

**研究生教学用书**

教育部研究生工作办公室推荐

# 软件工程

Software Engineering

齐治昌 谭庆平 宁 洪 编著

高等教育出版社

## 内容简介

本书比较全面、系统地介绍了软件工程的概 念、技术与方法,内容包括:软件开发模型,软件项目管理,计算机系统工程,需求分析,软件设计,编码与测试,软件维护,配置管理,软件重用,快速原型,集成化 CASE 环境,Java 与 Internet 环境下的软件开发,对象链接与嵌入(OLE)技术,CORBA 与分布计算技术。书中含有丰富的例题与习题,便于教学与自学。

本书强调理论与实践相结合,软件开发方法与 CASE 工具相结合,选材新颖,语言精练,条理性和逻辑性强,可作为研究生或高等学校计算机专业高年级学生教材,也可作为软件开发人员的参考书。

## 图书在版编目(CIP)数据

软件工程/齐治昌,谭庆平,宁洪编著。—北京:高等

教育出版社,2001.8(2002 重印)

ISBN 7-04-009909-8

I. 软… II. ①齐…②谭…③宁… III. 软件工程 IV. TP311

中国版本图书馆 CIP 数据核字(2001)第 18735 号

责任编辑 肖子东 封面设计 李卫青 版式设计 史新薇

责任校对 存 怡 责任印制 杨 明

软件工程

齐治昌 谭庆平 宁洪 编著

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

传 真 010-64014048

购书热线 010-64054588

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京联华印刷厂

开 本 787×960 1/16

印 张 30.75

字 数 409 000

版 次 2001 年 8 月第 1 版

印 次 2002 年 9 月第 4 次印刷

定 价 32.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

**版权所有 侵权必究**

# 序

软件工程是计算机学科中一个年轻并且充满活力的研究领域。自 20 世纪 60 年代末期以来,人们为克服“软件危机”在这一领域做了大量工作,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了重要作用。今天,现代科学技术将人类带入了信息社会,计算机软件扮演着十分重要的角色,软件工程已成为信息社会高技术竞争的关键领域之一。

“软件工程”是高等学校计算机教学计划中的一门核心课程,主要内容包括支持软件开发和维护的理论、方法、技术、标准以及计算机辅助工具和环境。这些内容对于软件研制人员、软件项目管理人员都是必需的。本书比较全面、系统地反映了软件工程课程的全貌,既兼顾了传统的、实用的软件开发方法,又介绍了软件工程领域比较新颖的技术和方法,包括面向对象的需求分析与软件设计方法,实时系统的分析与设计技术,软件重用技术,快速原型技术和集成化软件开发环境。对计算机辅助软件工程(CASE)技术的介绍贯串全书的始终,对近年来兴起的主要研究热点,如 Java 与 Internet 环境下的软件开发、对象链接与嵌入(OLE)技术、通用对象请求代理机制(CORBA)与分布计算技术等,都做了专门介绍,并附有较完整的参考文献以供有兴趣的读者进一步学习和研究。本书的另一重要特点是理论与实践相结合,软件工程的技术、方法与 CASE 工具相结合。全书内容的选材强调实用价值和可操作性,强调 CASE 工具和环境对软件开发全过程的支持。此外,作者在本书的写作过程中深感工程的内容较易流于琐碎、冗繁,往往给读者造成软件工程缺乏深度、缺乏内在逻辑关联等印象。所以,本书力求语言的精练,注重内容的条理性、系统性和逻辑性。当然,作者的这一初衷是否已经实现还取决于读者的评论。

本书共二十章,大致分为五个部分。前三章构成本书的第一部分,内容包括:软件工程的基本概念,软件开发模型,软件的度量与估算,开发过程管理与质量控制,计算机系统工程。第四至七章构成本书的需求分析部分,内容包括:需求分析的任务与原则,面向数据流、面向对象与面向数据的需求分析方法,形式化方法,需求规格说明与评审。第八至十二章构

