

普通高等教育“十一五”国家级规划教材
高等院校计算机科学与技术系列教材

软件工程

(第2版)

李代平 编著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

本书是论述软件工程学及应用的专著。从理论上介绍软件工程的基本理论，可行性研究方法。详细介绍了结构化方法和面向对象方法的基本理论，系统需求分析与设计的基本概念、用户界面设计、开发过程、关键问题以及在各领域的应用。同时介绍了形式化方法，软件质量，软件实现，软件容错技术，软件测试，软件维护，软件项目管理与计划。

本书内容详实，立论严谨，实例丰富，图文并茂，适合作为高等学校相关专业的教材及工程技术人员的参考用书。

图书在版编目(CIP)数据

软件工程 / 李代平编著. —2 版. —北京: 冶金工业出版社, 2006.11
ISBN 7-5024-4125-5

I. 软... II. 李... III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2006) 第 128687 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2006 年 11 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 31.5 印张; 729 千字; 492 页

45.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

随着科学技术的进步，软件的理论及开发方法不断涌现。软件工程是指导计算机软件开发的工程科学技术。软件工程的理论、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。

用软件工程进行软件设计与开发的先进性是众所周知的，它在计算机科学技术领域占据了无可争议的主流地位。作为软件技术人员，接受软件工程的理论并不难，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。鉴于此编写了本书，其目的就是向读者提供一本关于软件系统分析、设计和实施的教科书，以使更多同行受益。

二、本书结构

本书由五部分组成：

第一部分基础理论，包括：第1章，第2章，第3章。

第二部分结构化方法，包括：第4章，第5章，第6章。

第三部分面向对象方法，包括：第7章，第8章，第9章，第10章，第11章，第12章，第13章，第14章，第15章。

第四部分质量与实现，包括：第16章，第17章，第18章，第19章。

第五部分工程管理，包括：第20章，第21章，第22章。

三、本书特点

本书对每章的概念都进行了严格的论述，每一个概念都有相应的例子解释，同时每章都配有习题，使读者巩固所学知识。

四、适用范围

软件工程是软件系统开发课程的教科书。讲授时间一般为44~54学时。本书适合开设有软件工程课程的大学高年级和低年级研究生作教材，也可作为工程技术人员的参考用书。

在选修本课程之前，读者应该具有计算机的基础知识，掌握数据结构和数据库技术。同时具有可视化类语言的编程经验，会有助于深入理解系统开发过程。

五、编写方法

本书是作者根据近十年来对软件工程学、面向对象方法等的教学与研究，以及作者领导或参与的二十项软件项目开发的实际应用经验，并结合软件开发新技术编写而成。根据过去的教学经验，读者学习一门新技术，教材是非常重要的。因此，在编写之前，在各方

面进行了充分的准备。

六、如何使用本书

根据读者的实际情况，教师在教授本书时，一般在 48~60 学时。可以按照自己的风格和喜好删除章节，也可以根据教学目标灵活调整章节顺序。另外，前面带*为选学内容。

第 1 章：绪论（1 学时）

第 2 章：基本理论（2 学时）

第 3 章：可行性研究（2 学时）

第 4 章：软件需求分析（4 学时）

第 5 章：总体设计（4 学时）

第 6 章：软件详细设计（2 学时）

第 7 章：面向对象方法概论（3 学时）

第 8 章：模型（2 学时）

第 9 章：发现对象、建立对象类（4 学时）

第 10 章：定义属性与服务（3 学时）

第 11 章：定义结构与连接（3 学时）

*第 12 章：面向对象设计原则（2 学时）

第 13 章：控制驱动部分的设计（4 学时）

第 14 章：对象设计（4 学时）

*第 15 章：数据库及其接口设计（4 学时）

第 16 章：用户界面设计（2 学时）

*第 17 章：形式化方法（4 学时）

第 18 章：软件质量（2 学时）

第 19 章：软件实现（2 学时）

第 20 章：软件测试（2 学时）

第 21 章：软件维护（2 学时）

第 22 章：软件项目管理与计划（2 学时）

本书得到了教育部高等学校计算机科学与技术教学指导分委会委员，中国计算机学会理事，中国软件行业协会理事，东南大学计算机科学与工程系教授、博士生导师徐宝文的监审，在此表示衷心的感谢！

