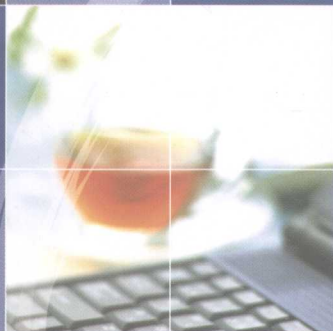
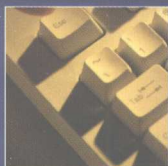




21st CENTURY  
实用规划教材

21世纪全国应用型本科计算机案例型规划教材

# J2EE 课程设计指导



Computer

主 编 王立丰



北京大学出版社  
PEKING UNIVERSITY PRESS

21 世纪全国应用型本科计算机案例型规划教材

## J2EE 课程设计指导

主 编 王立丰  
副主编 林章琼 蒙文荣  
参 编 田保军 石 杰



北京大学出版社  
PEKING UNIVERSITY PRESS

## 内 容 简 介

本书选择 Sun 公司的 Java 作为编程语言,通过一个项目实例阐述项目开发过程及方法;本书不仅给出了项目设计的技术方法,还引入了软件开发项目管理的理念,使学生了解项目开发的过程及项目管理流程,为今后的职业生涯做好准备。

本书可作为高等院校计算机及相关专业程序语言类课程的课程设计辅导教材,也适用于各类培训机构作为培训教材以及编程爱好者作为提高程序开发能力的指导书。

### 图书在版编目(CIP)数据

J2EE 课程设计指导/王立丰主编. —北京:北京大学出版社, 2008.8

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-14132-8

I. J… II. 王… III. Java 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 118765 号

书 名: J2EE 课程设计指导

著作责任者: 王立丰 主编

策 划 编 辑: 孙哲伟

责 任 编 辑: 徐 凡

标 准 书 号: ISBN 978-7-301-14132-8/TP · 0967

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: [pup\\_6@163.com](mailto:pup_6@163.com)

印 刷 者: 河北滦县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 20 印张 465 千字

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷

定 价: 32.00 元

---

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: [fd@pup.pku.edu.cn](mailto:fd@pup.pku.edu.cn)

# 前 言

本书是 Java 语言及 Java 网站开发课程的课程设计教程。为了配合 Java 语言课程的教学,提高学生的动手能力,加强编程技巧的训练,同时为适应软件开发项目管理流程,我们根据多年的教学及项目开发经验,精心设计了多个完整的项目开发案例,从软件项目开发管理的角度,讲解项目开发的具体步骤。由于是一个具体的开发项目,因此,关键技术是一个重要环节,本书针对一些项目开发的关键技术有详细的讲解,同时还给出了一部分重点代码。

J2EE 开发技术已经成为众多应用开发技术的核心,特别是在互联网应用开发方面更是关键技术之一。J2EE 技术的核心是 Java 语言、JSP 编程技术,以及 MVC 设计模式三者的结合,因此,我们打破了以往以独立编程技术模块为中心的训练方式,而代之以一个用综合技术编写的完整项目作为课程设计和综合实训的题目,同时结合企业标准的项目开发方法。目的是不仅使学生掌握每一个单独而实际的编程技巧,而且要掌握模块之间如何衔接,设计与实施如何协调,也就是要用系统的观点来进行项目设计,而不是个人英雄主义,一个人埋头写代码。

本书属于案例开发系列,通过多个案例,从不同的技术角度训练学生的 J2EE 开发技术,包括模式 1 和模式 2 等。通过实例将知识点与开发实战紧密结合,从而达到学以致用的目的。通读本书,不仅可以全面掌握 Java 开发知识,而且可以了解更多 Java 应用技巧。研究本书中的案例还有助于读者认识到使用 J2EE 开发大型应用程序时需要注意的问题,帮助读者建立大局观。

课程设计的任务就是用系统的观点来完成此项目,因此,管理信息系统的系统分析就成了课程设计的分析,系统设计就成了课程内容的设计,系统实施就成了课程设计的实施过程。也就是说,系统设计对应课程设计。

本书第 1 章介绍了标准课程设计的总体要求,阐述了课程设计的目的、对象、设计要求及设计计划的指定。第 2 章重点介绍了案例“项目任务管理系统”开发的过程要求,详细阐述了从系统分析到系统实现的全过程。第 3 章介绍了第二个案例“公司访客登记系统”,给出了系统分析、概要设计和详细设计的过程,读者可根据所提供的资料进行课程设计。第 4 章给出了第三个案例“技术论坛系统”。第 5~7 章给出了三个习题案例,分别是“客户关系管理系统”、“学生信息管理系统”和“应急补货系统”。其中,“客户关系管理系统”采用了标准的 SSH 技术架构,全面展现了 J2EE 高级开发方法。

