

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
软件工程

软件工程设计 案例教程

李代平等编著



清华大学出版社

高等学校教材
软件工程

软件工程设计 案例教程

李代平等编著

清华大学出版社
北京

内 容 简 介

本书简要介绍了软件工程设计的基本原理、方法和技术,涵盖了传统软件工程方法和面向对象方法。书中所提供的各种设计案例都是来自实际项目开发过程中经验性案例,实用性强。书中给出了软件工程设计开发环节中具体的设计活动过程和产品规范,通过多方面的案例向读者介绍软件设计的全过程,以提高读者设计软件的能力。

本书可作为本科院校计算机及相关专业的软件工程设计课程教材,也可供从事计算机软件开发及应用的广大科技人员参考。

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 13501256678 13801310933

图书在版编目(CIP)数据

软件工程设计案例教程/李代平等编著. —北京:清华大学出版社,2008.2

(高等学校教材·软件工程)

ISBN 978-7-302-15860-8

I. 软… II. 李… III. ①软件工程—高等学校—教材 ②软件设计—高等学校—教材
IV. TP311.5

中国版本图书馆 CIP 数据核字(2007)第 116394 号

责任编辑:丁 岭 林都嘉

责任校对:时翠兰

责任印制:何 芊

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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:25 字 数:609千字

版 次:2008年2月第1版 印 次:2008年2月第1次印刷

印 数:1~3000

定 价:39.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026468-01

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

- (1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 高等学校教材·信息管理与信息系统。
- (6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过 20 多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
E-mail: dingl@tup.tsinghua.edu.cn

一、本书背景

目前,软件业受到世界各国前所未有的关注。软件工程是指导计算机软件开发的一门工程科学技术。软件工程的概念、原理、技术与方法已成为计算机科学与技术的一项重要内容。软件的开发和研制能力已成为一个国家科技、经济和国防实力的重要标志。由于软件的规模不断扩大,软件的生产成本也随着软件的复杂程度而急剧增加。它的复杂性给它的大规模发展和维护带来了困难,使软件技术的发展远远落后于软件的需求。但是人们在开发研制软件的长期实践中也积累总结了许多成功的经验,如果能坚持不懈地总结和使用这些经过实践检验的方法和经验,软件业是可以快速健康发展的。

作为软件技术人员,接受软件工程的概念并不难,但是要真正理解、掌握和运用这门先进的技术并完整地进行系统开发,是有一定难度的。鉴于此我们编写了本教材,其目的是向读者提供一本关于软件工程详细设计的教科书,以使更多同行受益。

二、本书结构

本书由如下部分组成。

基础理论包括:

第1章 系统设计

实践案例包括:

第2章 需求分析

第3章 总体方案

第4章 业务管理子系统设计

第5章 技术开发管理子系统设计

第6章 品质保证子系统设计

第7章 事业管理子系统设计

第8章 财务子系统设计

第9章 人事管理子系统设计

第10章 后勤管理子系统设计

- 第 11 章 数据迁移设计
- 第 12 章 系统平台设计
- 第 13 章 系统组织与信息分类编码标准
- 第 14 章 接口
- 第 15 章 关键技术与实施计划
- 第 16 章 系统维护

三、本书特点

本书第一部分对软件工程的观念做介绍后,第二部分介绍了 15 个软件工程设计案例。这些案例都是作者与同行曾做过的软件工程项目,包括计算统计、软件平台、信息系统以及系统接口等方面的内容。每个案例都说明了一个应用方面。

四、适用范围

软件工程案例课程是软件系统理论与开发课程的应用创新型课程,本书是适合给开设有软件工程课程的大学高年级和低年级研究生作参考和实践课程的教材,也可作为工程技术人员的参考用书。

