

高职高专 计算机专业

GAOZHI GAOZHUAN
JISUANJI ZHUANYE XILIE JIAOCAI 系列教材

杨文元 编著

鄂大伟 主审

软件工程 与实践



厦门大学出版社
XIAMEN UNIVERSITY PRESS

软件工程与实践

杨文元 编著

鄂大伟 主审

厦门大学出版社

图书在版编目(CIP)数据

软件工程与实践/杨文元编著. —厦门:厦门大学出版社, 2005. 5

ISBN 7-5615-2384-X

I. 软… II. 杨 III. 软件工程-高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 035045 号

厦门大学出版社出版发行
(地址:厦门大学 邮编:361005)

<http://www.xmupress.com>

[xmup @ public. xm. fj. cn](mailto:xmup@public.xm.fj.cn)

南平市武夷美彩印中心印刷

2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

开本:787×1092 1/16 印张:18.25

字数:463 千字 印数:0 001-2 000 册

定价:27.00 元

本书如有印装质量问题请直接寄承印厂调换

总序

计算机的快速发展不过几十年的时间,而它的应用已渗透到各行各业。可以说,计算机的知识与技能已经成为现今人才构成的最重要要素之一。顺应时代要求,计算机教育也在迅猛地发展。

高职高专的计算机教育不同于本科,有着自己的特点。既需要一定的理论基础,同时理论又不能太过深入;学生注重计算机应用技术与能力,需要熟悉各种应用软件,特别是办公软件、数据库等。市场上各种高职高专计算机教材繁多而庞杂,可真正适合教学的却很难寻觅。为了推进福建省计算机教材的建设与改革,提高福建省高职高专计算机教学水平,我们和厦门大学出版社一起,组织编写了这套高职高专的计算机专业系列教材。本系列教材目前包括《计算机导论》、《电路电子学基础》、《计算机数学基础》、《数据结构》、《微机原理与接口技术》、《操作系统与应用实践》、《办公软件使用与操作》、《计算机网络基础》、《Web技术与网页制作》、《SQL数据库应用基础》、《软件工程与实践》、《Visual FoxPro7.0程序设计实训教程》12种。其中,既包括基础的“长线”课程,又有高职高专学生必须掌握的“短线”课程。以后随着计算机科学的发展,其他计算机教材也会陆续补充进来,将会逐步形成一个较为完整的计算机类高职高专教材体系。

本套教材是依照高职高专计算机课程的教学特点而编写的。编写之前,厦门大学出版社做了大量的调研工作,考察了各校的教学实际。在具体编写过程中,厦门大学出版社负责联络各参编学校,组织召开教材编写会议;福建省计算机学会出版与高等教育工作委员会负责审定各教材的编写大纲,并在编写过程中给予具体指导。具体编写过程是:组建了福建省高职高专计算机专业教材编委会,各教材也成立一编写小组,设立一主编及主审,主编由对该领域非常熟悉,有着丰富教学经验者担任,主审由福建省计算机学会出版与高等教育工作委员会负责推荐或指定,亦是相关领域的专家、教授;编写者是从事教学多年的一线骨干教师,他们熟悉高职高专的教学规律和特点,具有丰富的教学实践;主编负责联系各参编老师,按各校教学计划,编写符合高职高专教学实际的大纲,大纲经主审审定后再编写各章节,完稿后主编统稿,同时主审审阅,根据主审意见再讨论修改,直至定稿。整个过程有条不紊,周密而审慎。

高职高专教育是今后一个时期高等教育发展的重点,正如火如荼地发展着,而加强高职高专教材建设为它的发展提供了保证。所以说,我们编好这套教材可谓意义深远,责任重大。在这里,我谨向为撰写、审阅和编辑出版该系列教材付出辛勤劳动,做出重要贡献的各位专家和厦门大学出版社编辑出版的同志致以衷心的感谢!也愿福建省高职高专的计算机教育更上一层楼!