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

电子邮箱：service@cnbook.net

网址：www.cnbook.net

本书电子教案和习题参考答案可从该网站下载中心下载。

编者

2006 年 9 月

于广州小谷围岛

目 录

第1章 绪论	1	2.4.2 Jackson 方法	30
1.1 软件概述.....	1	2.4.3 维也纳开发方法	30
1.1.1 什么是计算机软件	1	2.4.4 面向对象的开发方法	31
1.1.2 软件的特点	2	2.5 软件工具与开发	31
1.1.3 软件分类	3	2.5.1 软件工具箱	31
1.1.4 软件的发展	5	2.5.2 软件开发环境	32
1.1.5 软件危机	5	2.5.3 计算机辅助软件工程	32
1.2 软件工程	7	小结	32
1.2.1 软件工程与方法学	8	综合练习二	33
1.2.2 软件工程的基本原理	9	一、填空题	33
1.2.3 软件工程的目标	11	二、选择题	33
1.2.4 软件工程的内容	12	三、简答题	33
1.2.5 软件工程原则	13	第3章 可行性研究	34
1.2.6 软件工程面临的问题	14	3.1 可行性研究任务与步骤	34
小结	15	3.1.1 研究任务	35
综合练习一	15	3.1.2 研究步骤	37
一、填空题	15	3.2 系统分析	39
二、选择题	15	3.2.1 系统分析员	39
三、简答题	16	3.2.2 面临的问题域	40
第2章 基本理论	17	3.2.3 通信技术	40
2.1 软件工程过程	17	3.3 分析原理	42
2.2 软件生命周期	17	3.3.1 信息域	42
2.2.1 软件分析时期	18	3.3.2 建立模型	43
2.2.2 软件设计时期	19	3.3.3 分解	44
2.2.3 编码与测试时期	20	3.4 系统模型与模拟	45
2.2.4 运行与维护时期	21	3.4.1 系统模型	45
2.3 软件生命周期模型	21	3.4.2 系统建模和模拟	46
2.3.1 软件生命周期模型的概念	21	3.5 成本-效益分析	47
2.3.2 瀑布模型	22	3.6 可行性研究的文档	49
2.3.3 原型模型	23	3.7 项目开发计划	50
2.3.4 增量模型	24	3.7.1 方案选择	50
2.3.5 螺旋模型	26	3.7.2 制定项目开发计划	50
2.3.6 喷泉模型	27	小结	51
2.3.7 基于知识的模型	28	综合练习三	51
2.3.8 变换模型	29	一、填空题	51
2.4 软件开发方法	29	二、选择题	51
2.4.1 结构化方法	29	三、简答题	52

第4章 软件需求分析	53	5.5.1 软件结构图.....	92
4.1 需求分析.....	53	5.5.2 模块的大小.....	94
4.1.1 需求分析的特点.....	53	5.5.3 扇出和扇入与深度和宽度.....	94
4.1.2 需求分析的原则.....	54	5.5.4 模块的耦合.....	95
4.1.3 需求分析的任务.....	54	5.5.5 模块的内聚.....	97
4.1.4 需求分析的方法.....	55	5.5.6 结构设计的一般准则.....	99
4.2 结构化分析.....	57	5.5.7 模块的作用域与控制域.....	101
4.2.1 自顶向下逐层分解.....	57	5.6 结构化设计.....	101
4.2.2 结构化分析步骤.....	58	5.6.1 数据流的类型.....	102
4.3 系统流程图.....	59	5.6.2 过程步骤.....	103
4.4 数据流图.....	60	5.6.3 变换分析设计.....	104
4.4.1 基本图形符号.....	61	5.6.4 事务分析设计.....	105
4.4.2 画数据流图.....	62	5.6.5 混合流设计.....	106
4.4.3 结构化分析方法的应用.....	65	5.6.6 结构化设计方法应用示例.....	108
4.5 数据字典.....	69	5.6.7 设计的后期处理.....	109
4.5.1 内容及格式.....	69	5.7 软件结构优化.....	109
4.5.2 数据字典的实现.....	71	5.7.1 软件结构设计优化准则.....	109
4.6 关系数据理论.....	71	5.7.2 软件结构的 HIPO 图.....	111
4.6.1 数据依赖.....	72	小结.....	111
4.6.2 关系模式的操作异常.....	72	综合练习五.....	112
4.6.3 范式.....	73	一、填空题.....	112
4.6.4 EAR 方法.....	77	二、选择题.....	112
小结.....	80	三、简答题.....	112
综合练习四.....	80	第6章 软件详细设计	113
一、填空题.....	80	6.1 细节设计的任务与方法.....	113
二、选择题.....	80	6.1.1 细节设计的基本任务.....	113
三、简答题.....	81	6.1.2 细节设计方法.....	114
第5章 总体设计	82	6.2 设计表示法.....	115
5.1 软件设计的重要性.....	82	6.2.1 结构化语言.....	115
5.2 设计过程.....	83	6.2.2 判定表.....	115
5.3 软件总体设计.....	84	6.2.3 判定树.....	116
5.4 设计基本原理.....	85	6.3 结构化程序设计.....	117
5.4.1 抽象.....	86	6.3.1 程序流程图.....	117
5.4.2 细化.....	86	6.3.2 三种基本控制结构.....	118
5.4.3 模块化.....	87	6.3.3 常用符号.....	119
5.4.4 软件体系结构.....	88	6.4 结构化定理.....	120
5.4.5 程序结构.....	89	6.4.1 程序函数.....	121
5.4.6 数据结构.....	90	6.4.2 基本定理.....	121
5.4.7 软件过程.....	91	6.4.3 常见错误.....	122
5.5 体系结构设计.....	92	6.4.4 过程设计语言.....	124