五、如何使用本书

根据读者的实际情况,如果读者已经掌握了软件开发的技术,那么就可以不用学习第 1、2 章。读者可以按照自己的工作目标和喜好有选择地学习相关章节。

参与本书编写的除笔者外,刘志武、王挺、陈容征、张信一、罗伟刚、文剑、李永朝、郝搏、谭炯中、何俊亮、胡志斌、王秋杰、李华、段生林、温琮韬、林玮、陈平华、陈靖宇、杨文伟、李毅坚、梁良、卢迅恒、梁锡亮、吴晓明、黎旺生、李志伟也做了部分工作。

由于软件工程知识面广,在介绍中很难面面俱到,加上时间仓促,作者水平有限,书中的不足之处在所难免,恳请读者批评指正。

编 者

2007 年 5 月

于广州小谷围岛

第 1 章 系统设计	1
1.1 设计基本原理	1
1.1.1 抽象	1
1.1.2 细化	2
1.1.3 模块化	2
1.1.4 软件体系结构	4
1.1.5 程序结构	5
1.1.6 数据结构	5
1.1.7 软件过程	7
1.2 体系结构设计	7
1.2.1 软件结构图	8
1.2.2 扇出、扇入与深度、宽度	9
1.2.3 模块的耦合	10
1.2.4 模块的内聚	12
1.2.5 结构设计的一般准则	14
1.3 结构化设计	15
1.3.1 数据流的类型	15
1.3.2 过程步骤	16
1.3.3 变换分析设计	17
1.3.4 事务分析设计	19
1.3.5 混合流设计	19
1.3.6 结构化设计方法应用示例	21
1.3.7 设计后的处理	21
1.4 详细设计	22
1.4.1 结构化语言	22
1.4.2 判定表	23
1.4.3 判定树	24

1.5	结构化程序设计	25
1.5.1	节点	25
1.5.2	3种基本控制结构	25
1.5.3	过程设计语言	26
1.6	面向对象设计	30
1.6.1	转向面向对象的设计	30
1.6.2	面向对象抽象方法	30
1.6.3	信息隐藏	31
1.6.4	模块化	31
1.6.5	类的设计准则	31
1.6.6	面向对象设计的基本原理	31
1.6.7	软件复用	32
1.6.8	面向对象设计的步骤	32
1.7	数据库系统模型和结构	32
1.7.1	层次模型	33
1.7.2	网状模型	34
1.7.3	关系模型	34
1.8	数据库管理系统	36
1.9	关系数据库的范式理论	37
1.9.1	数据依赖	38
1.9.2	关系模式的操作异常	39
1.9.3	范式	39
1.10	EAR方法	44
1.10.1	EAR图	44
1.10.2	EAR方法向关系数据库转换	46
第2章	需求分析	47
2.1	概况	47
2.1.1	企业状况	47
2.1.2	组织机构	48
2.1.3	计算机应用状况	49
2.2	企业特点	49
2.2.1	生产特点	50
2.2.2	产品特点	50
2.2.3	管理特点	51
2.2.4	生产计划特点	51
2.3	作业主要流程	52
2.3.1	业务作业流程	52
2.3.2	品保作业流程	52

2.3.3	技开作业流程	52
2.3.4	事业作业流程	53
2.3.5	财务作业流程	54
2.3.6	后勤作业流程	55
2.3.7	人事作业流程	55
2.3.8	总裁办作业流程	56
2.4	企业经营目标	56
2.5	实现目标存在的问题	56
2.5.1	新产品的开发设计	56
2.5.2	生产经营管理	57
2.5.3	质量检测	57
2.5.4	人力资源管理	57
2.5.5	财务管理	57
2.6	实现目标采取的措施	57
2.6.1	利用新技术	58
2.6.2	实现管理规范化	58
2.6.3	提高计算机应用水平	58
2.6.4	密切外界关系合作	58
2.6.5	员工教育	58
第3章	总体方案	59
3.1	系统目标	59
3.1.1	总体目标	59
3.1.2	功能目标	61
3.1.3	性能目标	64
3.1.4	效益目标	65
3.1.5	技术目标	66
3.2	确定系统方案原则	66
3.2.1	技术的先进性	66
3.2.2	经济性原则	68
3.2.3	政策性原则	68
3.2.4	系统的实用性	68
3.2.5	信息的安全性	68
3.2.6	数据库设计	68
3.2.7	软件规范原则	68
3.2.8	全面规划、分步实施	69
3.2.9	系统的适应性	69
3.3	系统结构	69
3.3.1	应用系统结构	69