前 言

本书为高职高专计算机专业系列教材之一。

软件工程是一门年轻的学科，在计算机领域中的地位越来越重要，不仅在大型软件项目开发中离不开它，就是一般小型软件项目也必须运用它的基本概念、过程方法。软件系统分析员，软件设计、编程、测试和维护人员以及软件管理人员都必须掌握软件工程的知識，才能适应岗位工作的要求。实践已经证明，不掌握软件工程知识，不按照软件工程过程方法的要求去开发软件，只能得到质量差、水平低和难于维护的软件产品。

本教材由漳州职业技术学院杨文元副教授编写。在编写中把握“理论够用，重在实践实训”的原则，并贯穿“理论讲清讲透，实践尽可能多操作”的教学思想。软件工程的理论性和实践性均很强，而且发展变化速度快，为了让读者对软件工程有全面了解并掌握最新的软件工程知识，将教材的内容分为四部分：第一部分是传统软件工程的基本知识，虽然传统的方法可能已较少使用，但认识软件工程的整个发展过程对学习软件工程仍有积极意义；第二部分是面向对象内容，有面向对象的分析设计和实现、统一开发过程和统一建模语言 UML 以及用 UML 构建 Web 应用等内容；第三部分是软件项目管理，包括软件开发项目的管理和度量方法、软件项目维护、软件的质量保证以及极限编程方法等内容；第四部分是软件开发工具和软件工程实践，介绍 UML 实现工具 Rational Rose 的使用、软件分析设计工具 PowerDesigner 的使用以及用 UML 设计图书馆图书借阅和预定管理的应用程序等，可供实践实训教学使用。

由于软件工程理论有一定的难度，所以学生在学习会觉得有一些枯燥，为此在书中安排了若干小故事与读者共享，但愿能减缓读者学习之乏，并在一笑之后对故事所含道理能有所“悟”。

集美大学鄂大伟教授对本教材进行审稿，从大纲编写到书稿编写都提出了很好的指导意见和详细的修改意见；福建农林大学宁正元教授、厦门大学薛永生教授、南京理工大学刘传才教授、福建师范大学陈利永教授等对本书的编写给予了大力支持和帮助；厦门大学出版社宋文艳编审、眭蔚编辑为本教材的出版给予全力支持并做了大量工作。作者谨对他们热忱的帮助致以衷心的感谢。

本教材是在漳州职业技术学院院长朱国辉教授关怀下完成的，教务处康玉文处长、蔡晓雯副处长、电子系林绍中主任和方春春副主任对本教材的编写给予了大力支持，还有很多领导和同事也给予了很多帮助，在此表示作者真诚的谢意。

限于作者的学识和水平，书中错误之处敬请广大读者不吝指正。

本教材有电子教案，需要者请联系：chinaywy@126.com。

作者

2005年6月8日

目 录

总序
前言

第一篇 传统软件设计技术

第 1 章 软件工程概述	3
1.1 软件的概念及其特点	3
1.1.1 软件的定义	3
1.1.2 软件分类	4
1.1.3 软件的特点	4
1.2 软件工程	5
1.2.1 软件危机的表现及产生的原因	5
1.2.2 软件工程概念	7
1.2.3 软件工程的基本原理	8
1.3 软件生存周期和软件开发模型	9
1.3.1 软件生存周期	9
1.3.2 软件生存期模型	10
1.4 程序设计方法和软件开发方法	11
1.4.1 程序设计方法	12
1.4.2 软件开发方法	13
复习思考题	15
第 2 章 可行性分析和软件项目开发计划	16
2.1 可行性分析的任务和步骤	16
2.1.1 可行性分析的任务	16
2.1.2 可行性分析的步骤	17
2.2 成本/效益分析	18
2.2.1 成本估计方法	19
2.2.2 费用估算	20
2.2.3 几种度量效益的方法	21
2.3 软件项目开发计划	22
2.4 软件计划进度安排	23
2.4.1 项目计划与里程碑	23
2.4.2 项目计划安排技巧	24
复习思考题	25
第 3 章 软件需求分析	27