成本书的软件设计部分,内容包括:软件设计过程与一般性技术,面向数据流、面向对象与面向数据的设计方法,人机界面设计技术,设计规格说明与评审。第十三至十六章构成本书的软件实现与维护部分,内容包括:程序设计语言与编码,软件测试与排错,软件维护,逆向工程与重构工程,软件配置管理。第十七至二十章构成本书的最后一部分,专门介绍比较新颖、颇具发展潜力的软件开发技术,内容包括:软件重用技术,快速原型技术,集成化 CASE 环境以及新近出现的热点技术。

本书主要供计算机专业硕士研究生作为软件工程课程的教材使用,同时,本书也适合于软件开发人员与软件项目管理人员作为技术参考书使用。在教学计划中,如果安排 60 学时,建议采用第一至第二十章的自然顺序讲授,其中带“\*”的内容可酌情处理。实习以 8 至 12 学时为宜。如果以 40 学时讲授本书,对于高年级本科生,建议的教学内容及次序为:第一、二章→第四至第六章→第八至十章→第十二至十五章→第十七、十八章;对于已在本科阶段学过软件工程的硕士研究生,建议的教学内容及次序为:第二、三章→第四、第六章及第七章中的形式化方法→第十六章→第十七至十九章→第二十章\*。实习以 6~10 学时为宜。

齐治昌教授组织了本书的编写工作,并撰写了前三章。第四至七章及第十七至二十章由谭庆平博士撰写,第八至十六章由宁洪副教授撰写。本书内容曾在国防科学技术大学计算机专业本科生和硕士研究生教学中讲授过。

西北大学郝克刚教授认真审阅了本书的全部初稿,并提出了许多中肯的修改意见。国防科学技术大学计算机系陈怀义教授、殷建平博士也对书稿进行了认真细致的初审,并提出了很好的意见和建议。陈火旺教授在本书的成书过程中给予了多方面指导。此外,国防科学技术大学计算机系领导对本书的编写工作给予了有力支持,计算机系很多教师和学生的鼓励和建议也对本书的出版起了推动作用。在此,作者向所有对本书编写工作给予支持和帮助的人表示衷心的感谢。最后,作者还要特别感谢施伯乐教授、许卓群教授以及教育部计算机科学教学指导委员会的其他专家们,没有他们的信任、鼓励和支持,本书是不可能问世的。

最后,诚恳欢迎各位读者对本书的缺点、错误提出批评。

作者

2001 年 1 月

# 目 录

<b>第一章 软件与软件工程</b> .....	1
1.1 软件 .....	2
1.1.1 软件与软件的组成 .....	2
1.1.2 软件的特点 .....	3
1.1.3 软件分类 .....	4
1.1.4 软件的发展 .....	6
1.1.5 软件危机 .....	8
1.2 软件工程的定义 .....	10
1.2.1 软件工程的定义 .....	10
1.2.2 软件工程的目标 .....	11
1.2.3 软件工程的原理 .....	13
1.3 软件生存周期 .....	14
1.3.1 软件定义 .....	14
1.3.2 软件开发 .....	16
1.3.3 软件使用、维护和退役 .....	18
1.4 软件开发模型 .....	19
1.4.1 瀑布模型 .....	20
1.4.2 原型模型 .....	21
1.4.3 螺旋模型 .....	22
1.4.4 基于四代技术的模型 .....	23
1.4.5 变换模型 .....	24
1.4.6 组合模型 .....	25
1.5 CASE 工具及环境 .....	26
1.5.1 计算机辅助软件工程 .....	27
1.5.2 CASE 工具 .....	27
1.5.3 集成化的 CASE 环境 .....	29
小结 .....	30
习题 .....	30
<b>第二章 软件项目管理</b> .....	32
2.1 软件度量 .....	32

