

21 世纪高等院校计算机系列教材



□ 熊才权 杨舒 主编

软件工程



华中科技大学出版社

TP311.5
124

21 世纪高等院校计算机系列教材

软 件 工 程

主 编 熊才权 杨 舒

副主编 陈建峡 陈永辉 李志辉

华中科技大学出版社

图书在版编目(CIP)数据

软件工程/熊才权 杨 舒 主编

武汉:华中科技大学出版社,2005年3月

ISBN 7-5609-3309-2

I. 软…

II. ①熊… ②杨…

III. 软件工程-高等学校-教材

IV. TP311.5

软件工程

熊才权 杨 舒 主编

责任编辑:彭保林 曾 光

封面设计:刘 卉

责任校对:刘 竣

责任监印:熊庆玉

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87557437

印 刷:仙桃市新华印刷厂

开本:787×1092 1/16

印张:16

字数:286 000

版次:2005年3月第1版

印次:2005年3月第1次印刷

定价:24.00元

ISBN 7-5609-3309-2/TP·545

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书全面、系统地介绍了软件工程的观念、技术与方法。全书共 12 章，第 1 章讲述软件工程的根本观念和软件开发模型，第 2 章到第 8 章分别介绍软件生命周期各阶段的基本任务、过程和方法，其他章节分别介绍面向对象的方法、软件工程管理 and 软件工程标准化等。本书紧跟软件工程学科的最新发展脚步，强调理论与实践相结合。书中配有丰富的例题与习题，便于教学与自学。

本书既可作为高等院校的教材，也可作为 IT 产业从业人员的学习参考资料。

“21 世纪高等院校计算机系列教材”丛书编委会

主任 何炎祥

委员 (按姓氏拼音排序):

戴光明	都志辉	桂 超	金先级
柯敏毅	李康顺	李克清	李禹生
刘腾红	卢强华	陆 迟	吕顺营
沈海波	石 清	王江晴	王伟军
王 忠	叶骏民	余敦辉	湛为芳

序 言

21世纪是信息时代,以计算机为核心的信息技术是21世纪科技发展的大趋势。作为计算机专业人才培养基地的大学计算机专业和相关专业,如何适应这种发展,培养出符合时代要求和社会欢迎的人才,是近年来计算机教育界讨论的热门话题,也是我们长期思考并努力探索的课题。

教材是人才培养的基础。在华中科技大学出版社的委托下,我们组织了有关高等院校的部分专家、教授共同编写了这套“面向21世纪计算机系列教材”,以期在适应21世纪的教材建设方面做出自己的努力。由于计算机行业发展日新月异,“21世纪计算机系列教材”编委会将负责系列教材的选题、每本教材大纲的编写和审定,以及教材、教学辅导书和课件的修订、更新等工作,以确保教材的正确性和先进性,使这套教材努力走在同类教材的前列。

这套系列教材包括计算机专业课和部分专业基础课教材,以及与之配套的实践课教材和教学辅导书等等。

我们希望这套教材具有以下特点:

1. 注重基础性和先进性的结合。计算机学科的一个显著特点就是知识和技术更新快,这对教学内容、课程知识结构的选取和组织提出了新的要求。我们把编写的重点放在基础知识、基本技能和基本方法上,希望在提高学生的理论素养和分析问题、解决问题的能力同时,注重介绍新的技术和方法,以拓展学生的知识面,激发他们学习的积极性和创新意识。
2. 注重理论性与应用性的结合。良好的理论素养是应用的前提,而掌握理论的目的就是为了更好的应用。在教材的编写过程中,我们注意理论的系统性,在讲深讲透主要知识的基础上,融理论性和应用性于一体,注意基本方法的讲授,以培养学生应用理论和技术的能力。
3. 注重时代性和实用性的结合。力求精简旧的知识点,增加新的知识点,体现教材的时代特征。而且充分考虑一般高校目前所拥有的师资条件和教学设备,注重教材的实用性。
4. 注重科学性与通俗性的结合。概念、原理、新技术的阐述力求准确、精练;写作上尽量通俗易懂、深入浅出、图文并茂,增强可读性,便于学生自学。
5. 网络技术辅助教学。针对本系列教材我们开发有专门的网站(<http://www.hzpress.org>)、课件发布演示系统和考试系统等,以便为任课老师的教学提供更便捷、更全面的服务,并将通过网站开展各种形式的教材网上专家答疑、内容修订发布、课件定期升级等活动,以与读者随时互动,为读者提供立体化的服务。

