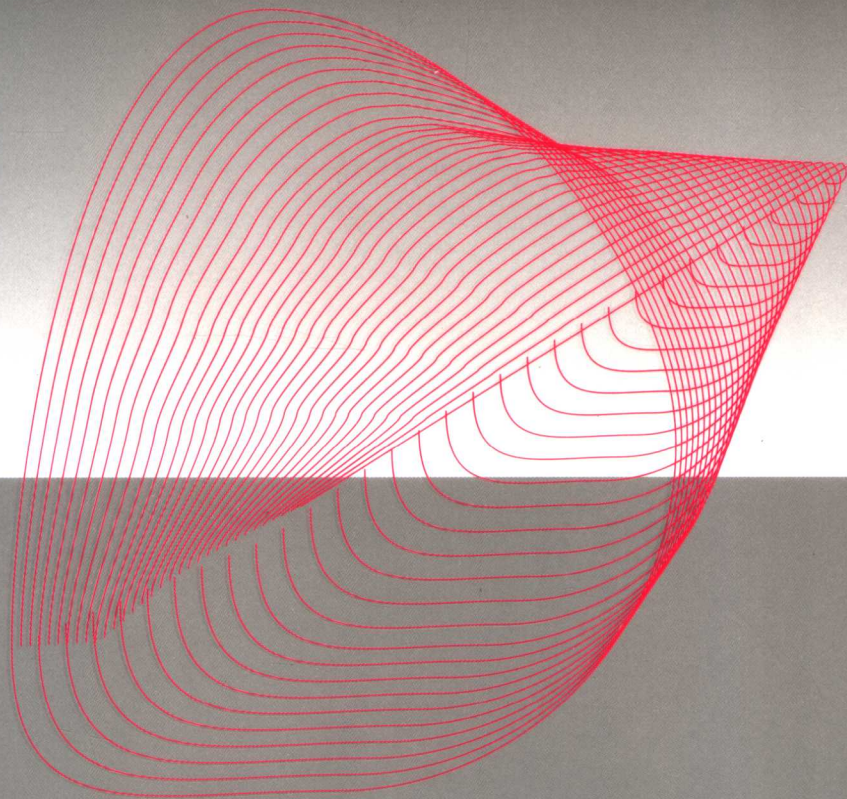


21

世纪高等学校计算机教育实用规划教材

# 实用软件工程

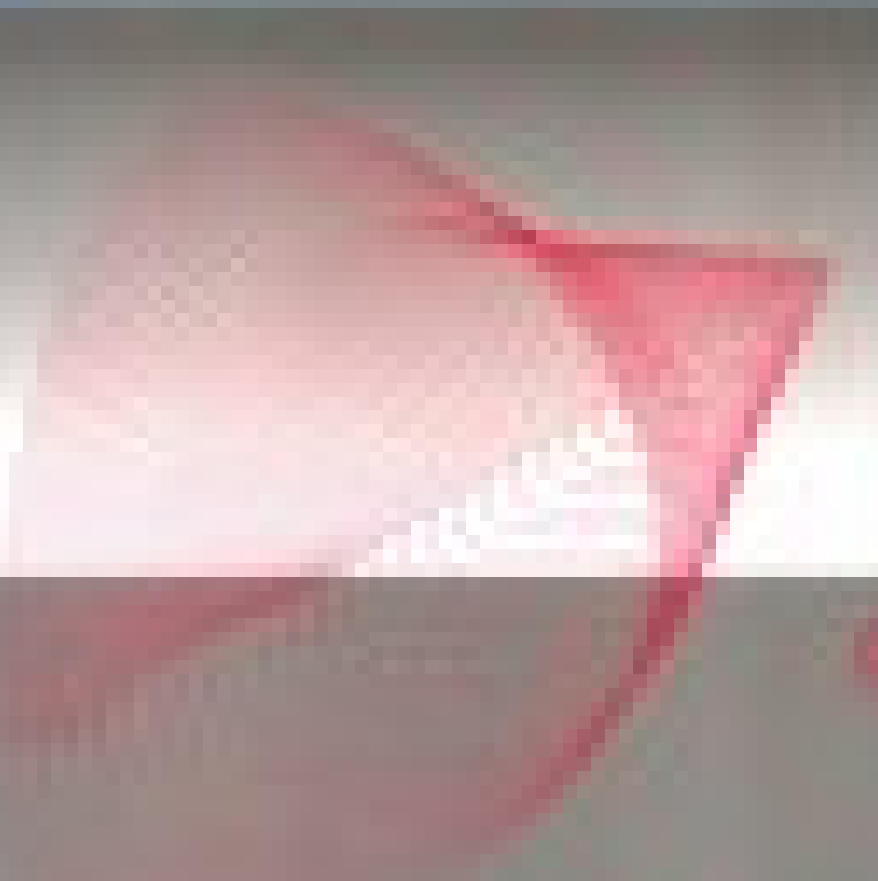
陆惠恩  
编著



清华大学出版社



# 应用软件工程



21 世纪高等学校计算机教育实用规划教材

# 实用软件工程

陆惠恩 编著

清华大学出版社  
北京

## 内 容 简 介

本书着重从实用角度讲述软件工程的基本概念、原理和方法,系统地介绍目前较成熟的、广泛使用的软件工程技术。

本书内容包括软件工程概述,软件计划(软件定义、可行性分析、需求分析),结构化设计(概要设计、详细设计、界面设计和数据代码设计等),软件编码与软件测试(程序设计、软件测试、调试、验证与确认),软件维护,面向对象方法学与统一建模语言 UML 的应用,软件工程管理技术,软件工具与软件开发环境,软件开发实例(招聘考试成绩管理系统)。每章都有小结供读者复习总结,精心挑选了习题供读者选用,选择题和填空题有答案。

本书可作为应用型本科计算机相关专业的教材,也可供从事计算机软件开发及应用的广大科技人员参考。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

实用软件工程 / 陆惠恩编著. —北京:清华大学出版社, 2006.5

(21 世纪高等学校计算机教育实用规划教材)

ISBN 7-302-12559-7

I. 实… II. 陆… III. 软件工程 - 高等学校 - 教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2006) 第 010818 号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦  
<http://www.tup.com.cn> 邮 编: 100084  
社总机: 010-62770175 客户服务: 010-62776969

责任编辑: 魏江江

印刷者: 北京市清华园胶印厂

装订者: 三河市李旗庄少明装订厂

发行者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 15 字数: 367 千字

版 次: 2006 年 5 月第 1 版 2006 年 5 月第 1 次印刷

书 号: ISBN 7-302-12559-7/TP · 8034

印 数: 1 ~ 4000

定 价: 21.00 元

# 出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善, 社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要, 科学运用市场调节机制, 合理调整和配置教育资源, 在改革和改造传统学科专业的基础上, 加强工程型和应用型学科专业建设, 积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业, 