6.5 面向数据结构的设计.....	127	7.3.10 角色名.....	161
6.5.1 Jackson 图.....	127	7.3.11 排序.....	162
6.5.2 纲要逻辑.....	129	7.3.12 资格符.....	162
6.5.3 Jackson 方法.....	129	7.4 聚合.....	163
6.5.4 JSP 应用.....	130	7.4.1 聚合与关联.....	163
6.5.5 JSD 方法.....	132	7.4.2 聚合与概括.....	163
小结.....	136	7.4.3 递归聚合.....	164
综合练习六.....	137	7.4.4 操作的传播.....	165
一、填空题.....	137	7.4.5 物理聚合与分类聚合.....	165
二、选择题.....	137	7.4.6 物理聚合的语义扩展.....	166
三、简答题.....	137	7.4.7 分类聚合的语义扩展.....	166
第 7 章 面向对象方法概论.....	139	7.5 概括.....	166
7.1 面向对象简介.....	139	7.5.1 一般概念.....	166
7.1.1 什么是面向对象.....	139	7.5.2 概括的使用.....	167
7.1.2 面向对象方法的历史及现状.....	141	7.5.3 重写特征.....	167
7.2 面向对象的相关概念.....	141	7.5.4 抽象类和具体类.....	168
7.2.1 对象.....	141	7.5.5 概括与其他对象建模结构.....	168
7.2.2 类.....	143	7.6 构造分组.....	169
7.2.3 对象图.....	144	7.6.1 模块.....	169
7.2.4 属性.....	144	7.6.2 表.....	169
7.2.5 服务(操作或方法).....	145	小结.....	169
7.2.6 封装.....	145	综合练习七.....	169
7.2.7 继承.....	146	一、填空题.....	169
7.2.8 多重继承.....	149	二、选择题.....	170
7.2.9 消息.....	151	三、简答题.....	170
7.2.10 结构与连接.....	152	第 8 章 模型.....	171
7.2.11 多态性.....	153	8.1 统一建模语言.....	171
7.2.12 永久对象.....	156	8.1.1 模型的建立.....	172
7.2.13 主动对象.....	156	8.1.2 UML 的基本图标.....	172
7.2.14 对象类的表示方法.....	157	8.1.3 UML 的基本元素.....	181
7.3 链接与关联.....	157	8.1.4 UML 的语法规则.....	181
7.3.1 一般概念.....	157	8.1.5 UML 的词别.....	182
7.3.2 重数.....	158	8.2 对象模型.....	182
7.3.3 关联的重要性.....	159	8.2.1 表示方法.....	182
7.3.4 三元关联.....	159	8.2.2 表示结构.....	183
7.3.5 关联的候选关键字.....	159	8.2.3 例子.....	185
7.3.6 异或关联.....	160	8.3 动态模型.....	185
7.3.7 资格关联.....	160	8.3.1 事件和状态.....	186
7.3.8 链接属性.....	160	8.3.2 操作.....	190
7.3.9 用关联模型化为类.....	161	8.3.3 嵌套状态图.....	191

8.3.4 实践技巧	194	综合练习九	219
8.4 功能模型	195	一、填空题	219
8.4.1 数据流图	195	二、选择题	220
8.4.2 指定的操作	196	三、简答题	220
8.4.3 约束	197	第 10 章 定义属性与服务	221
8.4.4 功能模型的实例 (飞行模拟机装置)	197	10.1 对象的属性和服务	221
8.4.5 功能模型与对象模型和 动态模型的关系	200	10.2 表示法	222
小结	201	10.3 定义属性	222
综合练习八	201	10.3.1 策略与启发	222
一、填空题	201	10.3.2 审查与筛选	223
二、选择题	201	10.3.3 推迟到 OOD 考虑的问题	225
三、简答题	202	10.3.4 属性的命名和定位	226
第 9 章 发现对象、建立对象类	203	10.3.5 属性的详细说明	226
9.1 对象、主动对象以及它们的类	203	10.4 定义服务	226
9.2 表示法	204	10.4.1 对象的状态与状态转换图	226
9.3 研究问题域和用户需求	205	10.4.2 行为分类	228
9.3.1 研究用户需求, 明确系统责任	205	10.4.3 发现服务的策略与启发	229
9.3.2 研究问题域	206	10.4.4 审查与调整	229
9.3.3 确定系统边界	207	10.4.5 认识对象的主动行为	230
9.4 发现对象	207	10.4.6 服务的命名和定位	230
9.4.1 正确地运用抽象原则	207	10.4.7 服务的详细说明	230
9.4.2 策略与启发	208	10.5 建立类图的特征层	231
9.4.3 审查和筛选	210	10.6 电梯例子	231
9.4.4 识别主动对象	212	10.6.1 电梯系统的属性描述	232
9.5 对象的发现和标识	212	10.6.2 电梯系统的服务定义	233
9.5.1 动机	212	10.6.3 电梯系统的特征层	238
9.5.2 方法	213	小结	238
9.5.3 三视图模型(3VM)	213	综合练习十	238
9.5.4 语言信息分析	214	一、填空题	238
9.6 对象分类, 建立类图的对象层	216	二、选择题	239
9.6.1 异常情况的检查和调整	216	三、简答题	239
9.6.2 类的命名	217	第 11 章 定义结构与连接	240
9.6.3 建立类图的对象层	217	11.1 整体-部分结构	240
9.7 电梯例子	218	11.1.1 整体-部分结构及其用途	240
9.7.1 功能需求	218	11.1.2 表示法	242
9.7.2 发现对象	218	11.1.3 如何发现整体-部分结构	243
9.7.3 对象层表示	219	11.1.4 审查与筛选	244
小结	219	11.1.5 简化对象的定义	244
		11.1.6 支持软件复用	245
		11.1.7 整体-部分结构的进一步运用	246