教学改革是需要不断探索的课题。要达到以上目标，还需要不断地努力实践和完善。欢迎使用这套教材的教师、学生和其他读者提出宝贵意见。

最后，对参加这套教材编写的所有作者，对为这套教材的编写提供支持的有关学校、院系的领导和老师表示诚挚的谢意！感谢华中科技大学出版社为本系列教材的出版所付出的辛勤劳动！

教材编委会主任 **何炎祥**

(教授、博导、武汉大学计算机学院院长)

前 言

自 20 世纪 60 年代末以来,人们为克服软件危机做了大量研究工作,逐渐形成了系统的软件开发理论、技术和方法,它们在软件开发实践中发挥着重要作用。今天,软件工程作为一个正式的工程学科,已经为业界所广泛接受。实践表明,不按照软件工程的要求进行软件开发,提供给用户的只能是低质量、低可靠性、难以维护的软件产品。

近年来,软件工程学科发展日新月异,新的思想、技术和方法不断涌现。现在不少高校已经将软件工程作为一个专业开设,已经发展成为一门单独的课程。“软件工程”是高等学校计算机教学计划中的一门核心课程,作为一门概论性的课程,要在内容上做出合适的取舍,实属不易。本教材的指导原则是,针对本科教学特点,既要继承传统的软件工程概念、工具和方法,同时又要紧跟软件工程的最新发展,介绍新的理论和技术。

全书共 12 章。第 1 章讲述软件工程的基本思想和软件开发模型,引出了软件生命周期的概念,是后面章节学习的基础。第 2 章到第 8 章分别介绍软件生命周期各阶段的基本任务、过程和方法,它们基于传统的结构化方法,内容包括需求分析、总体设计、详细设计、编码、测试和维护。第 9 章介绍面向对象的方法,内容包括面向对象需求分析、面向对象设计和面向对象实现。第 10、11、12 章分别对软件度量、软件工程管理、软件工程标准化做了详细介绍,它们与前几章一起,共同构造了软件工程的管理、工具和技术三大方面的内容。

本书内容满足国家教育委员会颁布的计算机专业软件工程课程教学基本要求,可以作为高等学校工科本科各专业软件工程课程的教材或教学参考书,也适合软件开发人员与软件项目管理人员参考。全书计划学时数为 40,其中实验学时数为 10,授课教师可以根据实际情况和教学的具体要求,进行适当的取舍。

本书由熊才权、杨舒担任主编,陈建峡、陈永辉、李志辉担任副主编,第 1、4、6、7、11 章由熊才权编写,第 5、9、10、12 章由杨舒编写,第 2、3 章由陈建峡编写,第 8 章由陈永辉编写,熊才权负责全书的统稿工作,李志辉负责稿件的审核工作。