积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度, 从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时, 不断更新其教学内容、改革课程体系, 使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用, 工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展, 急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前, 工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践, 如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材), 重理论、轻实践, 不能满足按新的教学计划、课程设置的需要; 一些课程的教材可供选择的品种太少; 一些基础课的教材虽然品种较多, 但低水平重复严重; 有些教材内容庞杂, 书越编越厚; 专业课教材、教学辅助教材及教学参考书短缺, 等等, 都不利于学生能力的提高和素质的培养。为此, 在教育部相关教学指导委员会专家的指导和建议下, 清华大学出版社组织出版本系列教材, 以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业, 强调计算机在各专业中的应用。教材内容坚持基本理论适度, 反映基本理论和原理的综合应用, 强调实践和应用环节。

(2) 反映教学需要, 促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要, 正确把握教学内容和课程体系的改革方向, 在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养, 为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略, 突出重点, 保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上; 特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版, 逐步形成精品教材; 提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本, 合理配套。基础课和专业基础课教材要配套, 同一门课程可以有

多本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人: 丁岭 [dingl@tup.tsinghua.edu.cn](mailto:dingl@tup.tsinghua.edu.cn)

# 前 言

---

软件工程是指导计算机软件开发和维护的学科。软件工程采用工程的概念、原理、技术和方法，把良好的技术方法和正确的管理方法结合起来开发软件。软件工程学已成为计算机科学与技术专业的一门重要学科。