3.1 需求分析概述	27
3.1.1 需求分析的任务	27
3.1.2 需求分析的过程	28
3.1.3 需求分析的困难性	28
3.1.4 需求的目的	29
3.2 结构化分析方法和工具	30
3.2.1 结构化分析概述	31
3.2.2 数据流图	31
3.2.3 数据词典	34
3.2.4 加工逻辑描述工具	35
3.3 原型化方法	37
3.3.1 软件原型的分类	37
3.3.2 快速原型开发模型	37
复习思考题	39
第4章 软件设计	40
4.1 总体设计概述	40
4.1.1 总体设计原则	41
4.1.2 总体设计过程	42
4.1.3 总体设计的图形工具	42
4.2 软件设计	43
4.2.1 模块化	43
4.2.2 抽象	45
4.2.3 信息隐藏	45
4.2.4 模块独立性	45
4.2.5 模块的深度、宽度、扇出和扇入	48
4.3 详细设计	49
4.3.1 详细设计概述	49
4.3.2 结构化程序设计	50
4.4 详细设计工具	51
4.4.1 程序流程图	51
4.4.2 盒图(N-S图)	52
4.4.3 PAD(problem analysis diagram)图	52
4.4.4 过程设计语言PDL(program design language)	53
4.5 面向数据结构的设计方法	54
4.5.1 Jackson图	54
4.5.2 Jackson程序设计方法	55
复习思考题	57
第5章 程序编码与软件测试	59
5.1 程序设计语言	59
5.1.1 语言的分类和特点	59

5.1.2 常用开发语言的评价	61
5.1.3 开发语言的选择	62
5.2 程序编写规范	63
5.2.1 编码风格	63
5.2.2 结构化编码	64
5.2.3 程序清晰性	64
5.2.4 变量和表达式	65
5.2.5 输入和输出	65
5.2.6 程序效率	65
5.2.7 程序注释	66
5.2.8 成熟技术的使用与“创新”	66
5.3 程序员的修养	67
5.3.1 程序员的素质	67
5.3.2 编程修养	67
5.4 软件测试概述	68
5.4.1 软件测试及其发展	68
5.4.2 软件测试角色	69
5.5 软件测试的方法	70
5.5.1 bug 和 bug 跟踪	70
5.5.2 测试的分类	70
5.6 软件测试用例的设计	73
5.6.1 测试用例文档	73
5.6.2 白盒测试和黑盒测试	73
复习思考题	75

第二篇 面向对象分析设计方法

第 6 章 面向对象方法	79
6.1 面向对象的概述	79
6.1.1 面向对象的发展历史和动因	79
6.1.2 面向对象的概念	81
6.2 面向对象的软件工程	83
6.2.1 面向对象的分析	84
6.2.2 面向对象的设计	86
6.2.3 面向对象的实现	91
6.3 面向对象的建模方法	92
6.3.1 Booch 方法	92
6.3.2 Rumbaugh 的 OMT 方法	93
6.3.3 Coad/Yourdon 方法	94
6.3.4 Jacobson 方法	94