本书第 1、3 章由王立丰编写,第 2 章由林章琼编写,第 4 章由蒙文荣编写,第 5 章由内蒙古工业大学田保军编写,第 6、7 章由石杰编写。请读者多提宝贵意见。

本书配套有教学支持系统及项目源代码,请到 <http://www.pup6.com> 网站上下载。

编 者  
2008 年 5 月

# 目 录

## 第1章 课程设计详细说明..... 1

- 1.1 课程设计目的..... 1
- 1.2 课程设计对象..... 1
- 1.3 课程设计内容说明..... 1
- 1.4 课程设计的要求..... 2
  - 1.4.1 文档要求..... 2
  - 1.4.2 开发要求..... 3
- 1.5 课程设计计划..... 9
  - 1.5.1 项目计划..... 9
  - 1.5.2 配置计划..... 13
  - 1.5.3 开发计划..... 17
- 1.6 本章小结..... 17

## 第2章 项目任务管理系统..... 18

- 2.1 系统需求分析..... 18
  - 2.1.1 系统介绍..... 18
  - 2.1.2 系统的功能性需求..... 18
  - 2.1.3 软硬件环境需求..... 19
- 2.2 系统设计..... 20
  - 2.2.1 系统概述..... 20
  - 2.2.2 系统总体结构..... 20
  - 2.2.3 子系统的结构与功能..... 20
  - 2.2.4 开发环境的配置..... 24
- 2.3 详细设计及实现..... 25
  - 2.3.1 数据库系统设计..... 25
  - 2.3.2 物理设计..... 26
  - 2.3.3 数据库安全性设计..... 28
  - 2.3.4 界面原型设计..... 29
- 2.4 系统代码编写..... 34
  - 2.4.1 代码编写技巧和思路概述..... 35
  - 2.4.2 系统主工作界面的代码实现... 36
  - 2.4.3 系统登录界面的代码实现..... 46
  - 2.4.4 任务信息管理的实现..... 52
  - 2.4.5 系统管理界面的代码实现..... 65

2.4.6 员工信息管理界面代码实现... 73

2.4.7 优先级管理界面的代码实现... 90

2.4.8 项目信息管理和状态信息管理的实现..... 102

2.5 本章小结..... 102

## 第3章 公司访客登记系统..... 103

- 3.1 系统需求分析..... 103
  - 3.1.1 系统介绍..... 103
  - 3.1.2 系统的功能性需求..... 103
  - 3.1.3 软硬件环境需求..... 103
- 3.2 系统设计..... 104
  - 3.2.1 系统概述..... 104
  - 3.2.2 系统总体结构..... 104
  - 3.2.3 子系统的结构与功能..... 104
  - 3.2.4 开发环境的配置..... 105
- 3.3 详细设计及实现..... 105
  - 3.3.1 数据库系统设计..... 105
  - 3.3.2 界面原型设计..... 106
- 3.4 系统代码编写..... 107
  - 3.4.1 代码编写思路..... 107
  - 3.4.2 系统主界面的实现..... 109
  - 3.4.3 系统功能模块配置的实现..... 112
  - 3.4.4 模块功能流程设计..... 114
  - 3.4.5 代码编写实现..... 116
  - 3.4.6 要点提示..... 147
- 3.5 本章小结..... 148

## 第4章 技术论坛系统..... 149

- 4.1 需求分析..... 149
  - 4.1.1 BBS 系统介绍..... 149
  - 4.1.2 系统的功能性需求..... 150
  - 4.1.3 软硬件环境需求..... 151
- 4.2 系统设计..... 151
  - 4.2.1 系统概述..... 151