依据培养应用型人才的要求，编者在多年从事软件工程学教学和软件开发实践的基础上，参阅大量国内外有关软件工程的教材和资料，编写而成。本书着重从实用角度讲述软件工程的基本概念、原理和方法，系统地介绍目前较成熟的、广泛使用的软件工程技术。

通过本课程理论教学与实践环节的结合，读者能基本掌握结构化方法和面向对象方法等软件开发技术；学习如何系统地、规范地开发和维护软件；合理地安排软件开发和维护的过程；培养和提高软件开发和维护的能力；规范地书写软件工程的文档资料；提高软件开发过程的效率和质量。

本书的特点如下：

(1) 语言流畅、深入浅出、详略适当，可读性好、应用性强、易于理解。

(2) 引入软件工程较新的技术。如，软件过程，人机界面设计，面向对象技术，统一建模语言 UML，估算软件开发工作量的 COCOMO2 模型，统一软件开发过程 RUP，软件重用技术等。

(3) 每章列出主要内容、重点和小结，配有经过精选的适量例题和习题，附录中有选择题和填空题的答案，便于读者对内容的学习和理解。

(4) 书中介绍了软件工程各阶段文档书写的规范，使读者在开发软件时有参考依据。

(5) 针对软件开发的实际需要，介绍了数据代码设计的原则和方法。

(6) 介绍为读者容易理解的软件开发实例——招聘考试成绩管理系统，使读者对软件开发的全过程有感性认识，并附有实习思考题。

实用软件工程在程序设计语言、数据库原理、数据结构等专业课之后，毕业实习、毕业设计之前开设。

课程的教学重点如下：

(1) 软件工程的结构化方法（结构化分析、结构化设计和结构化程序设计）；

(2) 面向对象方法与 UML；

(3) 软件测试；

(4) 软件质量保证。

实用软件工程课程，建议理论学习为 45~50 学时，并适当安排实践环节。通过软件开发的实际训练来培养和提高学生开发、维护软件的能力。

软件工程实践环节可要求学生完成一个难度适中的软件设计课题。时间安排上，可集中 2~4 周进行课程设计，也可在理论教学的适当阶段同步安排实践环节，分阶段逐步完成

课题。

本书可作为应用型本科计算机相关专业的教材，也可供从事计算机软件开发及应用的广大科技人员参考。

IV

书中难免存在错误和不足之处，敬请读者批评指正，编者不胜感激，编者的电子邮件地址为：[luhuien@sit.edu.cn](mailto:luhuien@sit.edu.cn)。

编 者

2006年4月



## 读者意见反馈

亲爱的读者：

感谢您一直以来对清华大学出版社教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 705（100084） 魏江江收

电话：010-62770175-4604

电子邮件：weijj@tup.tsinghua.edu.cn

教材名称：实用软件工程

ISBN：7-302-12559-7/TP·8034

个人资料：

姓名：\_\_\_\_\_ 年龄：\_\_\_\_\_ 所在院校/专业：\_\_\_\_\_

职称（或职务）：\_\_\_\_\_ 通信地址：\_\_\_\_\_

联系电话：\_\_\_\_\_ 电子邮箱：\_\_\_\_\_

您使用本书是作为：指定教材 选用教材 辅导教材 自学教材

您对本书封面设计的满意度：

很满意 满意 一般 不满意 改进建议：\_\_\_\_\_

您对本书印刷质量的满意度：

很满意 满意 一般 不满意 改进意见：\_\_\_\_\_

您对本书的总体满意度：

从语言质量角度看 很满意 满意 一般 不满意

从科技含量角度看 很满意 满意 一般 不满意

本书最令您满意的是：

指导明确 内容充实 讲解详尽 实例丰富

您认为本书在哪些地方应进行修改？（可附页）

\_\_\_\_\_

您近期有何教材出版计划？

\_\_\_\_\_

\_\_\_\_\_

## 电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材有配套的教学素材，有需要的教师可以与我们的联系，我们将向使用本教材进行教学的教师免费赠送电子教案，希望有助于教学活动的开展。相关信息请拨打电话 010-62770175-4604 或发送电子邮件至 weijj@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（<http://www.tup.com.cn>）上查询。