复习思考题	95
第 7 章 软件开发的统一过程 UP	96
7.1 统一过程 UP	96
7.1.1 UP 的概念	96
7.1.2 UP 和 Rational 统一过程 RUP	97
7.1.3 为项目实例化 UP	98
7.2 统一过程的开发方法	98
7.2.1 用例驱动和风险驱动	98
7.2.2 以体系结构为中心	99
7.2.3 UP 是迭代和增量过程	99
7.2.4 UP 结构	101
7.2.5 UP 阶段	102
复习思考题	104
第 8 章 统一建模语言 UML 简介	105
8.1 UML 概述	105
8.1.1 UML 简介	105
8.1.2 UML 历史	106
8.2 UML 语言基础	108
8.2.1 UML 语言描述	108
8.2.2 元素关系	109
8.3 UML 的静态建模	110
8.3.1 用例图	110
8.3.2 类图、对象图和包	112
8.3.3 构件图和配置图	117
8.4 动态建模	118
8.4.1 消息	118
8.4.2 状态图	118
8.4.3 顺序图	119
8.4.4 合作图	121
8.4.5 活动图	121
8.4.6 四种图的运用	122
8.5 使用 UML 的过程	122
8.5.1 UML 过程的基础	123
8.5.2 面向对象方法的一般过程	123
8.5.3 Rational Objectory 过程	126
复习思考题	127
第 9 章 用 UML 构建 Web 应用	128
9.1 Web 应用概述	128
9.1.1 Web 工程过程	128
9.1.2 Web 开发团队	129

9.2 UML 构建数字商店	130
9.2.1 概况	130
9.2.2 需求和特性	131
9.2.3 软件构架文档	131
复习思考题	139

第三篇 软件开发的项目管理

第 10 章 软件开发的项目管理	143
10.1 软件项目管理概述	143
10.1.1 项目管理中的均衡三角形	143
10.1.2 项目均衡	144
10.2 软件开发团队	144
10.2.1 开发团队构成	145
10.2.2 如何组织软件开发团队	147
10.2.3 项目经理素质要求	149
10.3 项目的开发规范	150
10.3.1 软件工程的标准	150
10.3.2 标准化的意义	150
10.4 软件项目的成功原则	151
10.4.1 平衡原则	151
10.4.2 高效原则	151
10.4.3 分解原则	151
10.4.4 实时控制原则	152
10.4.5 分类管理原则	152
10.4.6 简单有效原则	152
10.4.7 规模控制原则	152
10.5 微软公司的开发管理和演变	153
复习思考题	153
第 11 章 软件过程和项目的度量方法	154
11.1 软件过程的度量	154
11.1.1 过程度量	154
11.1.2 软件过程的改善	155
11.2 软件项目的度量概述	156
11.3 软件项目的度量方法	156
11.3.1 面向规模的度量	157
11.3.2 面向功能的度量	158
11.3.3 软件质量的度量	160
11.4 风险分析和管理	161
11.4.1 被动和主动的风险策略	162



11.4.2 软件风险	162
11.4.3 软件风险的估计	163
11.4.4 风险的监控管理	163
复习思考题	164
第 12 章 软件的质量保证	165
12.1 软件质量保证概述	165
12.1.1 软件质量保证的概念和任务	165
12.1.2 质量保证和检验	165
12.1.3 软件的质量成本	166
12.2 软件质量保证体系与实施	167
12.2.1 软件质量保证体系	167
12.2.2 软件质量保证的实施	168
12.3 软件配置管理	169
12.3.1 软件配置项管理	169
12.3.2 软件配置管理的过程和任务	170
12.4 软件工程标准化	171
12.4.1 软件工程标准的制定	171
12.4.2 软件工程标准的层次	172
12.4.3 我国标准	173
12.5 软件过程能力评估的 CMM 模型	174
12.5.1 软件机构的过程成熟度模型 CMM	175
12.5.2 CMM 的评估	175
12.5.3 国内 IT 企业采用 CMM 的状况	176
复习思考题	176
第 13 章 极限编程	177
13.1 极限编程概述	177
13.1.1 什么是极限编程	177
13.1.2 极限编程的特点	178
13.1.3 极限编程的设计原则	180
13.2 极限编程方法	181
13.2.1 规划策略	181
13.2.2 小型发布	182
13.2.3 系统比喻	182
13.2.4 简化设计	182
13.2.5 测试驱动	183
13.2.6 重构	183
13.2.7 结对编程	184
13.2.8 代码全体拥有	184
13.2.9 持续集成	185
13.2.10 每周 40 小时工作制	185