11.1.8 调整对象层和属性层	247	12.1.1 类与类型	270
11.2 一般-特殊结构	247	12.1.2 类型一致性原则	270
11.2.1 一般-特殊结构及其用途	247	12.1.3 闭合行为原则	271
11.2.2 表示法	248	12.2 封装与共生性	272
11.2.3 如何发现一般-特殊结构	249	12.2.1 封装结构	272
11.2.4 审查与调整	250	12.2.2 共生性	273
11.2.5 多继承及多态性问题	250	12.2.3 面向对象系统中共生性的滥用	275
11.2.6 一般-特殊结构的简化	253	12.2.4 共生性的术语	276
11.2.7 调整对象层和特征层	253	12.3 领域、依附集和内聚	276
11.3 实例连接	253	12.3.1 对象类的领域	276
11.3.1 简单的实例连接	254	12.3.2 依附集	278
11.3.2 复杂的实例连接及其表示	255	12.3.3 类的内聚：一个类和它的特性	280
11.3.3 三元关联问题	257	12.4 继承与多态性的危险性	282
11.3.4 如何建立实例连接	257	12.4.1 继承的滥用	282
11.3.5 对象层、特征层的增补及 实例连接说明	259	12.4.2 错误的聚集	282
11.4 消息连接	259	12.4.3 倒置的层次结构	282
11.4.1 消息的定义	259	12.4.4 混淆类及其实例	283
11.4.2 顺序系统中的消息	259	12.4.5 误用	283
11.4.3 并发系统中的消息	260	12.4.6 多态性的危险性	284
11.4.4 消息对 OOA 的意义	263	12.5 状态空间和行为	285
11.4.5 OOA 对消息的表示—— 消息连接	263	12.5.1 一个类的状态空间和行为	285
11.5 如何建立消息连接	264	12.5.2 子类的状态空间	286
11.5.1 建立控制线程内部的消息连接	264	12.5.3 子类的行为	286
11.5.2 建立控制线程之间的消息连接	265	12.5.4 状态空间的一个约束条件： 类的不变式	287
11.5.3 对象分布问题及其消息的影响	265	12.5.5 前置条件和后置条件	288
11.6 消息的详细说明	266	12.5.6 类接口中支持的状态	288
11.7 电梯例子	267	12.5.7 类接口中支持的行为	289
11.7.1 一般-特殊关系	267	12.5.8 类接口中操作的聚合	290
11.7.2 整体-部分关系	267	小结	290
11.7.3 连接	267	综合练习十二	290
11.7.4 电梯控制系统的关系层	268	一、填空题	290
小结	268	二、选择题	291
综合练习十一	268	三、简答题	291
一、填空题	268	第 13 章 控制驱动部分的设计	292
二、选择题	268	13.1 什么是控制驱动部分	292
三、简答题	269	13.2 相关技术问题	292
第 12 章 面向对象设计原则	270	13.2.1 系统总体方案	292
12.1 类型一致性与闭合行为	270	13.2.2 软件体系结构	293
		13.2.3 分布式系统的体系结构风格	294