# 目 录

---

<b>第 1 章 概述</b> .....	1
1.1 软件工程的产生 .....	1
1.1.1 软件工程发展史 .....	1
1.1.2 软件危机 .....	2
1.2 软件工程学 .....	4
1.2.1 什么是软件 .....	4
1.2.2 什么是软件工程 .....	4
1.2.3 软件工程学的内容 .....	4
1.2.4 软件过程 .....	7
1.2.5 软件工程基本原理 .....	7
1.3 软件生命周期 .....	7
1.3.1 软件生命周期定义 .....	7
1.3.2 软件生命周期划分阶段的原则 .....	8
1.3.3 软件生命周期各阶段的任务 .....	8
1.4 软件开发模型 .....	9
1.4.1 瀑布模型 .....	9
1.4.2 快速原型模型 .....	11
1.4.3 增量模型 .....	12
1.4.4 喷泉模型 .....	13
1.4.5 螺旋模型 .....	14
1.4.6 统一过程 .....	15
小结 .....	16
习题 1 .....	17
<b>第 2 章 软件计划</b> .....	19
2.1 软件问题定义及可行性研究 .....	19
2.1.1 软件问题定义 .....	19
2.1.2 可行性研究 .....	20
2.1.3 制定项目开发计划 .....	21
2.2 需求分析的任务 .....	22
2.2.1 确定目标系统的具体要求 .....	22

2.2.2	建立目标系统的逻辑模型	24
2.2.3	软件需求规格说明	24
2.2.4	修正系统开发计划	25
2.2.5	制定初步的系统测试计划	25
2.2.6	编写初步的用户手册	25
2.2.7	编写数据要求说明书	26
2.3	需求分析步骤	26
2.3.1	进行调查研究	27
2.3.2	分析和描述系统的逻辑模型	27
2.3.3	复审	28
2.4	实体-关系图	28
2.4.1	实体	28
2.4.2	属性	29
2.4.3	关系	29
2.5	数据流图	30
2.5.1	数据流图四种基本符号	30
2.5.2	数据流图几种附加符号	31
2.5.3	画数据流图的步骤	32
2.5.4	几点注意事项	32
2.6	状态转换图	35
2.7	数据字典	36
2.7.1	数据字典的内容	36
2.7.2	数据字典使用的符号	37
2.7.3	数据字典与图形工具	38
2.8	需求分析的其他图形工具	38
2.8.1	层次图	39
2.8.2	Warnier 图	39
2.8.3	IPO 图	40
2.9	软件计划阶段文档编写步骤	40
小结		41
习题 2		41
<b>第 3 章</b>	<b>结构化设计</b>	<b>43</b>
3.1	软件设计步骤	43
3.1.1	概要设计步骤	44
3.1.2	详细设计的基本任务	45
3.2	软件结构设计	46
3.2.1	软件结构设计的基本原理	46
3.2.2	模块化	48

3.2.3	模块独立性	49
3.2.4	模块设计启发规则	53
3.3	软件结构设计的图形工具	54
3.3.1	层次图(或 HIPO 图)	54
3.3.2	结构图	54
3.4	面向数据流的设计方法	56
3.4.1	变换型	57
3.4.2	事务型	57
3.5	过程设计工具	58
3.5.1	流程图	59
3.5.2	盒图(N-S)	63
3.5.3	PAD 图	65
3.5.4	判定表	66
3.5.5	判定树	66
3.5.6	过程设计语言	67
3.6	系统人机界面设计	67
3.6.1	人机界面设计问题	67
3.6.2	人机界面设计过程	69
3.6.3	评估界面设计标准	69
3.6.4	界面设计指南	69
3.7	数据代码设计	70
3.7.1	代码设计目的	70
3.7.2	代码设计原则	71
3.7.3	代码种类	72
3.7.4	代码设计方法	73
3.8	面向数据结构的设计方法	74
3.9	软件设计文档	77
3.9.1	概要设计说明书	77
3.9.2	数据库设计说明书	78
3.9.3	详细设计说明书	79
3.9.4	操作手册编写提示	80
小结		81
习题 3		82
<b>第 4 章</b>	<b>软件编码和软件测试</b>	<b>84</b>
4.1	结构化程序设计	84
4.1.1	程序设计语言的选择	84
4.1.2	程序设计风格	86
4.2	软件测试目标	88