|                             |     |                             |     |
|-----------------------------|-----|-----------------------------|-----|
| 4.2.2 系统总体结构 .....          | 152 | 5.3.1 数据库设计 .....           | 216 |
| 4.2.3 子系统的结构与功能 .....       | 152 | 5.3.2 界面及功能设计 .....         | 228 |
| 4.2.4 开发环境的配置 .....         | 154 | 5.3.3 编码实现方法要点 .....        | 247 |
| 4.3 详细设计 .....              | 154 | 5.4 本章小结 .....              | 255 |
| 4.3.1 数据库系统设计 .....         | 154 | <b>第 6 章 学生信息管理系统</b> ..... | 256 |
| 4.3.2 系统界面原型设计 .....        | 156 | 6.1 系统需求分析 .....            | 256 |
| 4.4 系统详细设计流程 .....          | 159 | 6.1.1 系统介绍 .....            | 256 |
| 4.4.1 系统设计思路 .....          | 159 | 6.1.2 系统的功能性需求 .....        | 256 |
| 4.4.2 系统功能流程 .....          | 160 | 6.1.3 软硬件环境需求 .....         | 256 |
| 4.4.3 站点框架配置 .....          | 166 | 6.2 系统设计 .....              | 257 |
| 4.5 编码系统实现 .....            | 168 | 6.2.1 系统概述 .....            | 257 |
| 4.5.1 数据库系统的建立 .....        | 168 | 6.2.2 系统总体结构 .....          | 257 |
| 4.5.2 创建应用项目 .....          | 169 | 6.2.3 开发环境的配置 .....         | 257 |
| 4.5.3 系统主程序设计 .....         | 170 | 6.3 详细设计及实现 .....           | 258 |
| 4.5.4 系统登录程序设计 .....        | 172 | 6.3.1 设计流程和技术方法 .....       | 258 |
| 4.5.5 系统注册程序设计 .....        | 180 | 6.3.2 数据库设计 .....           | 261 |
| 4.5.6 发帖程序设计 .....          | 184 | 6.3.3 界面原型及设计方法 .....       | 265 |
| 4.5.7 回帖程序设计 .....          | 192 | 6.4 本章小结 .....              | 275 |
| 4.5.8 论坛分类板块程序设计 .....      | 200 | <b>第 7 章 应急补货系统</b> .....   | 276 |
| 4.5.9 全局过滤器的设计 .....        | 202 | 7.1 需求分析 .....              | 276 |
| 4.5.10 数据库编程设计 .....        | 209 | 7.2 概要设计 .....              | 284 |
| 4.6 本章小结 .....              | 212 | 7.2.1 系统运行环境 .....          | 284 |
| <b>第 5 章 客户关系管理系统</b> ..... | 213 | 7.2.2 总体结构设计 .....          | 284 |
| 5.1 系统需求分析 .....            | 213 | 7.2.3 数据库结构设计 .....         | 285 |
| 5.1.1 系统产生的背景 .....         | 213 | 7.3 系统实现的要点解析 .....         | 287 |
| 5.1.2 系统的功能性需求 .....        | 214 | 7.4 界面设计及处理流程 .....         | 297 |
| 5.1.3 软硬件环境需求 .....         | 214 | 7.4.1 界面设计 .....            | 297 |
| 5.2 系统设计 .....              | 214 | 7.4.2 处理流程 .....            | 306 |
| 5.2.1 系统概述 .....            | 214 | 7.5 本章小结 .....              | 309 |
| 5.2.2 系统总体结构 .....          | 214 | <b>参考文献</b> .....           | 310 |
| 5.2.3 子系统的结构与功能 .....       | 215 |                             |     |
| 5.2.4 开发环境的配置 .....         | 215 |                             |     |
| 5.3 详细设计 .....              | 216 |                             |     |

# 第1章 课程设计详细说明

## 1.1 课程设计目的

课程设计是课程教学中的一项重要内容,是完成教学计划,达到教学目标的重要环节,是教学计划中综合性较强的实践教学环节。它对帮助学生全面牢固地掌握课堂教学内容,培养学生的实践和实际动手能力,提高学生全面素质具有很重要的意义。

“J2EE 课程设计”是“Java 语言程序设计”和“JSP 程序设计”课程的一个重要的实践性教学环节,其目的是使学生能够在以下几个方面获益匪浅。

(1) 进一步巩固和加深理解“Java 语言程序设计”和“JSP 程序设计”的基本知识,了解 J2EE 在项目开发中的应用。

(2) 综合运用“Java 语言程序设计”和“JSP 程序设计”基本知识和“软件工程”理论,来分析和解决课程设计问题,从而进行课程设计的训练。

(3) 学习程序设计开发的一般方法,了解和掌握信息系统项目开发的过程及方式,培养正确的设计思想、分析问题和解决问题的能力,特别是项目设计能力。

(4) 通过对标准化、规范化文档的学习和查阅有关技术资料等,培养项目设计开发能力,同时提倡团队合作精神。

通过本次实践活动,使学生能够熟练地运用 Java 及 JSP 语言进行项目开发,并能按照系统工程化的方法开发一般的管理信息系统项目。

## 1.2 课程设计对象

本课程设计适用于已学习过“Java 语言程序设计”和“JSP 程序设计”的学生,包括在校的计算机及其相关专业的学生,以及非计算机专业的工科学生或部分管理类学生。本课程设计对于程序爱好者及初涉项目管理的人员也有一定的帮助。