2.1.1	度量、测量和估算 .....	33
2.1.2	面向规模的度量 .....	34
2.1.3	面向功能的度量 .....	35
2.1.4	代码行度量与功能点度量的比较 .....	38
2.2	软件项目估算 .....	39
2.2.1	代码行、功能点和工作量估算 .....	40
2.2.2	经验估算模型之一:CoCoMo 模型 .....	41
2.2.3	经验估算模型之二:Putnam 模型 .....	45
2.3	软件质量度量 .....	47
2.3.1	软件质量定义及三层次度量模型 .....	47
2.3.2	软件质量要素 .....	48
2.3.3	软件质量要素评价准则 .....	49
2.4	软件复杂性度量 .....	54
2.4.1	软件复杂性及度量原则 .....	54
2.4.2	控制结构的复杂性度量 .....	54
2.4.3	文本复杂性度量 .....	55
2.5	软件可靠性度量 .....	57
2.5.1	软件可靠性的概念 .....	57
2.5.2	软件修复和软件有效性 .....	59
2.5.3	软件可靠性估算 .....	60
2.6	软件开发过程的管理 .....	63
2.6.1	风险分析 .....	63
2.6.2	进度安排 .....	67
2.6.3	软件开发标准 .....	70
2.6.4	软件质量保证 .....	72
2.6.5	软件开发人员的组织与分工 .....	74
2.6.6	软件项目的开发过程管理 .....	76
2.7	软件项目管理中的 CASE 工具 .....	76
	小结 .....	77
	习题 .....	77
<b>第三章</b>	<b>计算机系统工程 .....</b>	<b>79</b>
3.1	计算机系统工程 .....	80
3.1.1	硬件和硬件工程 .....	80
3.1.2	软件和软件工程 .....	81
3.1.3	人机工程 .....	84
3.1.4	数据库工程 .....	85

3.2	可行性研究	86
3.2.1	引言	86
3.2.2	经济可行性	87
3.2.3	技术可行性	89
3.2.4	方案选择	90
3.3	系统模型与模拟	92
3.3.1	系统模型	92
3.3.2	系统建模和模拟	94
3.4	系统规格说明及评审	96
3.4.1	系统规格说明	96
3.4.2	系统规格说明评审	97
	小结	97
	习题	98
<b>第四章</b>	<b>需求分析基础</b>	99
4.1	分析的任务与原则	99
4.2	初步需求获取技术	101
4.2.1	访谈与会议	101
4.2.2	观察用户工作流程	101
4.2.3	用户和开发人员共同组成联合小组	102
4.2.4	实例分析	102
4.3	需求建模	103
4.4	问题抽象、问题分解与多视点分析	104
4.5	支持需求分析的快速原型技术	105
4.6	需求规格说明与评审	106
4.6.1	需求规格说明书的目标与内容	106
4.6.2	需求评审	109
	小结	110
	习题	110
<b>第五章</b>	<b>面向数据流的分析方法</b>	114
5.1	数据流图与数据字典	114
5.2	实体-关系图	117
5.2.1	数据对象、属性与关系	117
5.2.2	实体-关系图	118
5.3	数据流图的实时系统扩充*	120
5.3.1	Ward & Mellor 扩充	120
5.3.2	Hatley & Pirhai 扩充	122

5.4	基于数据流的分析方法 .....	126
5.4.1	创建数据流模型 .....	126
5.4.2	创建控制流模型* .....	127
5.4.3	过程规格说明 .....	128
5.5	基于数据流图的需求分析 CASE 工具* .....	130
5.5.1	核心思想 .....	130
5.5.2	语言机制 .....	132
5.5.3	动态分析 .....	138
5.5.4	基于 CASE 工具的需求分析 .....	139
	小结 .....	140
	习题 .....	140
<b>第六章</b>	<b>面向对象的需求分析 .....</b>	<b>142</b>
6.1	面向对象的概念与思想 .....	142
6.2	面向对象的分析方法 .....	144
6.2.1	识别对象 .....	144
6.2.2	标识对象的属性 .....	146
6.2.3	识别对象的行为 .....	147
6.2.4	识别对象所属的类 .....	153
6.2.5	定义主题词 .....	155
6.2.6	多视点需求分析 .....	156
6.3	面向对象的需求分析 CASE 工具 .....	158
6.3.1	概述 .....	158
6.3.2	对象及消息传递图 .....	159
6.3.3	类结构图 .....	161
6.3.4	对象状态转换图 .....	161
6.3.5	主题词结构图 .....	162
6.3.6	基于 CASE 工具的需求分析 .....	162
	小结 .....	163
	习题 .....	164
<b>第七章</b>	<b>面向数据的分析方法与形式化方法 .....</b>	<b>165</b>
7.1	面向数据结构的系统开发方法 .....	165
7.1.1	Warnier 图 .....	165
7.1.2	DSSD 方法 .....	166
7.2	Jackson 系统开发方法 .....	169
7.2.1	标识实体与行为 .....	169
7.2.2	生成实体结构图 .....	170