4.3	软件测试方法	89
4.3.1	静态分析与动态测试	89
4.3.2	黑盒法与白盒法	89
4.4	软件测试步骤	90
4.4.1	模块测试	90
4.4.2	集成测试	90
4.4.3	程序审查会和人工运行	91
4.4.4	确认测试	91
4.4.5	平行运行	92
4.5	设计软件测试方案	93
4.5.1	等价类划分法	93
4.5.2	边界值分析法	93
4.5.3	错误推测法	94
4.5.4	逻辑覆盖法	94
4.5.5	程序环行复杂程度的度量	97
4.5.6	因果图法	99
4.6	软件测试原则和策略	102
4.6.1	软件测试原则	102
4.6.2	实用测试策略	102
4.7	软件调试、验证与确认	103
4.7.1	软件调试	103
4.7.2	软件验证	104
4.7.3	软件确认	104
4.8	软件测试文档	105
	小结	105
	习题 4	106
<b>第 5 章 软件维护</b>		<b>110</b>
5.1	软件维护的种类、过程和副作用	110
5.1.1	软件维护的种类	110
5.1.2	软件维护的特点	111
5.1.3	软件维护的过程	112
5.1.4	软件维护的副作用	115
5.2	软件的可维护性	116
5.2.1	决定可维护性的因素	116
5.2.2	可维护性的度量	116
5.2.3	如何提高软件的可维护性	117
	小结	118
	习题 5	118

<b>第 6 章 面向对象方法学与 UML</b> .....	120
6.1 面向对象方法概述 .....	120
6.1.1 面向对象方法学的要素和优点 .....	121
6.1.2 面向对象方法的概念 .....	122
6.2 UML 概述 .....	125
6.2.1 UML 的发展 .....	125
6.2.2 UML 的内容 .....	125
6.3 UML 图 .....	127
6.3.1 用例图 .....	127
6.3.2 类图和包 .....	129
6.3.3 对象图 .....	133
6.3.4 状态图 .....	134
6.3.5 顺序图 .....	135
6.3.6 活动图 .....	135
6.3.7 协作图 .....	136
6.3.8 构件图 .....	137
6.3.9 部署图 .....	138
小结 .....	139
习题 6 .....	139
<b>第 7 章 面向对象软件设计及 UML 应用</b> .....	141
7.1 面向对象分析 .....	141
7.1.1 面向对象分析过程 .....	141
7.1.2 面向对象分析原则 .....	142
7.2 建立对象模型 .....	143
7.2.1 建立对象模型的基本方法 .....	143
7.2.2 确定对象和类 .....	144
7.2.3 确定类的相互关系 .....	145
7.2.4 划分主题 .....	149
7.3 建立动态模型 .....	150
7.3.1 编写脚本 .....	151
7.3.2 设计用户界面 .....	152
7.3.3 画 UML 顺序图或活动图 .....	152
7.3.4 画状态转换图 .....	152
7.4 建立功能模型 .....	153
7.5 面向对象设计 .....	155
7.5.1 系统设计 .....	155
7.5.2 对象设计 .....	158

7.5.3	面向对象设计的准则	159
7.5.4	面向对象设计的启发规则	160
7.6	UML 的应用	160
7.6.1	UML 模型	161
7.6.2	UML 视图	162
7.6.3	UML 使用准则	164
7.6.4	UML 的扩展机制	164
7.6.5	UML 的应用领域	165
7.7	面向对象系统的实现	166
7.7.1	选择程序设计语言	166
7.7.2	面向对象程序设计	167
7.7.3	面向对象的测试	168
7.8	RUP 统一过程	168
7.8.1	统一过程的开发模式	168
7.8.2	统一过程的特点	169
7.8.3	统一过程的要素	170
7.9	软件重用	170
7.9.1	可重用的软件成分	171
7.9.2	软件重用过程模型	172
7.9.3	开发可重用的软件构件	173
7.9.4	分类和检索软件构件	173
	小结	174
	习题 7	175
<b>第 8 章</b>	<b>软件工具与软件开发环境</b>	<b>177</b>
8.1	CASE 技术	177
8.1.1	CASE 的基本组成部分	177
8.1.2	CASE 的软件平台	178
8.1.3	CASE 的硬件平台	178
8.2	软件工具	178
8.2.1	软件工具的功能	179
8.2.2	软件工具的性能	179
8.2.3	软件工具的分类	180
8.3	软件开发环境	181
8.3.1	软件开发环境的定义	181
8.3.2	软件开发环境的分类	181
8.3.3	软件开发环境的构成和特性	181
8.3.4	集成化环境	182
8.3.5	集成化的层次	182