## 1.3 课程设计内容说明

J2EE 的课程设计应在指导教师的帮助下完成各项步骤,具体内容如下。

### 1. 选题

选题可分为指导教师选题和学生自己选题两种。教师选题可选择统一的题目,学生选题应通过指导教师批准后方可进行。

#### 1) 选题内容

选题要符合本课程的教学要求,要注意选题的完整性,要能进行分析建模、设计、编

程、复审、测试等一系列工作，并能以规范的文档形式将其表现出来。

## 2) 选题要求

(1) 注意选题内容的先进性、综合性、实践性，应适合实践教学并有利于启发创新，选题内容不应过于简单，难度要适中。

(2) 结合企事业单位应用的实际情况进行选题。

(3) 题目成果应具有相对完整的功能。

## 2. 拟出具体的设计方案

学生应在指导教师的指导下，进行项目的总体方案设计与论证。学生应根据自己所接受的设计题目设计出具体的实施方案，报告指导教师批准后开始实施。

## 3. 程序的设计与调试

学生在指导教师的指导下，应完成所接受题目的项目开发工作，并进行编程和上机调试，最后得出预期的成果。

## 4. 撰写课程设计总结报告

课程设计总结报告是课程设计工作的整理和总结，主要包括需求分析、总体设计、详细设计、复审、编码、测试等部分，最后写出课程设计的总结报告。

# 1.4 课程设计要求

## 1.4.1 文档要求

课程设计总结报告的撰写规范应参照 CMM 模型编写，最终以课程设计实验报告的形式上交。

课程设计的总结报告是在完成设计、编程、调试之后。对归纳技术文档、撰写科学技术总结报告能力的训练，可以培养学生严谨的作风和科学的态度。通过撰写课程设计总结报告，不仅可以把分析、设计、安装、调试及技术参考等内容进行全面总结，而且还可以把实践内容提升到理论高度。总结报告可按如下内容，依次用 A4 纸进行打印(撰写)并装订成册。

(1) 统一的封面(含软件工程课程设计课题名称、专业、班级、姓名、学号、指导教师等)。

(2) 设计任务和技术要求(由指导教师在选题时提供给学生)。

(3) 内容摘要。

(4) 目录。

(5) 课程设计总结报告正文。正文可按章节来撰写，应包含如下内容。

① 项目需求分析(方案的可行性分析、方案的论证等内容)。

② 项目总体设计(系统的总体结构设计等内容)。

③ 项目详细设计(各模块或单元程序的设计、算法原理阐述、完整的程序框图)。



④ 项目复审(对分析、总体设计和详细设计进行详细的复审)。

⑤ 编码(根据某一程序设计语言对设计结果进行编码的程序清单)。

⑥ 项目测试(使用程序调试的方法和技巧排除故障;选用合理的测试用例进行程序系统测试和数据误差分析等)。

⑦ 总结(本课题核心内容程序清单及使用价值、程序设计的特点和方案的优缺点、改进方法和意见)。

(6) 按统一格式列出主要参考文献。

#### 1.4.2 开发要求

##### 1. 程序开发过程要求

软件系统的开发是按阶段进行的,一般可划分为八个阶段:可行性分析,需求分析,系统设计(概要设计、详细设计),程序开发,编码,单元测试,系统测试,系统维护。

在软件开发过程中,要明确各阶段的工作目标,也要了解实现该目标所必需的工作内容,并要明确须达到的标准。只有在上一个阶段的工作完成后,才能开始下一阶段的工作。

##### 1) 可行性分析

明确系统的目的、功能和要求,了解目前所具备的开发环境和条件。需要论证的内容有以下五个方面。

(1) 在技术能力上是否可以支持。

(2) 在经济上效益如何。

(3) 在法律上是否符合要求。

(4) 与部门、企业的经营和发展是否吻合。

(5) 系统投入运行后的维护有无保障。

讨论可行性的目的是判定软件系统的开发有无价值。将分析和讨论的内容整理、完善,从而形成“项目开发计划书”,其主要内容有以下六个方面。

(1) 开发的目的及所期待的效果。

(2) 系统的基本设想,所涉及的业务对象和范围。

(3) 开发进度表,开发组织结构。

(4) 开发、运行的费用。

(5) 预期的系统效益。

(6) 开发过程中可能遇到的问题及注意事项。

可行性研究报告是可行性分析阶段软件文档管理的标准化文档。

##### 2) 系统需求分析

系统需求分析是软件系统开发中最重要的一个阶段,直接决定着系统的开发质量和成败,必须明确用户的要求和应用现场环境的特点,了解系统应具有哪些功能、数据的流程和数据之间的联系。需求分析应有用户参加,需要到使用现场进行调研学习,软件设计人员应虚心向技术人员和使用人员请教,共同讨论解决需求问题的方法,对调查结果进行分析,明确问题的所在。需求分析的内容应编写成“需求分析规格说明书”。