3.3.2	子系统界面及技术指标	71
3.3.3	网络拓扑结构	81
3.3.4	逻辑结构	82
第4章	业务管理子系统设计	84
4.1	概述	84
4.1.1	业务管理机构	84
4.1.2	工作任务与目标	85
4.2	功能设计	86
4.3	功能组件设计	87
4.3.1	业务科处理	87
4.3.2	采购处理	91
4.3.3	仓储科处理	99
4.3.4	生控科处理	110
第5章	技术开发管理子系统设计	114
5.1	概述	114
5.2	功能设计	114
5.3	组件设计	116
5.3.1	开发作业组件设计	117
5.3.2	试作处理组件设计	130
5.3.3	技术转移组件设计	140
5.3.4	模具作业组件设计	148
5.3.5	开版作业组件设计	154
第6章	品质保证子系统设计	157
6.1	背景说明	157
6.1.1	系统目标	157
6.1.2	万里通企业集团品保系统组织图	157
6.1.3	参考资料	158
6.1.4	术语和缩写词	158
6.2	软件总概述	159
6.2.1	目标系统特点	159
6.2.2	假设和约束	159
6.2.3	品保系统与其他部门的信息传送	159
6.2.4	品保子系统内部模块关系	162
6.3	功能设计	162
6.3.1	功能介绍	162
6.3.2	品保系统详细功能树	182

6.4	性能需求	184
6.5	组件设计	184
第7章	事业管理子系统设计	185
7.1	概述	185
7.1.1	事业部管理机构图	185
7.1.2	管理职能	186
7.2	功能设计	186
7.2.1	作业管理	186
7.2.2	预防管理	188
7.2.3	监控管理	188
7.2.4	金箭管理	189
7.3	性能设计	190
7.3.1	精度	190
7.3.2	时间特性	190
7.3.3	灵活性	190
7.4	组件设计	190
7.4.1	组件功能设计	190
7.4.2	组件所用物理表	190
第8章	财务子系统设计	191
8.1	概述	191
8.2	账务	192
8.2.1	凭证处理	193
8.2.2	登账结账	196
8.2.3	会计报表	200
8.2.4	系统初始化	203
8.3	出纳事务	206
8.4	原材料核算	210
8.5	生产成本核算	212
8.5.1	农场生产成本核算	212
8.5.2	福利社成本核算	216
8.6	系统管理	218
第9章	人事管理子系统设计	221
9.1	概述	221
9.1.1	人事管理机构	221
9.1.2	工作任务与目标	221
9.2	系统设计	221

9.2.1	人事资料管理	223
9.2.2	考勤管理	225
9.2.3	考评管理	229
9.2.4	薪资管理	232
9.2.5	员工培训管理	235
第 10 章	后勤管理子系统设计	237
10.1	概述	237
10.1.1	后勤管理机构	237
10.1.2	工作任务与目标	237
10.2	系统设计	237
10.2.1	财产管理	239
10.2.2	事务性采购	241
10.2.3	设备管理	244
10.2.4	零件管理	247
10.2.5	马达维修管理	248
10.2.6	车辆管理	249
10.2.7	事务科管理	253
10.2.8	宿舍管理	254
10.2.9	图书馆管理	256
10.2.10	食堂管理	257
10.2.11	农场管理	259
10.2.12	稽核管理	260
第 11 章	数据迁移设计	262
11.1	省征管系统介绍	262
11.2	异构数据迁移途径的选择	265
11.2.1	数据转换途径的选择	265
11.2.2	数据迁移方法的选择	265
11.3	数据迁移的总体规划	266
11.3.1	数据迁移范围	268
11.3.2	提升省征管软件的数据质量	269
11.3.3	数据迁移的运行框架	270
11.3.4	新旧系统代码的对应	272
11.3.5	新旧系统流水号的对应	272
11.3.6	数据迁移的技术验证	273
11.3.7	数据迁移的业务验证	273
11.4	数据迁移的详细实现	274
11.4.1	新旧系统流水号对应的实现	274
11.4.2	新旧系统代码对应的实现	275