13.2.4 系统的并发性.....	297	14.8 对象的表示.....	324
13.3 如何设计控制驱动部分.....	300	14.9 物理打包.....	324
13.3.1 选择软件体系结构风格.....	300	14.9.1 信息隐藏.....	324
13.3.2 确定系统分布方案.....	301	14.9.2 实体的相关性.....	325
13.3.3 识别控制流.....	304	14.9.3 构造模块.....	326
13.3.4 用主动对象表示控制流.....	306	14.10 设计决策文档.....	326
13.3.5 把控制驱动部分看作一个主题.....	308	小结.....	327
小结.....	308	综合练习十四.....	327
综合练习十三.....	308	一、填空题.....	327
一、填空题.....	308	二、选择题.....	327
二、选择题.....	308	三、简答题.....	328
三、简答题.....	309	第15章 数据库及其接口设计.....	329
第14章 对象设计.....	310	15.1 数据管理系统及其选择.....	329
14.1 对象设计综述.....	310	15.2 技术整合.....	336
14.1.1 从分析和系统结构着手.....	310	15.3 数据接口.....	337
14.1.2 对象设计的步骤.....	311	15.4 对象存储方案和数据接口的 设计策略.....	338
14.1.3 对象模型工具.....	311	15.4.1 针对文件系统的设计.....	339
14.2 组合三种模型.....	311	15.4.2 针对RDBMS的设计.....	343
14.3 设计算法.....	312	15.4.3 使用OODBMS.....	352
14.3.1 选择算法.....	312	小结.....	352
14.3.2 选择数据结构.....	314	综合练习十五.....	353
14.3.3 定义内部类和操作.....	314	一、填空题.....	353
14.3.4 指定操作的职责.....	315	二、选择题.....	353
14.4 设计优化.....	315	三、简答题.....	353
14.4.1 添加冗余关联获取有效访问.....	316	第16章 用户界面设计.....	354
14.4.2 重新安排执行次序以获得效率.....	317	16.1 人的因素.....	354
14.4.3 保存导出属性避免重复计算.....	317	16.1.1 分析活动者——与系统 交互的人.....	354
14.5 控制实现.....	318	16.1.2 从Use Case分析人机交互.....	355
14.5.1 在程序内进行状态设置.....	318	16.1.3 分析处理异常事件的人机交互.....	358
14.5.2 状态机器引擎.....	319	16.1.4 命令的组织.....	358
14.5.3 控制作为并发任务.....	319	16.1.5 输出信息的组织结构.....	362
14.6 继承的调整.....	319	16.2 界面设计风格.....	362
14.6.1 重新安排类和操作.....	319	16.2.1 菜单的选择.....	363
14.6.2 抽象出公共的行为.....	320	16.2.2 对话框.....	364
14.6.3 使用授权共享实现.....	321	16.2.3 窗口.....	365
14.7 关联设计.....	322	16.3 人机界面的设计准则.....	365
14.7.1 分析关联遍历.....	322	16.4 人机界面设计过程.....	367
14.7.2 单向关联.....	322	16.4.1 用户界面模型.....	367
14.7.3 双向关联.....	323		
14.7.4 链接属性.....	323		

16.4.2 界面支持系统	368	18.3.2 软件质量保证原则	409
16.4.3 界面元素	369	18.3.3 软件质量保证计划	410
16.4.4 设计的形式	369	18.3.4 软件质量保证的措施	410
16.5 描述方法与技术	371	18.3.5 软件质量管理小组	410
16.5.1 灵境技术	372	18.4 技术评审与审查	411
16.5.2 多通道人机交互技术	372	18.4.1 评审过程	411
小结	373	18.4.2 选择参加评审的成员	412
综合练习十六	373	18.4.3 评审的管理和组织	412
一、填空题	373	18.4.4 评审的方法	412
二、选择题	374	18.4.5 走查和审查	412
三、简答题	374	18.4.6 开发过程的评审	413
第 17 章 形式化方法	375	18.4.7 对评审的综合评价	414
17.1 基础知识	375	18.5 软件的可靠性	415
17.1.1 形式化方法概念	375	小结	415
17.1.2 形式化规约语言	376	综合练习十八	415
17.2 有限状态机 (FSM)	378	一、填空题	415
17.3 Petri 网基本原理	379	二、选择题	416
17.3.1 静态结构	379	三、简答题	416
17.3.2 动态特征	380	第 19 章 软件实现	417
17.3.3 转移启动规则	381	19.1 程序设计语言的特性及选择	417
17.3.4 行为特性	383	19.1.1 程序设计语言特性	417
17.3.5 行为特性分析方法	386	19.1.2 程序设计语言的选择	418
17.3.6 结构特性分析方法	392	19.2 程序设计风格	421
17.3.7 Petri 网到程序结构的转换	395	19.3 程序设计效率	423
小结	398	19.3.1 代码效率	423
综合练习十七	398	19.3.2 内存效率	423
一、填空题	398	19.3.3 I/O 效率	424
二、选择题	398	19.4 冗余编程	424
三、简答题	399	19.5 软件容错技术	425
第 18 章 软件质量	400	19.5.1 容错软件	425
18.1 软件质量概述	400	19.5.2 容错的一般方法	426
18.1.1 软件质量的定义	400	19.5.3 容错软件的设计过程	429
18.1.2 软件质量特性	400	19.5.4 软件的容错系统结构	429
18.1.3 软件质量特性之间的竞争	404	小结	432
18.2 软件质量的度量和评价	405	综合练习十九	432
18.2.1 软件质量的度量	405	一、填空题	432
18.2.2 软件质量度量的分类	406	二、选择题	432
18.2.3 软件质量评价	406	三、简答题	433
18.3 软件质量保证	409	第 20 章 软件测试	434
18.3.1 软件质量保证的概述	409	20.1 软件测试概述	434

