础的部分。用例建模的最主要功能就是用来表达系统的功能性需求。

用例建模可分为用例图绘制和用例描述两部分, 用例图由执行者 (Actor)、用例 (Use Case)、系统边界和箭头组成 (如图 1-2 所示), 用例描述则用来详细描述用例图中每个用例, 用文本文档来完成。初学者在这个地方容易走入误区, 以为用例建模就是画几个图, 其实不然, 虽然用例图可以帮助我们表达功能需求的全貌, 但用例的根本仍然是文本描述。

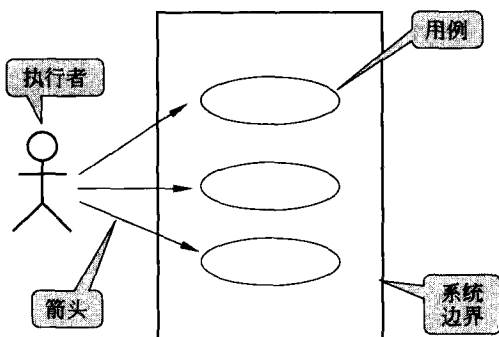


图 1-2 用例图范例

(1) 用例图

- 执行者: 不是特指人, 是指系统以外的在使用系统或与系统交互中所扮演的角色。因此执行者可以是人, 可以是事物, 也可以是时间或其他系统等。执行者在画图中用简笔人物画来表示, 人物下面附上执行者的名称, 如图 1-3 所示。
- 用例: 执行者为达成特定目标操作系统的一组动作序列的描述, 系统执行这些动作, 并产生可观察结果。换句话说, 用例是执行者想要系统做的事情。对于用例的命名, 可以给用例取一个简单却具有描述性的名称, 一般采用动宾结构短语。用例在画图中用椭圆来表示, 椭圆下面附上用例的名称, 如图 1-4 所示。
- 系统边界: 用来表示正在建模系统的边界。边界内表示系统的组成部分, 边界外表示系统外部。系统边界在用例图中用方框来表示, 同时附上系统的名称, 执行