11.4.3	调度程序的设计	275
11.4.4	监控程序的设计	278
11.4.5	数据导出的实现	279
11.4.6	数据导入 Oracle 的实现	279
11.4.7	一个税款迁移例子的设计	281
11.5	数据迁移的实施要点	281
11.5.1	数据迁移的质量保证计划	282
11.5.2	数据迁移的正确性验证	282
11.5.3	提高数据转换的性能	282
11.5.4	提高数据导入的性能	284
第 12 章	系统平台设计	285
12.1	概述	285
12.1.1	概述	285
12.1.2	企业分布	285
12.2	数据库系统设计	286
12.2.1	数据库系统的设计目标与原则	286
12.2.2	信息特点与信息量估算	286
12.2.3	数据库概念模式设计与数据字典	288
12.2.4	数据库体系结构	288
12.2.5	数据库主要内容	290
12.2.6	数据库逻辑设计与物理设计	290
12.2.7	数据库管理系统的选型	290
12.3	网络系统设计	293
12.3.1	网络负荷与容量分析	293
12.3.2	网络系统设计目标与原则	293
12.3.3	网络系统的需求	294
12.3.4	网络结构设计	296
12.3.5	网络系统管理	301
12.3.6	机房工程	303
12.3.7	网络系统配置	305
12.3.8	结构布线方案	306
12.3.9	实施计划	310
12.3.10	经费预算	313
第 13 章	系统组织与信息分类编码标准	319
13.1	概述	319
13.2	编码内容	321
13.2.1	数据元素标准	321

13.2.2	用户视图标准	321
13.2.3	概念数据库标准	322
13.2.4	逻辑数据库描述标准	323
13.2.5	数据库结构标识	324
13.2.6	功能结构描述标准	324
13.2.7	系统结构描述标准	325
13.2.8	系统文件命名原则与标准	325
13.2.9	报告编写的组织	328
13.2.10	信息分类编码标准	332
第 14 章	接口	344
14.1	外部接口	344
14.2	系统结构接口	344
14.3	子系统间的接口	345
14.4	人机接口	368
第 15 章	关键技术与实施计划	370
15.1	关键技术	370
15.2	技术路线	374
15.3	实施计划与进度	374
第 16 章	系统维护	376
16.1	概述	376
16.2	设备维护	376
16.3	数据库维护	378
16.4	安全维护	379
16.5	日常维护	381
16.6	防病毒维护	382
参考文献	383

系统设计

系统设计的基本目标就是概要地回答系统应该如何实现。所以在任何工程产品或系统中,设计是开发阶段的第一步。设计(design)可以定义为应用各种技术和原理,对一个设备、一个过程或一个系统,作出足够详细的决策,使之在物理上得以实现的过程。

1.1 设计基本原理

软件设计要回答下列问题:

- (1) 使用什么样的准则才能把软件划分成为各个单独的构件?
- (2) 怎样把功能或数据结构的细节从软件概念表达式中分离出来?
- (3) 定义软件设计的技术质量有统一的标准吗?