作为分析结果,软件需求规格说明是软件开发、软件验收和管理的依据。因此,必须特别重视,不能有一点错误或不当,否则将来可能要付出很大的代价。

### 3) 系统设计

可根据系统的规模,将系统设计分成概要设计和详细设计两个阶段。

概要设计主要包括以下九个方面。

- (1) 划分系统模块。
- (2) 每个模块的功能确定。
- (3) 用户使用界面概要设计。
- (4) 输入、输出数据的概要设计。
- (5) 报表概要设计。
- (6) 数据之间的联系、流程分析。
- (7) 文件和数据库表的逻辑设计。
- (8) 硬件、软件开发平台的确定。
- (9) 有规律数据的规范化及数据唯一性要求。

系统概要设计阶段的规范化文档是软件系统概要说明书。

系统详细设计是对系统概要设计的进一步具体化,其主要工作有以下四个方面。

- (1) 文件和数据库的物理设计。
- (2) 输入、输出记录的方案设计。
- (3) 对各子系统的处理方式和处理内容进行细化设计。
- (4) 编制程序设计任务书。程序说明书通常包括程序规范、功能说明和程序结构图,

通常用 HPIPO(Hierarchy Plus Input Process Output)图描述。

系统详细设计阶段的规范化文档为软件系统详细设计说明书。

### 4) 程序开发

程序开发是根据程序设计任务书的要求,用计算机算法语言实现解题的步骤,主要工作包括以下四个方面。

- (1) 模块的理解和进一步划分。
- (2) 以模块为单位的逻辑设计,也就是模块内的流程图的编制。
- (3) 编写代码,用程序设计语言编制程序。
- (4) 进行模块内功能的测试、单元测试。

对程序质量的要求包括以下五个方面。

- (1) 满足要求的确切功能。
- (2) 处理效率高。
- (3) 操作方便,用户界面友好。
- (4) 程序代码的可读性好,函数、变量标识符合规范。
- (5) 扩充性和维护性好。

降低程序的复杂性也是十分重要的。系统的复杂性由模块间的接口数来衡量,一般来说, $n$  个模块的接口数的最大值为  $n(n-1)/2$ ;若是层次结构, $n$  个模块的接口数的最小值为  $n-1$ 。为使  $z$  复杂性达到最小,对模块的划分设计常常采用层次结构。要注意,编制的程序应易于理解,容易修改,模块应相互独立,对某一模块的修改应确保对其他模块的功能不产生影响,模块间的联系尽可能少。

### 5) 系统测试

测试是为了发现程序中的错误。系统测试通常由经验丰富的设计人员设计测试方案和测试样品,并写出测试过程的详细报告。系统测试是在单元测试的基础上进行的,包括下面四个方面。

- (1) 测试方案的设计。
- (2) 进行测试。
- (3) 写出测试报告。
- (4) 用户对测试结果进行评价。

除非是测试一个小程序,否则,一开始就把整个系统作为一个单独的实体来测试是不现实的。与开发过程类似,测试过程也必须分步骤进行,每个步骤在逻辑上是前一个步骤的继续。大型软件系统通常由若干子系统组成,每个子系统又由许多模块组成。因此,大型软件系统的测试基本上由下述四个步骤组成。

- (1) 模块测试。
- (2) 子系统测试。
- (3) 系统测试。
- (4) 验收测试。

软件测试方法常用黑盒法和白盒法。

### 6) 文档资料

文档包括开发过程中的所有技术资料以及用户所需的文档,软件系统的文档一般可分为系统文档和用户文档两类。用户文档主要描述系统功能和使用方法,并不考虑这些功能是怎样实现的;系统文档主要描述系统设计、实现和测试等方面的内容。文档是影响软件可维护性、可用性的决定因素。不夸张地讲,系统编程人员的每一张纸片都要保留,所以文档的编制是软件开发过程中的一项重要工作。

系统文档包括开发软件系统时在计划、需求分析、设计、编制、调试、运行等阶段的有关文档。在对软件系统进行修改时,系统文档应同步更新,并注明修改者和修改日期,如有必要应注明修改原因。

用户文档包括如下四个方面。

- (1) 系统功能描述。
- (2) 安装文档:说明系统安装步骤以及系统的硬件配置方法。
- (3) 用户使用手册:说明使用软件系统的方法和要求,疑难问题解答。
- (4) 参考手册:描述可以使用的所有系统设施,解释系统出错信息的含义及解决途径。

### 7) 系统的运行与维护

系统只有投入运行后,才能进一步对其进行检验,发现潜在的问题。为了适应环境的变化和用户要求的改变,可能会对系统的功能、使用界面进行修改。要对每次发现的问题和修改的内容建立系统维护文档,并使系统文档资料同步更新。

