



21st CENTURY
规划教材

面向21世纪高职高专计算机系列规划教材
COURSES FOR VOCATIONAL HIGHER EDUCATION, COMPUTER

软件工程基础教程

SOFTWARE ENGINEERING FUNDAMENTALS

郭庚麒 主 编
余明艳 杨 丽 副主编



科学出版社
www.sciencep.com



面向21世纪高职高专计算机系列规划教材

COURSES FOR VOCATIONAL HIGHER EDUCATION: COMPUTER

软件工程基础教程

郭庚麒 主编

余明艳 杨 丽 副主编

2004年8月第1版 2004年8月第1次印刷

ISBN 7-03-014207-1

W·71311.2

中国版本图书馆CIP数据核字(2004)第084330号

责任编辑: 郭庚麒
封面设计: 北京理工大学出版社

科学出版社

北京

www.sciencep.com

http://www.sciencep.com

北京理工大学出版社

北京理工大学出版社

2004年8月第1版 2004年8月第1次印刷

ISBN 7-03-014207-1

科学出版社

北京

北京理工大学出版社

内 容 简 介

本书主要介绍现代软件工程的基本原理、方法和应用。内容包括：软件工程概述、软件开发模型、系统分析、系统设计、软件编码与测试、软件维护、配置管理、质量管理、文档技术、软件复用和软件工程环境等。根据软件工程的最新发展，着重介绍了面向对象的开发方法，并将 UML 建模技术应用到实际系统的分析和设计中。

本书语言简练，通俗易懂，采用案例教学方法；注重培养实际开发能力和文档的写作能力。本书具有很强的实用性，可作为高职高专院校计算机专业或信息类专业的教材，也可作为软件开发人员的参考书。

图书在版编目(CIP)数据

软件工程基础教程/郭庚麒主编. —北京:科学出版社, 2004

(面向 21 世纪高职高专计算机系列规划教材)

ISBN 7-03-014207-1

I. 软… II. 郭… 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(2004)第 084330 号

责任编辑:舒 立 孙露露/责任校对:柏连海

责任印制:吕春珉/封面设计:飞天创意

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

世界知识印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004年8月第一版 开本:787×1092 1/16

2004年8月第一次印刷 印张:15 3/4

印数:1—3 000 字数:348 000

定价:21.00 元

(如有印装质量问题,我社负责调换(世知))

面向 21 世纪高职高专规划教材专家委员会

主 任 李宗尧

副主任 (按姓氏笔画排序)

丁桂芝 叶小明 张和平 林 鹏

黄 藤 谢培苏

委 员 (略)

信息技术系列教材编委会

主 任 丁桂芝

副主任 (按姓氏笔画排序)

万金保 方风波 徐 红 鲍 泓

委 员 (按姓氏笔画排序)

于晓平	马国光	仁英才	王东红	王正洪
王 玉	王兴宝	王金库	王海春	王爱梅
邓 凯	付百文	史宝会	本柏忠	田 原
申 勇	任益夫	刘成章	刘克敏	刘甫迎
刘经玮	刘海军	刘敏涵	安志远	许殿生
何瑞麟	余少华	吴春英	吴家碚	吴瑞萍
宋士银	宋锦河	张红斌	张环中	张海鹏
张蒲生	张德实	李云程	李文森	李 洛
李德家	杨永生	杨 闯	杨得新	肖石明
肖洪生	陈 愚	周子亮	周云静	胡秀琴
赵从军	赵长旭	赵动庆	郝 梅	唐铸文
徐洪祥	徐晓明	袁德明	郭庚麒	高延武
高爱国	康桂花	戚长政	曹文济	黄小鸥
彭丽英	董振珂	蒋金丹	韩银峰	魏雪英

出版前言

随着世界经济的发展，人们越来越深刻地认识到经济发展需要的人才多元化、多层次的，既需要大批优秀的理论性、研究性的人才，也需要大批应用性人才。然而，我国传统的教育模式主要是培养理论性、研究性的人才。教育界在社会对应用性人才需求的推动下，专门研究了国外应用性人才教育的成功经验，结合国情大力度地改革我国的“高等职业教育”，制定了一系列的方针政策。联合国教科文组织 1997 年公布的教育分类中将这种教育称之为“高等技术与职业教育”，也就是我们通常所说的“高职高专”教育。

我国经济建设需要大批应用性人才，呼唤高职高专教育的崛起和成熟，寄希望于高职高专教育尽快向国家输送高质量的紧缺人才。近几年，高职高专教育发展迅速。目前，各类高职高专学校已占全国高等院校的近 1/2，约有 600 所之多。教育部针对高职高专教育出台的一系列政策和改革方案主要体现在以下几个方面：

- “就业导向”成为高职高专教育的共识。高职高专院校在办学过程中充分考虑市场需求，用“就业导向”的思想制定招生和培养计划。
- 加快“双师型”教师队伍建设。已建立 12 个国家高职高专学生和教师的实训基地。
- 对学生实行“双认证”教育。学历文凭和职业资格“双认证”教育是高职高专教育特色之一。
- 高职高专教育以 2 年学制为主。从学制入手，加快高职高专教学方向的改革，充分办出高职高专教育特色，尽快完成紧缺人才的培养。
- 开展精品专业和精品教材建设。已建立科学的高职高专教育评估体系和评估专家队伍，指导、敦促不同层次、不同类型的学校办出一流的教育。