7.2.3	创建软件系统模型 .....	171
7.3	形式化方法* .....	173
7.3.1	主要思想 .....	173
7.3.2	形式化规格说明语言简介 .....	175
7.3.3	形式化需求描述 .....	176
7.3.4	形式化方法的现状与发展趋势 .....	183
小结	.....	184
习题	.....	184
<b>第八章</b>	<b>软件设计基础</b> .....	<b>186</b>
8.1	软件设计过程 .....	186
8.2	软件设计基本概念 .....	187
8.2.1	抽象与逐步求精 .....	187
8.2.2	模块化与信息隐藏 .....	190
8.2.3	软件总体结构设计 .....	193
8.2.4	数据结构设计 .....	195
8.2.5	软件过程设计 .....	196
8.3	过程设计技术和工具 .....	197
8.3.1	结构化程序设计 .....	197
8.3.2	图形表示法 .....	198
8.3.3	判定表 .....	200
8.3.4	过程设计语言(PDL).....	201
8.3.5	过程设计工具之比较 .....	204
8.4	设计规格说明与评审 .....	205
小结	.....	207
习题	.....	207
<b>第九章</b>	<b>面向数据流的设计方法</b> .....	<b>209</b>
9.1	基本概念和设计过程 .....	209
9.2	变换分析 .....	211
9.3	事务分析 .....	217
9.4	启发式设计策略 .....	221
9.5	设计优化原则 .....	223
9.6	实时系统设计 .....	224
9.6.1	实时系统性能要求 .....	224
9.6.2	实时系统设计要素 .....	224
9.6.3	实时系统设计方法 .....	225
9.6.4	设计实例 .....	227

小结 .....	230
习题 .....	230
<b>第十章 面向对象的设计方法 .....</b>	<b>233</b>
10.1 面向对象设计的概念 .....	234
10.1.1 对象、操作和消息 .....	234
10.1.2 类、实例和继承 .....	235
10.1.3 对象描述 .....	235
10.2 面向对象的设计方法 .....	236
10.2.1 主体部件(PDC)的设计 .....	237
10.2.2 用户界面部件(HIC)的设计 .....	242
10.2.3 任务管理部件(TMC)的设计 .....	244
10.2.4 数据管理部件(DMC)的设计 .....	246
10.3 基于对象的设计方法 .....	248
10.3.1 设计步骤 .....	248
10.3.2 详细设计 .....	251
小结 .....	252
习题 .....	253
<b>第十一章 面向数据的设计方法 .....</b>	<b>255</b>
11.1 面向数据设计 .....	255
11.2 Jackson 系统开发方法 .....	256
11.2.1 JSD 分析技术回顾 .....	256
11.2.2 扩充功能性过程 .....	257
11.2.3 施加时间约束 .....	262
11.2.4 实现与 JSP 技术 .....	263
11.2.5 产生过程表示 .....	265
11.3 基于结构化数据的系统开发(DSSD)方法 .....	266
11.3.1 DSSD 设计步骤 .....	267
11.3.2 推导输出数据的逻辑结构 .....	267
11.3.3 推导处理过程的逻辑结构 .....	268
11.3.4 复杂过程逻辑的描述 .....	270
小结 .....	271
习题 .....	271
<b>第十二章 人机界面设计 .....</b>	<b>273</b>
12.1 人的因素 .....	273
12.1.1 人类感知基础 .....	273
12.1.2 用户的技能 .....	274