由于水平有限且时间紧迫,错误及不当之处在所难免,敬请广大读者和同行批评指正。

编 者

2004 年 12 月

目 录

第 1 章 软件工程概述	(1)
1.1 软件	(1)
1.1.1 软件与程序	(1)
1.1.2 软件的分类	(1)
1.1.3 软件的特征	(2)
1.1.4 软件开发技术的发展	(3)
1.2 软件工程	(4)
1.2.1 软件危机	(4)
1.2.2 软件工程的定义	(6)
1.2.3 软件工程的目标	(6)
1.3 软件生命周期	(7)
1.3.1 软件生命周期的概念	(7)
1.3.2 软件生命周期各阶段的基本任务	(7)
1.4 软件开发模型	(10)
1.4.1 瀑布模型	(10)
1.4.2 原型模型	(11)
1.4.3 螺旋模型	(12)
1.4.4 基于第四代技术的模型	(13)
1.4.5 形式化方法模型	(14)
小结	(14)
习题	(15)
第 2 章 需求分析	(16)
2.1 可行性研究	(16)
2.1.1 可行性研究的任务	(16)
2.1.2 可行性研究的步骤	(18)
2.2 需求获取	(20)
2.2.1 需求获取的内容	(20)
2.2.2 需求获取的技术	(22)
2.2.3 需求分析的原则	(23)
2.3 开发原型系统	(25)
2.4 需求分析规格说明书	(25)
2.5 需求验证	(27)
小结	(29)
习题	(29)

第3章 分析建模	(31)
3.1 数据建模	(31)
3.1.1 基本概念	(32)
3.1.2 实体关系图	(33)
3.2 功能建模	(35)
3.2.1 数据流图的基本成分	(35)
3.2.2 数据流图的画法	(37)
3.2.3 画数据流图的注意事项	(39)
3.3 数据字典	(41)
3.3.1 数据字典的成分	(41)
3.3.2 数据字典的用途	(44)
3.4 加工小说明	(45)
3.4.1 结构化语言	(46)
3.4.2 判定表	(46)
3.4.3 判定树	(47)
小结	(48)
习题	(48)
第4章 总体设计	(50)
4.1 总体设计的过程	(50)
4.2 表示软件结构的图形工具	(52)
4.2.1 层次图和 HIPO 图	(52)
4.2.2 结构图	(53)
4.3 软件设计的概念和原理	(54)
4.3.1 模块化	(55)
4.3.2 抽象	(56)
4.3.3 信息隐藏	(57)
4.3.4 模块独立	(57)
4.4 软件结构设计优化准则	(60)
4.5 面向数据流的设计方法	(62)
4.5.1 两类基本的数据流图	(62)
4.5.2 面向数据流的设计方法的步骤	(63)
4.5.3 变换分析	(64)
4.5.4 事务分析	(66)
4.5.5 设计优化	(68)
小结	(68)
习题	(69)
第5章 详细设计	(71)
5.1 详细设计概述	(71)
5.2 结构化程序设计	(72)

5.3 详细设计工具	(73)
5.3.1 程序流程图	(73)
5.3.2 N-S 图(盒状图)	(74)
5.3.3 问题分析图	(75)
5.3.4 过程设计语言	(76)
5.3.5 判定表与判定树	(77)
5.4 Jackson 程序设计方法	(78)
5.4.1 Jackson 图	(78)
5.4.2 Jackson 方法	(79)
小结	(82)
习题	(83)
第 6 章 程序设计语言与编码	(84)
6.1 程序设计语言概述	(84)
6.1.1 程序设计语言的分类	(84)
6.1.2 程序设计语言的机制	(86)
6.1.3 程序设计语言的选择	(89)
6.2 程序设计风格	(90)
6.3 程序设计方法	(91)
6.4 程序设计工具和环境	(93)
小结	(95)
习题	(95)
第 7 章 软件测试	(97)
7.1 软件测试概述	(97)
7.1.1 软件测试的概念	(97)
7.1.2 软件测试的原则	(98)
7.1.3 软件测试流程	(99)
7.2 软件测试方法与技术	(100)
7.3 软件测试的过程	(101)
7.3.1 单元测试	(102)
7.3.2 集成测试	(104)
7.3.3 验收测试	(108)
7.4 测试用例设计	(109)
7.4.1 逻辑覆盖	(109)
7.4.2 黑盒测试用例设计	(113)
7.4.3 实用测试策略	(117)
7.5 调试技术和策略	(120)
7.5.1 调试技术	(121)
7.5.2 调试策略	(122)
7.6 软件测试工具	(123)