通过建立代码编写规范,可以形成开发小组编码约定,从而提高程序的可靠性、可读性、可修改性、可维护性和一致性,以保证程序代码的质量,继承软件开发成果,充分利用资源,同时提高程序的可继承性,使开发人员之间的工作成果可以共享。

软件编码要遵循如下原则。

- (1) 遵循开发流程, 在设计说明书的指导下进行代码编写。
- (2) 代码的编写以实现设计的功能和性能为目标, 要求正确完成设计要求的功能, 达到设计的性能。
- (3) 程序具有良好的程序结构, 提高程序的封装性, 降低程序的耦合程度。
- (4) 程序可读性强, 易于理解; 方便调试和测试, 可测试性好。
- (5) 易于使用和维护; 具有良好的可修改性和扩充性; 可重用性强, 移植性好。
- (6) 占用资源少, 能以较低代价完成任务。
- (7) 在不降低程序可读性的前提下, 尽量提高代码的执行效率。

## 2. 程序设计代码规范

在开发中保持良好的编码规范是十分重要的。以下 Java 编码规范是一种被证明能明显改善代码可读性, 并有助于代码管理、分类的编码规范。采用这种编码规范, 能避免使用匈牙利命名法带来的繁长前缀, 便于记忆变量的用途。在课程设计中建议使用以下这种编码规范。

注意: 本规范只是 Java 语言编写的基本规则, 由于该语言规范会不断升级和完善, 更加详细的内容可以到 Sun 公司网站上查阅, 该公司将随时发布最新规范版本。

### 1) 文件名

这一节列出了本书所用的大部分文件名和扩展名。

#### (1) 文件扩展名。

.java: Java 源文件的扩展名。

.class: Java 字节码文件的扩展名。

#### (2) 常用文件名。经常使用的文件名包括以下内容。

GNUmakefile: 首选的 makefile 的名字, 使用 gnumake 来建立的软件。

README: 那些专门概述特定文件夹内容的文件的首选名字。

### 2) 文件的组织

一个文件的各个部分之间应该用空行隔开, 并且应该用一个可选的注解来标示每个不同的部分。

如果一个代码文件超过 2000 行, 是非常笨重的, 应该尽量避免。

#### (1) Java 源码文件。

每一个 Java 源码文件都包括一个唯一的 public 类或 interface。当私有的类和 interface 都和这个 public 类有关联时, 可以把它们放到这个 public 类的源文件中。这个 public 类或 interface 应当是这个文件的第一个类或 interface。

#### (2) Java 源文件有以下的顺序。

文件开头注解和声明 package 的语句和载入语句。

类和 interface 的声明。

#### (3) 开头注解。

所有的源文件都应该以一个 C 语言风格的注解开头。这个注解应该列出类名, 版本信息, 日期和版权声明, 具体例子如下。

```
/*
 * Classname
 *
 * Version information
 *
 * Date
 *
 * copyright notice
 *
 */
```

#### (4) 声明包的语句和 import 语句。

绝大多数 Java 源文件中的第一非注释行应该是声明包的语句。此后, 紧接着是 import 语句。例如:

```
package java.awt;
import java.awt.peer.CanvasPeer;
```

#### (5) 类和接口的声明。

表 1-1 描述了部分类和接口的声明, 它们应该按照表格的顺序。

表 1-1 类声明规范

| 部分类/接口声明                    | 注 释                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------|
| 类/接口文档注解/**...*/            |                                                                                                          |
| 类或接口声明                      |                                                                                                          |
| 类/接口实现的注解(/*...*/), 如果有必要的话 | 这个注解应该包括任何整个类或接口范围内的不适合在类/接口文档注解中出现的内容                                                                   |
| 类(静态)变量                     | 首先是 public 类变量, 然后是 protected 类变量, 然后是 friendly(package level, 即默认)。然后是 private 变量                       |
| Instance variables          | 首先是 public, 其次是 protected, 接着是 package level。最后是 private 变量                                              |
| 类的构造函数                      |                                                                                                          |
| 方法(译注: 即类的成员函数)             | 这些方法应该以功能相近为标准组织在一起, 而不是看其作用域和可存取性。例如: 一个 private class 方法可以放在两个 public instance 方法中间。其目的是为了增加代码可读性和可理解性 |

#### (6) 缩进。

① 应该以四个空格为缩进的最小单位, 缩进的精确结构没有被详细定义。Tabs 必须被精确指定为八个空格(而不是四个);