12.1.3 任务与用户的特殊要求 .....	275
12.2 人机界面风格 .....	275
12.3 人机界面设计过程 .....	276
12.3.1 界面设计的有关模型 .....	277
12.3.2 任务分析与建模 .....	277
12.3.3 界面设计的一般问题 .....	278
12.3.4 实现工具 .....	279
12.4 人机界面实现的原则 .....	280
12.4.1 一般可交互性 .....	280
12.4.2 信息显示 .....	280
12.4.3 数据输入 .....	281
12.5 人机界面标准 .....	281
12.6 支持界面设计的 CASE 工具 .....	282
小结 .....	286
习题 .....	286
<b>第十三章 程序设计语言和编码</b> .....	<b>288</b>
13.1 程序设计语言 .....	288
13.1.1 程序设计语言的特性 .....	288
13.1.2 程序设计语言的基本机制 .....	290
13.1.3 程序设计语言的演变和分类 .....	292
13.1.4 程序设计语言的选择 .....	293
13.2 程序设计过程 .....	294
13.2.1 面向对象语言对 OOD 的支持 .....	295
13.2.2 基于对象语言对 OOD 的支持 .....	297
13.2.3 过程式语言对 OOD 的支持 .....	300
13.3 编程标准 .....	302
13.4 编程风格 .....	303
13.5 程序设计支持环境(PSE) .....	305
小结 .....	307
习题 .....	307
<b>第十四章 软件测试</b> .....	<b>309</b>
14.1 基本概念 .....	309
14.1.1 软件测试的目标 .....	309
14.1.2 测试阶段的信息流程 .....	310
14.1.3 测试用例和场景的设计 .....	311
14.1.4 软件测试的步骤 .....	312

14.2	软件测试技术	313
14.2.1	白盒测试	313
14.2.2	黑盒测试	322
14.3	软件测试策略	324
14.3.1	单元测试	324
14.3.2	综合测试	327
14.3.3	确认测试	331
14.3.4	系统测试	332
14.3.5	排错	333
14.4	基于 CASE 工具的软件测试和排错	336
14.4.1	自动测试工具	336
14.4.2	调试器	336
	小结	338
	习题	338
<b>第十五章</b>	<b>软件维护</b>	<b>342</b>
15.1	软件维护的分类	342
15.2	维护过程	343
15.2.1	结构化与非结构化的维护	343
15.2.2	维护的成本	344
15.2.3	可能存在的问题	345
15.3	可维护性	346
15.3.1	影响可维护性的因素	346
15.3.2	若干量化的测度	346
15.3.3	保证可维护性的复审	347
15.4	维护活动	347
15.4.1	维护组织	347
15.4.2	维护的报告与评估	348
15.4.3	维护活动的事件流	349
15.4.4	保存维护记录	350
15.4.5	评价维护活动	351
15.5	维护的副作用	352
15.6	逆向工程与重构工程	353
15.6.1	恢复信息的级别	353
15.6.2	恢复信息的方法	354
	小结	355
	习题	356

<b>第十六章 软件配置管理</b> .....	357
16.1 软件配置管理 .....	357
16.1.1 基线技术 .....	358
16.1.2 软件配置项 .....	358
16.2 软件配置管理任务 .....	360
16.2.1 标识配置对象 .....	360
16.2.2 版本控制 .....	362
16.2.3 系统建立 .....	362
16.2.4 修改控制 .....	362
16.2.5 配置审计 .....	364
16.2.6 配置状况报告 .....	365
16.3 软件配置管理标准 .....	365
16.4 配置管理的 CASE 工具 .....	366
小结 .....	368
习题 .....	368
<b>第十七章 软件重用技术</b> .....	369
17.1 软件重用 .....	370
17.1.1 软件重用的概念 .....	370
17.1.2 软件重用的过程与意义 .....	370
17.1.3 重用项目的管理 .....	371
17.2 软部件库的构造 .....	372
17.2.1 域分析 .....	372
17.2.2 软部件的开发 .....	374
17.2.3 软部件库的组织 .....	377
17.3 软部件的重用 .....	380
17.3.1 检索与提取软部件 .....	380
17.3.2 理解与评价软部件 .....	382
17.3.3 修改软部件 .....	383
17.3.4 软部件的合成 .....	383
17.4 面向对象的软件重用技术 .....	384
17.4.1 类库的构造 .....	384
17.4.2 类库的检索 .....	384
17.4.3 面向对象的合成 .....	385
17.5 软件重用项目的管理 .....	389
17.5.1 组织机构 .....	389
17.5.2 软件重用的考核指标 .....	390