20.1.1 软件测试的目的.....	434	21.6.1 预防性维护.....	464
20.1.2 软件测试的原则.....	434	21.6.2 逆向工程的元素.....	465
20.2 测试方法.....	435	21.6.3 再工程中的重构技术.....	465
20.2.1 静态测试.....	435	小结.....	468
20.2.2 动态测试.....	435	综合练习二十一.....	469
20.3 测试用例的设计.....	436	一、填空题.....	469
20.3.1 白盒技术.....	436	二、选择题.....	469
20.3.2 黑盒技术.....	441	三、简答题.....	469
20.4 测试过程.....	445	第22章 软件项目管理与计划.....	470
20.4.1 软件测试过程中的信息.....	445	22.1 软件项目管理概述.....	470
20.4.2 软件测试的步骤与各开发 阶段的关系.....	445	22.1.1 软件管理的对象.....	470
20.4.3 单元测试.....	446	22.1.2 软件开发中的资源.....	471
20.4.4 集成测试.....	447	22.1.3 分解技术.....	472
20.4.5 确认测试.....	450	22.2 项目管理过程.....	472
20.5 调试.....	451	22.3 软件开发成本估算.....	473
小结.....	452	22.3.1 软件开发成本估算方法.....	473
综合练习二十.....	452	22.3.2 软件开发成本估算的经验模型.....	474
一、填空题.....	452	22.4 风险分析.....	476
二、选择题.....	453	22.4.1 风险识别.....	477
三、简答题.....	453	22.4.2 风险估算.....	477
第21章 软件维护.....	454	22.4.3 风险评价.....	478
21.1 软件维护概述.....	454	22.4.4 风险驾驭和监控.....	479
21.1.1 软件维护的定义.....	454	22.5 进度安排.....	480
21.1.2 影响维护工作的因素.....	455	22.5.1 软件开发小组人数与软件 生产率.....	481
21.1.3 维护成本.....	455	22.5.2 任务的确定与并行性.....	481
21.2 软件可维护性.....	455	22.5.3 制定开发进度计划.....	482
21.2.1 软件可维护性的定义.....	455	22.5.4 进度安排的图形方法.....	483
21.2.2 可维护性的度量.....	456	22.5.5 项目的追踪和控制.....	484
21.3 软件维护的特点.....	458	22.6 软件项目的组织.....	484
21.3.1 非结构化维护和结构化维护.....	458	22.6.1 软件项目管理的特点.....	484
21.3.2 维护的困难性.....	459	22.6.2 软件项目组织的建立.....	485
21.3.3 软件维护的费用.....	459	22.6.3 人员配备.....	487
21.4 软件维护的实施.....	460	小结.....	489
21.4.1 维护的组织.....	460	综合练习二十二.....	489
21.4.2 维护的流程.....	460	一、填空题.....	489
21.4.3 维护技术.....	461	二、选择题.....	490
21.4.4 维护的副作用.....	462	三、简答题.....	490
21.5 维护“老化代码”.....	463	参考文献.....	491
21.6 逆向工程和再工程.....	464		

第1章 绪 论

软件工程是指导计算机软件开发和维护的一种工程科学，它涉及的知识相当广泛。在学习软件工程之前，必须对软件工程领域的一些基本概念有所了解，对软件工程有一个初步的认识，这样才能顺利地进入后面章节的学习。

1.1 软件概述

计算机系统由硬件和软件两大部分组成。电子计算机自 1946 年诞生后到 20 世纪 60 年代中期是计算机发展的早期。在早期，计算机系统还是以硬件为主，软件费用是总费用的 20% 左右。到了中期（20 世纪 60 年代中期到 20 世纪 80 年代初期），软件费用迅速上升到总费用的 60%，软件不再只是技巧性和高度专业化的神秘机器代码。而 1985 年以后到今天，软件费用已上升 80% 以上。软件相对硬件的费用比例在不断提高。可以从图 1-1 中看到硬件和软件费用比例的变化。

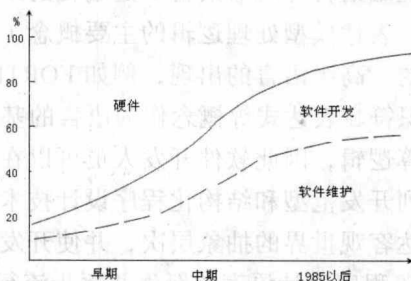


图 1-1 硬件和软件费用比例变化示意图

“软件”的定义是计算机程序及其说明程序的各种文档。在该定义中，“程序”是计算任务的处理对象和处理规则的描述；“文档”是有关计算机程序功能、设计、编制、使用的文字或图形资料。软件与硬件一起构成了计算机系统。

1.1.1 什么是计算机软件

计算机软件定义：

【定义 1-1】计算机运行所需要的各种程序和数据的总称，包括操作系统、汇编程序、编译程序、数据库、文字编辑及维护使用手册等。软件是计算机系统的重要组成部分。

软件是当代计算机行业中的重要产品，但它不是一种有形的物质，它表示的仅仅是一种思想，必须以某种形式表达，通常存储在磁盘（带）介质上或者以文本方式提供，也可以存储在 ROM 中。

计算机软件的主要组成部分是计算机程序。它是指以算法语言为基础而编制的，能够使计算机作出信息处理并产生一定结果的指令或指令组合。计算机程序一般是由文字和符号代码按照一定的规律所组成。计算机程序包括源程序和目标程序。同一程序的源文本和目标文本应当视为同一作品。