13.2.11 现场客户	185
13.2.12 代码规范	186
13.3 极限编程 XP 与 CMM/RUP 的比较	186
13.3.1 XP 与 CMM 的比较	186
13.3.2 XP 与 RUP 的比较	187
复习思考题	188
第 14 章 软件的维护与再工程	189
14.1 软件的维护	189
14.1.1 软件维护的概念	189
14.1.2 影响软件维护工作量的因素	190
14.1.3 软件维护的策略	191
14.1.4 软件维护的成本	191
14.1.5 软件的可维护性	192
14.1.6 提高软件可维护性的方法	193
14.2 软件再工程和逆向工程	196
14.2.1 业务过程再工程 (business process reengineer, BPR)	196
14.2.2 软件再工程	197
14.2.3 逆向工程	200
14.3 重构	201
14.3.1 代码重构	201
14.3.2 数据重构	202
复习思考题	202

第四篇 软件开发工具和软件工程实践

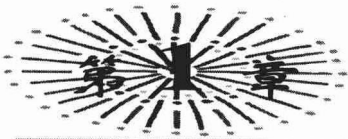
实践 1 Rational Rose 的使用	205
1.1 Rational Rose 界面	205
1.2 Rational Rose 用例视图	206
1.2.1 创建角色	206
1.2.2 创建用例	209
1.2.3 创建角色和用例之间的关联	209
1.2.4 创建顺序图和合作图	211
1.3 Rational Rose 逻辑视图	213
1.3.1 建立类图	213
1.3.2 建立包	216
1.2.3 建立状态图	216
1.4 Rational Rose 组件视图	217
1.5 Rational Rose 生成代码框架	218
1.6 新一代建模工具 XDE	220
实践 2 PowerDesigner 的使用	221

2.1 PowerDesigner 简介	221
2.1.1 PowerDesigner 概述	221
2.1.2 PowerDesigner 版本	222
2.1.3 PowerDesigner 模块组成	222
2.1.4 PowerDesigner 的 4 种模型文件	223
2.2 PowerDesigner 的操作	224
2.2.1 PowerDesigner 系统界面	224
2.2.2 PowerDesigner 基本操作	225
2.2.3 PowerDesigner 业务规则	228
2.2.4 产生实体	229
2.2.5 将 CDM 对象转换为 PDM 对象	230
2.3 正向工程和逆向工程	231
2.3.1 正向工程	231
2.3.2 逆向工程	232
2.4 生成报表	234
实践 3 用 UML 设计图书管理系统	236
3.1 需求分析	236
3.2 系统分析	236
3.2.1 总体分析	237
3.2.2 域分析	238
3.3 设计	240
3.3.1 体系结构设计	241
3.3.2 详细设计	242
3.3.3 用户界面设计	245
3.4 实现	245
3.5 测试和部署	246
3.6 小结	246
附录 软件工程文档模板	253
参考文献	275

史前史中，没有别的场景比巨兽在焦油坑中垂死挣扎的场面更令人震撼。上帝见证着恐龙、猛犸象、剑齿虎在焦油中挣扎。它们挣扎得越猛烈，焦油纠缠得越紧，没有任何猛兽足够强壮或具有足够的技巧能够挣脱束缚，它们最后都沉到了坑底。

过去几十年的大型系统开发就犹如这样一个焦油坑，很多大型和强壮的动物在其中剧烈地挣扎。他们中大多数开发出了可运行的系统——不过，其中只有非常少的项目满足了目标、时间进度和预算的要求。各种团队，大型的和小型的，庞杂的和精干的，一个接一个淹没在了焦油坑中。表面上看起来好像任何一个单独的问题都不会导致困难，每个都能解决，但是当它们相互纠缠和累积在一起的时候，团队的行动就会变得越来越慢。对问题的麻烦程度，每个人似乎都会感到惊讶，并且很难看清问题的本质。不过，如果我们想解决问题，就必须试图先去理解它。