17.5.3	创造重用氛围 .....	391
17.6	支持软件重用的 CASE 工具 .....	391
17.6.1	软件重用对 CASE 工具的需求 .....	391
17.6.2	类库的组织与检索工具 .....	393
小结	.....	395
习题	.....	395
<b>第十八章</b>	<b>快速原型技术</b> .....	<b>397</b>
18.1	瀑布模型的缺陷 .....	397
18.2	快速原型方法 .....	400
18.2.1	原型及其作用 .....	400
18.2.2	快速原型的构造过程 .....	401
18.3	基于快速原型的进化式软件开发 .....	403
18.3.1	螺旋模型 .....	403
18.3.2	螺旋模型的评价 .....	406
18.4	快速原型的技术支持 .....	407
18.4.1	用户界面自动生成工具 .....	407
18.4.2	面向数据库应用的开发工具 .....	410
18.4.3	四代语言 .....	415
18.4.4	可重用工具 .....	416
18.4.5	程序设计环境 .....	416
小结	.....	418
习题	.....	418
<b>第十九章</b>	<b>集成化 CASE 环境</b> .....	<b>420</b>
19.1	概述 .....	420
19.2	CASE 工具的集成形式 .....	421
19.2.1	信息交换 .....	421
19.2.2	公共界面 .....	422
19.2.3	公共信息管理与信息共享 .....	422
19.2.4	高度集成 .....	423
19.3	集成化 CASE 环境的总体结构 .....	424
19.3.1	界面集成 .....	424
19.3.2	工具集成 .....	425
19.3.3	信息集成 .....	426
19.3.4	软件工程信息库 .....	427
19.4	可移植的通用工具环境 PCTE .....	430
19.4.1	基本机制 .....	430

19.4.2 用户接口 .....	431
19.4.3 分布式机制 .....	432
小结 .....	432
习题 .....	433
<b>第二十章 新型软件开发技术 *</b> .....	<b>434</b>
20.1 Internet 与Java简介 .....	434
20.1.1 Internet 简介 .....	434
20.1.2 Java的发展历史 .....	436
20.1.3 Java的特征 .....	437
20.1.4 Java的语法机制 .....	439
20.1.5 Java的意义 .....	445
20.2 Java与 Internet 环境下的软件开发 .....	446
20.2.1 基于Java的软件开发过程 .....	446
20.2.2 Java与多媒体主页的制作 .....	447
20.2.3 Java与交互式主页的制作 .....	453
20.2.4 Java与图形界面 .....	454
20.2.5 Java与网络程序设计 .....	458
20.3 对象链接与嵌入(OLE)技术 .....	459
20.3.1 OLE 简介 .....	459
20.3.2 构件对象模型 .....	460
20.3.3 结构化存储与复合文件 .....	461
20.3.4 数据对象传送 .....	462
20.3.5 复合文档 .....	464
20.3.6 OLE 自动化 .....	465
20.3.7 OLE 控件 .....	466
20.4 CORBA 与分布计算技术 .....	466
20.4.1 分布计算环境 .....	467
20.4.2 分布计算技术 .....	467
20.4.3 CORBA 概述 .....	469
20.4.4 Internet 环境下的分布式软件开发 .....	470
小结 .....	471
习题 .....	472
<b>参考文献</b> .....	<b>474</b>