7.6.1 静态分析工具	(123)
7.6.2 动态分析程序	(124)
7.6.3 测试数据生成程序	(124)
7.6.4 文件比较程序	(125)
小结	(125)
习题	(126)
第8章 软件维护	(128)
8.1 软件维护的基本认识	(128)
8.1.1 软件维护的定义	(128)
8.1.2 维护分类及策略	(129)
8.1.3 软件维护的代价	(131)
8.2 软件可维护性	(132)
8.2.1 软件可维护性的度量	(132)
8.2.2 提高可维护性的方法	(134)
8.3 维护过程	(136)
8.3.1 维护过程的事件流	(136)
8.3.2 维护团队组织	(137)
8.3.3 维护报告	(138)
8.3.4 维护任务	(138)
8.3.5 保存维护记录	(140)
8.3.6 评价维护活动	(141)
小结	(141)
习题	(141)
第9章 面向对象方法学	(143)
9.1 面向对象方法学引论	(143)
9.1.1 传统方法学的不足与模型进化	(143)
9.1.2 面向对象方法学的技术要点	(145)
9.1.3 面向对象的基本概念	(148)
9.1.4 类及对象的主要特性	(149)
9.1.5 类及对象的基本图形符号	(151)
9.1.6 面向对象的软件开发过程	(153)
9.2 面向对象分析	(154)
9.2.1 标识类与对象及其相互联系	(154)
9.2.2 3种子模型与5个层次	(157)
9.2.3 建立对象模型	(159)
9.2.4 建立动态模型	(161)
9.2.5 建立功能模型	(162)
9.3 面向对象设计	(163)
9.3.1 面向对象设计概要	(164)

9.3.2 面向对象设计的准则	(165)
9.3.3 软件重用的设计	(166)
9.3.4 问题论域子系统的设计	(167)
9.3.5 人机交互子系统的设计	(167)
9.3.6 任务管理子系统的设计	(168)
9.3.7 数据管理子系统的设计	(168)
9.4 面向对象编程	(169)
9.4.1 面向对象程序设计语言	(169)
9.4.2 面向对象语言的技术特点	(170)
9.4.3 选择面向对象语言	(171)
9.5 面向对象测试	(172)
9.5.1 对象级的软件测试	(172)
9.5.2 系统级的软件测试	(173)
9.6 面向对象方法与统一建模语言	(174)
9.6.1 UML 建模基本要素	(174)
9.6.2 UML 建模图形的应用	(176)
9.6.3 UML 建模过程要点	(183)
小结	(186)
习题	(187)
第 10 章 软件度量	(188)
10.1 软件度量的基本概念	(188)
10.2 软件复杂性度量	(190)
10.2.1 软件复杂性及其度量标准	(191)
10.2.2 程序复杂性的度量	(191)
10.3 软件可靠性度量	(195)
10.3.1 软件可靠性的定义	(195)
10.3.2 软件可靠性估算	(196)
10.4 软件质量的度量	(198)
10.4.1 软件质量定义	(199)
10.4.2 软件质量要素	(200)
10.4.3 软件质量评审	(203)
小结	(204)
习题	(205)
第 11 章 软件工程管理	(206)
11.1 成本管理	(206)
11.1.1 成本估算模型	(206)
11.1.2 成本估算方法	(209)
11.2 风险管理	(210)
11.2.1 风险标识	(210)

11.2.2	风险评估	(211)
11.2.3	风险评价	(212)
11.2.4	风险驾驭	(213)
11.3	进度安排	(214)
11.3.1	甘特图法	(214)
11.3.2	工程网络图法	(215)
11.4	质量管理	(218)
11.4.1	软件质量	(218)
11.4.2	软件质量保证	(219)
11.5	人员管理	(221)
11.5.1	软件开发人员特点及分配规律	(221)
11.5.2	项目组的组织形式	(222)
11.6	软件配置管理	(224)
11.6.1	软件配置项及其标识	(224)
11.6.2	基线技术	(225)
11.6.3	变更控制	(226)
11.6.4	版本管理	(227)
11.6.5	配置审核与配置状态报告	(228)
小结		(229)
习题		(230)
第 12 章	软件工程标准化	(231)
12.1	软件工程标准化及其意义	(231)
12.2	软件工程标准的类型与层次	(232)
12.3	软件文档标准化	(236)
12.4	ISO 9000 质量标准简介	(240)
12.5	软件能力成熟度模型 CMM 概要	(242)
小结		(245)
习题		(246)