② 代码行的长度应避免超过 80 个字符, 因为很多终端和工具都不支持超过 80 个字符。

(注: 个人认为现在一般都支持行超过 80 个字符, 但是由于超过 80 个字符一般要滚屏, 所以尽量还是不要超过此限制)。

注意: 文档中的例子的行长度应该更短, 一般不超过 70 个字符。

③ Wrapping lines(断行的方法): 当一个表达式不适合一个单行时, 依照以下这些一般性原理来断行。

在逗号后断行。

在运算操作符前断行。

在较高的层次断行比在低层次断行要好。

把新行的表达式的开头和上一行的表达式开头对齐。

如果上一条规则导致代码的混乱或者代码超出了正常的限度。那么就仅仅以缩进 8 个空格来替代。下面有一些调用方法的断行例子。

```
someMethod(longExpression1, longExpression2, longExpression3,
            longExpression4, longExpression5);
var = someMethod1(longExpression1,
                  someMethod2(longExpression2,
                               longExpression3));
```

下面是两个算术表达式的断行的例子, 第一个比第二个好, 因为它的断行发生在括号的外面, 这是在高层次上的断行。

```
longName1 = longName2 * (longName3 + longName4 - longName5)
                + 4 * longName6;
```

```
longName1 = longName2 * ( longName3 + longName4
                          - longName5) + 4 * longName6;    // 应该避免这样
```

下面是两个方法声明的断行的例子, 第一个是规范的例子。第二个如果要使用规范的缩进的话, 将会使第二和第三行的缩进非常向右。所以仅仅使用 8 个空格来替代。

// 规范的缩进

```
someMethod ( int anArg, Object anotherArg, String yetAnotherArg,
             Object andStillAnother){
```

...

// 以 8 个空格来缩进, 以避免非常纵深的缩进

```
private static synchronized horkingLongMethodName(int anArg,
             Object anotherArg, String yetAnotherArg,
             Object andStillAnother) {
```

...

}

④ 行的包装(Line wrapping for): 对 if 语句一般应该用 8 个空格规则, 因为标准的缩进(4 个空格)会使得 if 的主体部分非常难以看明白。例如:

// 不要使用这种缩进



```

if ( ( condition1 && condition2)
    || (condition3 && condition4)
    || ( condition5 && condition6)) { //差的包装
    doSomethingAboutIt();
}
//以如下缩进方式代替。
if ( ( condition1 && condition2)
    || ( condition3 && condition4 )
    || ! (condition5 && condition6)) {
    doSomethingAboutIt();
}
//或者这样使用。
if ((condition1 && condition2) || (condition3 && condition4)
    || ! ( condition5 && condition6 )) {
    doSomethingAboutIt();
}

```

下面有三种可接受的三元运算符的缩进格式。

```

alpha = ( aLongBooleanExpression) ? beta : gamma;
alpha = ( aLongBooleanExpression) ? beta
                                     : gamma;
alpha = ( aLongBooleanExpression)
        ? beta
        : gamma;

```

一般地, 要求内容有名称、功能、作者、说明、创建、修改、参数与返回。

## 1.5 课程设计方案

### 1.5.1 项目计划

#### 1. 引言

##### 1) 编写目的

说明编写这份项目计划的目的, 并指出预期的读者。

例如, 为了保证项目团队按时保质地完成项目目标, 便于项目团队成员更好地了解项目情况, 使项目工作开展的各个过程合理有序, 因此以文件化的形式, 把对于在项目生命周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、团队内外沟通协作方式、开发进度、经费预算、项目内外环境条件、风险对策等内容做出的安排以书面的方式, 作为项目团队成员以及项目干系人之间的共识与约定、项目生命周期内的所有项目活动的行动基础和项目团队开展、检查项目工作的依据。

##### 2) 背景

主要说明项目的来历, 一些需要项目团队成员知道的相关情况。其主要内容如下。

- (1) 项目的名称。经过与客户商定或经过立项手续统一确定的项目名称。
- (2) 项目的委托单位。如果是根据合同进行的软件开发项目, 项目的委托单位就是合

同中的甲方；如果是自行研发的软件系统，项目的委托单位就是本企业。

(3) 项目的用户(单位)。软件或网络的使用单位，可以泛指某个用户群。

(4) 项目的任务提出者。本企业内部提出需要完成此项目的人员，一般是领导或商务人员。

(5) 项目的主要承担部门。有些企业根据行业方向或工作性质的不同把软件开发分成不同的部门(也有的分为不同事业部)。项目的特点就是其矩阵式组织，一般一个项目的项目成员可能由不同的部门组成，甚至可能由研发部门、开发部门、测试部门、集成部门、服务部门等其中的几个组成。需要根据项目所涉及的范围确定本项目的主要承担部门。

(6) 项目建设背景。从政治环境上、业务环境上说明项目建设背景，说明项目的大环境、来龙去脉。这有利于项目成员更好地理解项目目标和各项任务。

(7) 软件系统与其他系统的关系。说明与本系统有关的其他系统，说明它们之间的相互依赖关系。这些系统可以是这个系统的基础性系统(一些数据、环境等必须依靠这个系统才能运行)，也可以是以这个系统为基础的系统，或者是两者兼而有之、互相依赖的系统。