# 第一章 软件与软件工程

计算机的发展一直与微电子技术的进步紧密地联系在一起。自1946年计算机诞生以来,它已走过了电子管、晶体管、集成电路、大规模集成电路的时代,如今超大规模集成电路在计算机工业界已得到广泛的应用。据统计,计算机的性能平均每18个月提高一倍,呈几何级数增长。如果这样的趋势再持续20年,计算机的性能将比现在提高10 000倍。然而,随着计算机性能的提高,计算机的体积、功耗、价格却不断下降。今天的微处理器和个人计算机已广泛应用于办公自动化、工业自动控制、商业信息处理、家用电器控制等领域,大型机、巨型机正成功地担负着数值天气预报、石油勘探、地震数据处理、航天飞机发射等领域的任务。计算机和电视、汽车、电话一样,已成为人们工作和生活不可缺少的工具。

随着计算机技术、电子技术的惊人进步,计算机与全球互连网络Internet相连接,使今天的社会进入了以计算机为核心的信息社会。在信息社会中,信息的获取、处理、交流和决策都需要大量高质量的计算机软件,这样就促使人们对计算机软件的品种、数量、功能、质量、成本和开发时间等提出越来越高的要求。为了使世界上丰富的软件资源为人类共享,人们越来越重视软件、软件开发及运行环境的标准化。计算机的各类程序设计语言 and 多媒体人机交互工具已被越来越多的人所掌握,成为世界性的文化现象。

然而,不幸的是,要想使软件功能越强、使用越方便,开发出来的软件就越复杂、越庞大,人们的软件开发能力越显得力不从心,以致软件开发计划一拖再拖,成本失去控制,软件质量得不到保证。为了扭转这种被动局面,自20世纪60年代末期以来,人们十分重视软件开发方法、工具和环境的研究,并在这些领域取得了重要的成果。

这一章介绍软件和软件工程的基本概念,包括软件、软件工程、软件开发过程与模型、软件工具与环境,等等。

## 1.1 软 件

本节讨论软件的定义、组成、特点和软件的分类,介绍软件的发展过程与软件危机。

### 1.1.1 软件与软件的组成

计算机软件是与计算机系统操作有关的程序、规程、规则及任何与之有关的文档及数据。它由两部分组成:一是机器可执行的程序及有关数据;二是机器不可执行的,与软件开发、运行、维护、使用和培训有关的文档。

程序(program)是用程序设计语言描述的、适合于计算机处理的语句序列。它是软件开发人员根据用户需求开发出来的。程序设计语言编译器可以将程序翻译成一组机器可执行的指令。这组指令亦称机器语言程序,它将根据用户的需求,控制计算机硬件的运行,处理用户提供的或机器运行过程中产生的各类数据并输出结果。为了对程序设计语言进行机器自动翻译,人们不得不限制程序设计语言的词汇范围(如字符集、关键字等),并用良好的形式规则精确地定义程序设计语言的语法和语义。目前的程序设计语言有三种类型:依赖于具体计算机的机器语言、汇编语言,独立于机器的面向过程的语言,以及独立于机器的面向问题的语言。机器语言是用中央处理器(CPU)指令集表示的符号语言,优秀的软件开发人员使用机器语言可以开发出时空开销较小的高质量程序。但是,用机器语言编写程序时工作效率低,程序难以阅读和调试,不利于软件的维护,也难以在不同CPU系统中推广使用。高级语言与机器无关,其表达能力强,容易阅读和修改,大大提高了软件开发效率。高级语言的编译器或解释器是依赖于具体机器的,它把高级语言程序转换为机器语言程序,然后再运行。今天,世界上的程序设计语言有几百种,但广泛使用的高级语言不过十余种,如用于科学计算的FORTRAN语言,用于事务处理的COBOL语言,支持结构化程序设计的Pascal语言,支持现代软件开发的C语言、Ada语言,还有支持面向对象设计方法的C++语言,等等。机器语言、汇编语言和高级语言经常称为前三代的计算机语言或面向过程的