第 1 章 软件工程概述

人类已经进入了以计算机为基础的信息社会。在信息社会里,信息的获取、处理、交流和决策需要大量高质量的计算机软件。然而,一直以来计算机软件的发展不尽如人意,成本逐年上升,质量没有可靠的保证。其原因一方面来自软件本身,因为软件生产是一个智力密集型劳动,大部分都是手工劳动,软件的功能越强、通用性越好、可靠性越大,则软件开发的复杂性就越大,人们对软件开发就越显得力不从心;另一方面原因来自落后的开发技术和工具。

软件发展的滞后制约了计算机科学与技术的发展。为了扭转这一不利局面,自 20 世纪 60 年代以来,人们十分重视软件开发理论、技术和方法的研究,并在这些领域取得了重要的成果,形成了一门新的学科——软件工程。

1.1 软 件

1.1.1 软件与程序

广义的软件是相对于有形物理实体而言的,人们一般把技术条件、管理法规和人员素质等无形因素统称为软件。而计算机软件是与计算机硬件相对应的,是计算机系统操作和运行有关的程序、数据及其相关文档的完整集合。其中,程序是按事先设计的功能和性能要求执行的指令序列;数据是使程序能正常操纵信息的数据结构;文档是与程序开发、维护和使用有关的图文资料。

在计算机系统发展早期(20 世纪 60 年代中期以前),软件只是为具体应用而专门编写的,这时的软件通常是规模较小的程序,程序的编写者和使用者往往是同一个人(或同一小组),软件设计只是人们头脑中一个隐含的过程,提交的软件除了程序清单,就没有更多的其他文档资料了,这样人们就形成了“软件即程序”的错误概念。随着程序规模扩大,仅包含程序的软件是很难维护的。

1.1.2 软件的分·类

自 20 世纪 40 年代以来,人类开发了大量的软件,它们被广泛应用于科学研究、教育、工农业生产、国防和家庭。特别是近年来,随着计算机应用范围的不断扩大,计算机软件

的需求量日益增长,其功能越来越丰富,品种越来越多。不同的分类角度可以反映计算机软件的不同特征,下面将从多个角度对计算机软件进行分类,以了解计算机软件的各种特征。

(1) 按功能计算机软件可分为系统软件、支撑软件和应用软件。

系统软件是计算机管理自身资源(如 CPU、内存空间、外存空间和外部设备等),提高计算机的使用效率,为计算机用户提供各种服务支持的基础软件。系统软件包括操作系统、数据库管理系统、系统检查与诊断软件、网络通信软件等。

支撑软件是协助用户开发软件的工具性软件、文本编辑软件等,如 PSL/PSA(问题描述语言、问题描述分析器)、图形软件包、预编译程序、静态分析程序等。

应用软件是在特定领域为特定目的服务的一类软件,如数值计算、计算机辅助设计与辅助制造、人工智能、计算机辅助教学和管理信息系统等。

(2) 按规模计算机软件可分为微型(Micro)软件、小型(Small)软件、中型(Middle)软件、大型(Large)软件、特大型(Very-large)软件和超大型(Ultra-large)软件。

小型、微型软件大多为个人软件,由个人(或小组)开发和使用,如工程师们开发的用于解数值问题的科学计算程序、用于数据处理的小型应用程序等。中型软件包括汇编程序、编译程序、小型管理信息系统和用于过程控制的应用程序。

大型软件包括大型编译程序、小型分时系统、数据库软件包、实时控制系统和某些图形软件。特大型软件常见于实时处理、远程通信和多任务处理等应用领域,如大型操作系统和数据库管理系统,军事上的指挥和控制系统等。超大型软件一般由多个特大型软件的子系统组成,常含有实时处理、远程通信、多任务处理和分布式处理等软件,如空中交通管制系统、洲际导弹防御系统、军事指挥和控制系统等。