根据其表现形式，计算机程序可以分为源程序，即用高级语言或汇编语言编写的程序；

目的程序, 即指源程序经编译或解释加工以后, 可以由计算机直接执行的程序。

通常, 计算机程序要经过编译和链接而成为一种人们不易理解而计算机理解的格式, 然后运行。未经编译就可运行的程序通常称之为脚本程序。

计算机语言: Basic、Visual Basic、C、FoxBase、PowerBuilder、Hyper text 及所见即所得工具 (FrontPage、Dreamweaver、Flash)。

文档: 指用自然语言或者形式化语言所编写的文字资料和图表, 用来描述程序的内容、组成、设计、功能规格、开发情况、测试结果及使用方法。如程序设计说明书、流程图、用户手册等。

1.1.2 软件的特点

软件是对客观世界中问题空间与解空间的具体描述, 是客观事物的一种反映, 是知识的提炼和“固化”。客观世界是不断变化的, 因此, 构造性和演化性是软件的本质特征。如何使软件模型具有更强的表达能力、更符合人类的思维模式, 即如何提升计算环境的抽象层次, 在一定意义上来讲, 这紧紧围绕了软件的本质特征——构造性和演化性。

在高级语言出现以前, 汇编语言 (机器语言) 是编程的工具, 表达软件模型的基本概念 (或语言构造) 的是指令, 表达模型处理逻辑的主要概念 (机制) 的是顺序和转移。显然, 这一抽象层次是比较低的。高级语言的出现, 例如 FORTRAN 语言、PASCAL 语言、C 语言等, 使用了变量、标识符、表达式等概念作为语言的基本构造, 并使用 3 种基本控制结构来表达软件模型的计算逻辑, 因此软件开发人员可以在一个更高的抽象层次上进行程序设计。随后出现了一系列开发范型和结构化程序设计技术, 实现了模块化的数据抽象和过程抽象, 提高了人们表达客观世界的抽象层次, 并使开发的软件具有一定的构造性和演化性。近 20 年来, 面向对象程序设计语言的诞生并逐步流行, 为人们提供了一种以对象为基本计算单元, 以消息传递为基本交互手段来表达的软件模型。面向对象方法的实质是以拟人化的观点来看待客观世界, 即客观世界是由一系列对象构成, 这些对象之间的交互形成了客观世界中各式各样的系统。面向对象方法中的概念和处理逻辑更接近人们解决计算问题的思维模式, 使开发的软件具有更好的构造性和演化性。目前, 人们更加关注软件复用问题, 构建比对象粒度更大、更易于复用的基本单元——构件, 并研究以构件复用为基础的软体构造方法, 更好地凸现软件的构造性和演化特性。易于复用的软件, 一定是具有很好构造性和演化性的软件。

计算机系统软件与硬件是相互依存的, 缺一不可。而软件与其他产品的特点不同。它是一种特殊的产品, 具有下列特殊性质:

(1) 软件产品的生产主要是脑力劳动, 还未完全摆脱手工开发方式, 大部分产品是“定做”的。

(2) 软件是一种逻辑产品, 它与物质产品有很大的区别, 它是脑力劳动的结晶。软件产品是看不见摸不着的, 因而具有无形性。它以程序和文档的形式出现, 保存在存储介质上, 通过计算机的运行才能体现它的功能和作用。

(3) 软件产品不会用坏, 不存在磨损、消耗问题。

(4) 软件产品的生产主要是研制。其成本主要体现在软件的开发和研制上, 软件开发研制完成后, 通过复制就产生了大量软件产品。

(5) 软件费用不断增加, 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动, 它的成本非常高。

1.1.3 软件的分类

上世纪计算机产生以来, 人们围绕着它开发了大量的软件, 广泛应用于科学研究、教育、工农业生产、事务处理、国防和家庭等众多领域, 积累了丰富的软件资源。然而, 在软件的品种质量和价格方面仍然满足不了人们日益增长的需要。计算机软件产业是一项年轻的、充满活力的飞速发展的产业。因此, 关于其分类方法不同, 所分类型差别也很大。这里简单地介绍计算机软件在计算机系统、实时系统、嵌入式系统、科学和工程计算、事务处理、人工智能、个人计算机和计算机辅助软件工程 (CASE) 等方面的应用。

按照计算机的控制层次, 计算机软件分为系统软件和应用软件两大类。

1. 系统软件

计算机系统软件是计算机管理自身资源 (如 CPU、内存空间、外存、外部设备等), 提高计算机的使用效率并为计算机用户提供各种服务的基础软件。系统软件依赖于机器的指令系统、中断系统以及运算、控制、存储部件和外部设备。系统软件要为各类用户提供尽可能标准、方便的服务, 尽量隐藏计算机系统的某些特征或实现细节。因此, 系统软件是计算机系统的重要组成部分, 它支持应用软件的开发和运行。系统软件包括操作系统、网络软件、各种语言的编译程序、数据库管理系统、文件编辑系统、系统检查与诊断软件等等。

(1) 操作系统。DOS: 是基于字符界面的单用户单任务的操作系统; Windows: 是基于图形界面的单用户多任务的操作系统; UNIX: 是一个通用的交互式的分时操作系统, 用于各种计算机; NetWare: 是基于文件服务和目录服务的网络操作系统; Windows NT: 是基于图形界面 32 位多任务、对等的网络操作系统。