软件设计中需要考虑的最重要的一个问题就是软件质量问题,用什么标准衡量软件设计的技术质量呢?下面介绍软件发展中应用并经过时间考验的软件设计的一些基本原理。

1.1.1 抽象

抽象是认识复杂现象过程中使用的思维手段,即抽出事物本质的共同特性而暂不考虑它的细节,不考虑其他因素。抽象的概念被广泛应用到软件工程学中。在进行软件设计时,抽象与逐步求精、模块化密切相关,可帮助定义软件结构中模块的实体,由抽象到具体地分析和构造出软件的层次结构,提高软件的可理解性。

当考虑用模块化的方法解决问题时,可以提出不同层次的抽象(levels of abstraction)。在抽象的最高层,可以使用问题环境的语言,以概括的方式叙述问题的答案。在抽象的较低层,则采用更过程化的方法,在描述问题的答案时,面向问题的术语与面向实现的术语结合使用。最终,在抽象的最底层,可以用直接实现的方式来说明。软件工程实施中的每一步都可以看作是对软件抽象层次的一次细化。

随着对抽象不同层次的展开,过程抽象(procedural abstraction)和数据抽象(data abstraction)就建立了。过程抽象是指一个命名的指令序列,它具有一个特定的和受限的功能,例如有一个进入某场合的词“入口”,对这个词进行分析会发现,其中隐含了走到门口、伸出手、握住门把、旋转门把和推门、走进门的一系列的程序序列。数据抽象则是一个命名的

说明数据对象的数据集合,例如一个部门员工的“工资单”。这个数据对象实际上是许多不同方面信息:单位、姓名、工资总额、扣除房租、水电费、煤气费、电话费、电视费、实得金额等的集合。在说明这个数据抽象名时,指的是所有数据。控制抽象(Control abstraction)是软件设计中第3种抽象形式。像过程抽象和数据抽象一样,控制抽象隐含了程序控制机制,而不必说明它的内部细节。控制抽象的例子是操作系统中用于进程协调活动的同步信号标。

许多编程语言(如 Ada, Modula, CLU)都给出了建立抽象数据类型的机制(mechanisms)。例如,Ada的包(package)就是一种支持数据抽象和过程的编程语言机制。这种最初的抽象数据类型可以用作模板(template)或类属(generic)数据结构,由此而导出的其他数据结构可以是它们的实例。

1.1.2 细化

逐步细化是一种自顶向下的设计策略。程序的体系结构开发是由过程细节层次不断细化而成的。分层的开发则是以逐步的方式由分解一个宏功能直到获得编程语言语句。

在细化的每一步,已给定的程序的一条或几条指令被分解为更多细节的指令。当所有指令按计算机或编程语言写成时,这样不断的分解或规格说明的细化也将终止。随着任务的细化,数据也要细化、分解或结构化。程序的细化和数据的说明并行进行,这是很自然的事。

每一步细化都隐含着一定的设计决策,重要的是程序员应当了解一些最基本的准则和存在的可选方案。

细化实际上是一个详细描述(elaboration)的过程。在高层抽象定义时,从功能说明或信息描述开始,就是说明功能或信息的概念,而不给出功能内部的工作细节或信息的内部结构。细化则是设计者在原始说明的基础上进行详细说明,随着不断的细化(详细说明)而给出更多的细节。

1.1.3 模块化

在计算机软件中,几乎所有的软件体系结构都要体现模块(module)化。也就是说,所有的软件结构设计技术都是以模块化为基础的。模块是由单独命名和可编程的构件集成,以满足问题的需求。

模块化的概念在程序设计技术中就出现了。模块在程序中是数据说明、可执行语句等程序对象的集合,或者是单独命名和编址的元素,如高级语言中的过程、函数和子程序等;在软件的体系结构中,模块是可组合、分解和更换的单元。模块具有以下几种基本属性。

- (1) 接口:指模块的输入与输出。
- (2) 功能:指模块实现什么功能。
- (3) 逻辑:描述内部如何实现要求的功能及所需的数据。
- (4) 状态:指该模块的运行环境,即模块的调用与被调用关系。

功能、状态与接口反映模块的外部特性,逻辑反映它的内部特性。

模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。每一个模块完成一个特定的子功能,所有的模块按某种方法组织起来,成为一个整体,完成整