小结 .....	183
习题 8 .....	183
<b>第 9 章 软件工程管理 .....</b>	<b>184</b>
9.1 软件工程管理概述 .....	184
9.2 软件规模估算 .....	185
9.2.1 软件开发成本估算方法 .....	185
9.2.2 代码行技术 .....	186
9.2.3 功能点技术 .....	187
9.2.4 COCOMO2 模型 .....	188
9.3 进度计划 .....	190
9.3.1 Gantt 图 .....	190
9.3.2 工程网络技术 .....	191
9.4 人员组织 .....	194
9.5 软件配置管理 .....	196
9.6 软件质量保证 .....	199
9.6.1 软件质量的特性 .....	199
9.6.2 软件质量保证措施 .....	200
9.7 软件工程标准与软件文档 .....	201
9.7.1 软件工程标准 .....	201
9.7.2 软件文档的编写 .....	203
小结 .....	204
习题 9 .....	204
<b>第 10 章 实例——招聘考试成绩管理系统 .....</b>	<b>206</b>
10.1 问题定义 .....	206
10.2 可行性研究 .....	206
10.2.1 技术可行性 .....	206
10.2.2 经济可行性 .....	207
10.3 需求分析 .....	207
10.3.1 考生情况分析 .....	207
10.3.2 成绩输入 .....	207
10.3.3 录用 .....	207
10.3.4 输出需求 .....	208
10.3.5 数据流图和数据字典 .....	208
10.3.6 IPO 图 .....	209
10.4 概要设计 .....	209
10.4.1 数据库结构设计 .....	209
10.4.2 系统结构设计 .....	210



随着计算机系统的发展, 计算机的应用日益广泛, 计算机软件的开发、维护工作显得越来越重要。如何才能开发用户满意的软件; 如何以较低的成本开发出高质量的软件; 怎样使所开发的软件在运行过程中容易维护, 以延长软件的使用期限; 如何提高软件开发、维护过程中的自动化程度、提高软件开发效率; 软件工程如何管理等。这些就是软件工程研究的问题。软件工程是指导计算机软件开发和维护的学科。软件工程的目的是在规定的时间内、规定的开发费用内, 开发出满足用户需要的、质量合格的软件产品。

本章介绍软件工程的发展史, 软件危机是如何形成和消除的, 软件工程的基本概念、内容及基本原理等。

本章重点: 软件工程、软件生命周期。

## 1.1 软件工程的产生

计算机系统的发展、软件应用的日益广泛和软件危机的困扰促使了软件工程的产生和发展。

### 1.1.1 软件工程发展史

自从 20 世纪 40 年代电子计算机发明以来, 计算机软件随着计算机硬件的发展而逐步发展, 软件和硬件一起构成计算机系统。一开始只有程序的概念, 后来才出现软件的概念。

当软件需求量大大增加后, 人们把软件视为产品, 确定了软件生产的各个阶段必须完成的有关计算机程序的功能、设计和使用的文字或图形资料, 这些资料称为“文档”。软件是指计算机程序及其有关的数据和文档。

随着计算机系统的发展, 软件的生产大体经历了程序设计、软件、软件工程、第四代技术等阶段。在此过程中, 软件危机产生并越来越严重, 因而逐步形成了研究如何消除软件危机, 如何合理地开发和维护软件的学科——软件工程学。

#### 1. 程序设计阶段

20 世纪 40 年代中期到 20 世纪 60 年代中期, 电子计算机价格昂贵、运算速度低、存储量小。计算机程序描述计算任务的处理对象和处理规则。早期的程序规模小, 程序往往是个人设计、自己使用。程序设计通常要注意如何节省存储单元、提高运算速度。除了程序清单之外, 没有其他任何文档资料。

#### 2. “软件=程序+文档”阶段

20 世纪 60 年代中期到 20 世纪 70 年代中期, 集成电路计算机的运算速度和内存容量