——Frederick P. Brooks, Jr. 《人月神话》



软件工程概述

本章主要介绍软件工程的基本概念、内容、性质、特点、目标，软件生产发展的阶段，以及软件危机的产生及其表现形式，初步了解各种生存周期模型和开发方法，其中一些开发方法和模型将在后续章节中进一步详细介绍。

1.1 软件的概念及其特点

1.1.1 软件的定义

1. 计算机软件

什么是软件？计算机及其应用发展到今天，对这个问题的普遍看法是：计算机软件就是计算机系统上的程序、数据及有关文档。软件与硬件构成计算机系统，两者相互依存，缺一不可。

程序是指计算机任务中的处理对象和处理规则的描述，或者说程序是按事先设计的功能和性能需求执行的指令序列。程序并不就是软件，只是软件的一个组成部分；数据是程序能正常操纵的信息及信息结构；文档是与程序开发、使用和维护有关的图文材料。程序、数据和有关文档三个部分有机结合形成计算机软件。

2. 计算机软件的作用

计算机软件是用户与计算机硬件的接口，是计算机系统的指挥者和管理者，实现用户所需的各种计算机管理应用。

3. 软件的发展过程

第一阶段：程序设计阶段（1946—1956 年）——从第一台计算机上的第一个程序的出现到实用的高级程序设计语言出现之前。

第二阶段：程序系统阶段（1956—1968 年）——从实用的高级程序设计语言出现到软件工程出现之前。

第三阶段：软件工程阶段（1969 年以后）——“软件工程”学科诞生和发展。

第四阶段：软件产业化阶段（20 世纪 80 年代中期以后）——软件工程仍在发展和完善，但软件产业在世界经济中已经占有举足轻重的地位。

1.1.2 软件的分类

1. 按软件的功能划分

(1) 系统软件。主要有操作系统、数据库管理系统、设备驱动程序、通信处理程序等。

(2) 支撑软件。主要有文本编辑程序、文件格式化程序、程序库系统，支持需求分析、设计、实现、测试和支持管理的软件等。

(3) 应用软件。主要有商业数据处理软件、工程与科学计算软件、计算机辅助设计制造软件、系统仿真软件、智能产品嵌入软件，医疗、制药软件，事务管理办公自动化软件、计算机辅助教学软件等。

2. 按软件工作方式划分

- 实时处理软件；
- 分时软件；
- 交互式软件；
- 批处理软件。

3. 按软件规模划分

表 1-1 列出了软件的规模。

表 1-1 软件规模

软件类别	参加人数	开发时间	源代码行数
微型	1	1 个月以内	1 000 行以内
小型	1	1~6 个月	1 000~2 000 行
中型	2~5	1~2 年	5 000~50 000 行
大型	5~20	2~3 年	5 万~10 万行
特大型	100~1 000	4~5 年	几十万行
巨型	2 000~5 000	5~8 年	上百万行

另外，如果按软件服务对象的范围划分，可分为项目软件和产品软件；按使用的频度进行划分，可分为一次使用和频繁使用；按软件失效的影响进行划分，可分为高可靠性软件和一般可靠性软件。还可以有其他分法，此处不一一列举。

1.1.3 软件的特点

计算机软件具有与计算机硬件或其他商品不同的特点，主要有：

(1) 软件是一种逻辑实体，具有抽象性。这个特点使它与其他工程对象有着明显的差异。人们可以把它记录在纸上、内存、磁盘和光盘上，但却无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的功能、性能等特性。

(2) 软件没有明显的制造过程，一旦研制开发成功，就可以大量拷贝同一内容的副本。所以对软件的质量控制，必须着重在软件开发方面下工夫。

(3) 软件在使用过程中没有磨损、老化的问题。软件在生存周期内不会因为磨损而老