(2) 语言处理程序。机器语言: 计算机能直接执行的、由一串“0”或“1”所组成的二进制程序或指令代码, 是一种低级语言。汇编语言: 一种用符号表示的、面向机器的低级程序设计语言, 需经汇编程序翻译成机器语言程序才能被计算机执行。高级语言: 按照一定的“语法规则”、由表达各种意义的“词”和“数学公式”组成的、易被人们理解的程序设计语言, 需经编译程序翻译成目标程序 (机器语言) 才能被计算机执行。如: FORTRAN、C、BASIC 等。

(3) 数据库管理系统。普及式关系型: FoxPro、Paradox、Access; 大型关系型: Oracle、Sybase、SQL Server。

(4) 实用程序与软件工具: QAPLUS、PCTOOLS。

2. 应用软件

应用软件是计算机所应用程序的总称, 主要用于解决一些实际的应用问题。按业务、行业, 应用软件也可分为:

(1) 个人计算机软件。个人计算机上使用的软件也可包括系统软件和应用软件两类。近 20 年来, 个人计算机的处理能力已提高了三个数量级, 以前在中小型计算机上运行的系统软件和应用软件, 如今已经大量移植到个人计算机上。在个人计算机上开发了大量的文字处理软件 (Word、WPS)、图形处理软件 (AutoCAD、Photoshop)、报表处理软件 (Excel、

Lotus1-2-3)、个人和商业上的财务处理软件、数据库管理软件、网络软件(Terminal、Mail)、简报软件(Powerpoint)、统计软件(SPPA、SAS)、多媒体技术(Xingmpeg、Authorware、Director)。使个人计算机具有了用文字、图形、声音进行人机交互的能力,为个人计算机的普及创造了必要条件。人们还将个人计算机与计算机网络连接在一起,进行通信和共享网络资源,加速了人类社会信息化的进程。随着社会的进步,个人计算机及其软件的发展、普及和应用前景将更加广阔。

(2) 科学和工程计算软件。它们以数值算法为基础,对数值量进行处理的计算,主要用于科学和工程计算,例如数值天气预报、弹道计算、石油勘探、地震数据处理计算机系统仿真和计算机辅助设计(CAD)等。这类软件大多数用FORTRAN语言描述,近年来有的也用C语言或Ada语言描述。它是使用最早、最广泛、最为成熟的一类软件。从20世纪50年代起,有经验的程序员就把许多常用算法用程序设计语言编制成标准程序,如今已经积累了大量的科学和工程计算软件。人们将各种软件按学科或应用领域分类,开发了各种程序库、软件包和软件系统。这些软件具有质量高、使用方便等特点,为计算机在科学和工程上的应用做出了重要贡献。

(3) 实时软件(FIX、INTouch、Lookout)。监视、分析和控制现实世界发生的事件,能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的软件,称之为实时软件。实时软件依赖于处理机系统的物理特性,如计算速度和精度、I/O信息处理与中断响应方式、数据传输效率等。支持实时软件的操作系统称为实时操作系统。实时软件使用的计算机语言有汇编语言、Ada语言等。实时系统的服务经常是连续的,系统在规定的时间内必须处于能够响应的状态,因此,实时软件和计算机系统必须有很高的可靠性和安全性。

(4) 人工智能软件。支持计算机系统产生人类某些智能的软件。它们求解复杂问题时,不是采用传统的计算或分析方法,而是采用诸如基于规则的演绎推理技术和算法,在很多场合还需要知识库的支持。人工智能软件常用的计算机语言有LISP和Prolog等。迄今为止,在专家系统、模式识别、自然语言理解、人工神经网络、程序验证、自动程序设计、机器人学等领域开发了许多人工智能应用软件,用于诊断疾病、产品检测、自动定理证明、图像和语音的自动识别、语言翻译等。

(5) 嵌入式软件。嵌入式计算机系统将计算机嵌入在某一系统之中,使之成为该系统的重要组成部分控制该系统的运行,进而实现一个特定的物理过程。用于嵌入式计算机系统的软件称为嵌入式软件。大型的嵌入式计算机系统软件可用于航空航天系统、指挥控制系统和武器系统等。小型的嵌入式计算机系统软件可用于工业的智能化产品之中,这时嵌入式软件驻留在只读存储器内,为该产品提供各种控制功能和仪表的数字或图形显示功能等。例如,汽车的刹车控制,空调机、洗衣机的自动控制等等。嵌入式计算机系统一般都要和各种仪器、仪表、传感器连接在一起,因此,嵌入式软件必须具有实时的采集、处理和输出数据的能力,这样的系统称之为实时嵌入式系统。它广泛应用于连续的动力学系统的控制与仿真。

(6) 事务处理软件。用于处理事务信息,特别是商务信息的计算机软件。事务信息处理是软件最大的应用领域,它已由初期零散、小规模软件系统,如工资管理系统、人事档案管理系统等,发展成为管理信息系统(MIS),如世界范围内的飞机订票系统、旅馆