(8) 软件系统与机构的关系。说明软件系统除了委托单位和使用单位，还与哪些机构组织有关系。

### 3) 定义

列出为正确理解本计划书所用到的专门术语的定义、外文缩写词的原词及中文解释。注意尽量不要对一些业界使用的通用术语进行另外的定义，使它的含义和通用术语的惯用含义不一致。

### 4) 参考资料

列出本计划书中所引用的及相关的文件资料和标准的作者、标题、编号、发表日期和出版单位，必要时说明得到这些文件资料和标准的途径。

### 5) 标准、条约和约定

列出在本项目开发过程中必须遵守的标准、条约和约定。例如，相应的《立项建议书》、《项目任务书》、合同、国家标准、行业标准、上级机关有关通知和实施方案、相应的技术规范等。

“参考资料”一般具有“物质”特性，一般要说明参照了什么，要说明在哪里可以获得；“标准、条约和约定”一般具有“精神”特性，一般是必须遵守的，不说明在哪里可以获得。参考资料的内容应该涵盖“标准、条约和约定”。

## 2. 项目概述

### 1) 项目目标

设定项目目标就是把项目要完成的工作用清晰的语言描述出来，让项目团队每一个成员都有明确的概念。注意，不要简单地说成在何种时间完成开发何种软件系统或完成什么什么软件安装集成任务。“要完成一个系统”只是一个模糊的目标，它还不够具体和明确。明确的项目目标应该指出服务对象，所开发软件系统最主要的功能和系统本身的比较深层次的社会目的或系统使用后所起到的社会效果。

项目目标可以进行横向的分解也可以进行纵向的分解。横向分解一般按照系统的功能

或按照建设单位的不同业务要求,如分解为第一目标、第二目标等;纵向的分解一般是指按照阶段,如分解为第一阶段目标、第二阶段目标等,或近期目标、中期目标、远期目标等等。阶段目标一般应当说明目标实现的较为明确的时间。一般要在说明了总目标的基础上再说明分解目标,可加上“为实现项目的总目标,必须实现以下三个阶段目标……”

## 2) 系统目标与范围

根据项目输入(如合同、立项建议书、项目技术方案、标书等)说明此项目要实现的软件系统的目的与目标及简要的软件功能需求。对项目成果(软件系统)范围进行准确清晰的界定与说明,是软件开发项目活动开展的基础和依据。软件系统目标应当从用户的角度说明开发这一软件系统是为了解决用户的哪些问题。系统目标如“提高工作信息报送反馈工作效率,更好地进行工作信息报送的检查监督,提高信息的及时性和汇总统计信息的准确性,减轻各级相关工作人员的劳动强度。”

## 3) 假设与约束

对于项目必须遵守的各种约束(时间、人员、预算、设备等)进行说明,这些内容将限制你实现什么、怎样实现、什么时候实现、成本范围等种种制约条件。

假设是通过努力可以直接解决的问题,而这些问题是一定要解决才能保证项目按计划完成。如,“系统分析员必须在三天内到位”或“用户必须在某月某日前确定对需求文档进行确认”。约束一般是难以解决的问题,但可以通过其他途径回避或弥补、取舍,如,人力资源的约束限制,就必须牺牲进度或质量等等。

假设与约束是针对比较明确会出现的情况,如果问题的出现具有不确定性,则应该在风险分析中列出,分析其出现的可能性(概率)、造成的影响、应当采取的相应措施。

## 4) 项目工作范围

说明为实现项目的目标需要进行哪些工作。在必要时,可描述与合作单位和用户的工作分工。

注意系统范围与项目工作范围的不同含义。

系统范围界定:软件系统本身范围的特征和功能范围。

工作范围界定:为了能够按时保质交付一个有特殊的特征和功能的软件系统所要完成的那些工作任务。

系统范围的完成情况是参照客户的需求来衡量的,而项目范围的完成情况则是参照计划来检验的。这两个范围管理模型间必须要有较好的统一性,以确保项目的具体工作成果,能按特定的系统要求准时交付。

## 5) 应交付成果

(1) 需完成的软件。列出需要完成的程序的名称、所用的编程语言及存储程序的媒体形式。其中软件对象可能包括源程序、数据库对象创建语句、可执行程序、支撑系统的数据库数据、配置文件、第三方模块、界面文件、界面原稿文件、声音文件、安装软件和安装软件源程序文件等。

(2) 需提交用户的文档。列出需要移交给用户的每种文档的名称、内容要点及存储形式,如需求规格说明书、帮助手册等。此处需要移交用户的文档可参考合同中的规定。