(3) 按工作方式计算机软件可分为实时处理软件、分时软件、交互式软件和批处理软件等。

实时处理软件主要用于监视、分析和控制现实世界发生的事件,并以足够快的速度对输入信息进行处理并在规定的时间内做出反应;分时软件主要用于协调控制多个联机用户同时使用计算机;交互式软件是一种能实现人机通信的软件,主要用于人机界面的设计。

(4) 按服务对象计算机软件可分为项目软件、产品软件。

项目软件是一种受特定客户委托由一个或多个软件开发机构在合同的约束下开发出来的软件。产品软件是一类提供给市场的商品软件。

(5) 还可以按其他方式对计算机软件进行分类,如按使用频度分类,按软件失效的影响分类等。

1.1.3 软件的特征

软件是计算机系统逻辑部件而不是物理部件,因而软件在开发、生产、维护和使用等方面与硬件相比存在很大区别。

(1) 软件开发成本很高,但却可以低成本地复制。软件开发是一种高强度的脑力劳动,

更依赖于开发人员的业务素质和组织管理。硬件开发成功后,批量生产需要建设生产线,投入大量的人力、物力和资金,生产过程中需要进行产品的质量控制,对每件产品进行严格的检验,而软件开发成功后,只需要对原版本进行复制即可。

(2) 软件在使用过程中的维护工作远比硬件复杂。硬件产品经过检验、试用后一般不会出现什么问题,大多维护工作只对零部件的修复或更换,而软件在使用过程中可能暴露测试中没有发现的新的错误,同时用户在使用软件一段时间后可能会提出一些新的要求,要求软件适应新的运行环境或增加新的功能,因而其维护工作较复杂且难度较大。

(3) 软件的使用不存在老化的问题。软件是逻辑的,而不是物理的产品,一个久经考验的优质软件,只要其运行环境不发生改变,且没有新的功能需求,就可以长期使用下去,无须修理或更换零部件。

(4) 相当多的软件开发和使用还涉及一些社会因素,如社会制度、法制法规、伦理道德和开发者与使用者的心理因素等。

1.1.4 软件开发技术的发展

从 20 世纪 40 年代第一台计算机问世以来,软件开发技术的发展大致经历了以下 3 个阶段。

(1) 程序设计阶段。这个阶段大约发生在 20 世纪 50 年代到 60 年代的十余年间,是计算机软件开发初级阶段。这个阶段软件开发主要依赖于程序员个人的“技巧”,软件开发不规范,生产效率低,开发过程很难管理,开发成本难以估算和控制。

在这个阶段中,计算机软件多是用户为了解决自己的实际问题而开发的,多数软件是用户自己设计、自己使用、自己维护。软件开发规模小,文档不完整,软件开发的主要工作只是编写程序,其交付的软件产品也主要是程序。

(2) 程序系统阶段。这个阶段大约发生在 20 世纪 60 年代到 70 年代。这一时期软件开发的主要工作仍是编写程序,但已经开始注重开发方法和程序的测试工作,提交的软件产品除了程序外还有相应的使用说明书。

软件开发组织不再是程序员个人,出现了开发小组。由于每个软件开发人员都有自己独特的个性,加之软件项目管理不够完善,软件的可理解性和可维护性问题十分突出,“软件危机”问题开始引起人们的注意。

(3) 软件工程阶段。这个阶段大约发生在 20 世纪 70 年代末。随着“软件危机”问题日益加剧,人们意识到软件开发是一个系统工程,必须用工程的原则和方法来开发、维护计算机软件。

在软件工程阶段,开发工作遵循工程化开发方法,按照软件生命周期分阶段开发软件,开发工具和开发环境起着重要作用,提交的软件产品除了程序以外,还有规范严格的文档,软件开发目标面向市场、面向用户,开发组织一般为大中型软件开发机构,软件质量决定于软件开发过程中的管理水平。