在教育部关于“高职高专”教育思想和方针指导下，科学出版社积极参与到高职高专教材的建设中去。在组织教材过程中采取了“请进来，走出去”的工作方法，即由教育界的专家、领导和一线的教师，以及企事业从事人力资源工作的人员组成顾问班子，充分分析我国各地区的经济发展、产业结构以及人才需求现状，研究培养国家紧缺人才的关键要素，寻求切实可行的教学方法、手段和途径。

通过研讨认识到，我国幅员辽阔，各地区的产业结构有明显的差异，经济发展也不平衡，各地区对人才的实际需求也有所不同。相应地，对相同专业和相近专业，不同地区的教学单位在培养目标和培养内容上也各有自己的定位。鉴于此，

一线的老师、用人单位交流，掌握了不同地区、不同类型的高职高专院校的教师、学生和教学设施情况，清楚了各学校所设专业的培养目标和办学特点，明确了用人单位的需求条件。各区域编辑对采集的数据进行统计分析，在相互交流的基础上找出各地区、各学校之间的共性和个性，有的放矢地制定选题项目，并进一步向老师、教育管理者征询意见，在获得明确指导性意见后完成“高职高专规划教材”策划及教材的组织工作：

- 第一批“高职高专规划教材”包括三个学科大系：经济管理、信息技术、建筑。
- 第一批“高职高专规划教材”在注意学科建设完整性的同时，十分关注具有区域人才培养特色的教材。
- 第一批“高职高专规划教材”组织过程正值高职高专学制从3年制向2年制转轨，教材编写将其作为考虑因素，要求提示不同学制的讲授内容。
- 第一批“高职高专规划教材”编写强调
 - ◆ 以就业岗位对知识和技能需求下的教材体系的系统性、科学性和实用性。
 - ◆ 教材以实例为先，应用为目的，围绕应用讲理论，取舍适度，不追求理论的完整性。
 - ◆ 提出问题→解决问题→归纳问题的教、学法，培养学生触类旁通的实际工作能力。
 - ◆ 课后作业和练习（或实训）真正具有培养学生实践能力的作用。

在“高职高专规划教材”编委的总体指导下，第一批各科教材基本是由系主任，或从教学一线中遴选的骨干教师执笔撰写。在每本书主编的严格审读及监控下，在各位老师的辛勤编撰下，这套凝聚了所有作者及参与研讨的老师们的经验、智慧和资源，涉及三个大的学科近200种的高职高专教材即将面世。我们希望经过近一年的努力，奉献给读者的这套书是他们渴望已久的适用教材。同时，我们也清醒地认识到，“高职高专”是正在探索中的教育，加之我们的水平和经验有限，教材的选题和编辑出版会存在一些不尽人意的地方，真诚地希望得到老师和学生的批评、建议，以利今后改进，为繁荣我国的高职高专教育不懈努力。

科学出版社

2004年6月1日

前 言

软件工程是一门实践性很强的课程，它是各种开发经验的总结与提炼。软件工程不仅是计算机专业学生十分重要的一门专业课程，也是各类软件开发人员必需掌握的技术。目前，软件工程已从第一代传统的软件工程发展为第二代面向对象的软件工程，如今正向基于软件复用的第三代软件工程发展。为适应信息技术迅速发展和教学的需要，我们特意编写了本书。

本教材的任务是使学生掌握软件工程的基本概念、基本原理、实用的开发方法和技术；了解软件工程各领域的发展动向。在本书的编写过程中，我们根据软件工程的最新发展，着重介绍面向对象的开发方法，并将 UML 建模技术应用到实际系统的分析和设计中。全书共分 13 章，内容包括软件工程概述、软件开发模型、系统分析、系统设计、软件编码与测试、软件维护、配置管理、质量管理、文档技术、软件复用和软件工程环境等。本书语言简练，通俗易懂，采用案例教学方法；注重培养学生的实际开发能力和文档的写作能力，具有很强的实用性，可作为高职高专院校计算机专业或信息类专业的教材，也可作为软件开发人员的参考书。

本书第 1、2、3、4、9 章由余明艳编写，第 5、6 章由杨丽编写，第 7、8、10、11、12、13 章由郭庚麒编写。全书由郭庚麒负责统稿，周江协助书稿的整理并准备了附录。

在本书的编写过程中，我们参阅了大量的资料，在此对所有的编著者表示衷心的感谢。由于编者时间仓促，水平有限，书中不足之处敬请批评指正。

编 者

2004 年 7 月

目 录

第 1 章 软件工程概述	1
1.1 软件	2
1.1.1 软件的定义	2
1.1.2 软件的特点及最新发展	2
1.1.3 软件分类	4
1.2 软件工程概念	5
1.2.1 软件危机	5
1.2.2 软件危机的原因及解决方法	5
1.2.3 软件工程的定义	6
1.2.4 软件工程的原则	6
1.3 软件生存周期与开发模型	7
1.3.1 软件生存周期	7
1.3.2 软件开发模型	9
习题	11
第 2 章 结构化分析	13
2.1 软件定义过程概述	13
2.1.1 问题的定义	13
2.1.2 可行性研究	13
2.1.3 需求分析	14
2.2 结构化分析	16
2.2.1 结构化分析方法概述	16
2.2.2 数据流图	17
2.2.3 数据词典	21
2.2.4 加工逻辑的描述方法	23
2.3 需求规格说明与评审	24
2.3.1 需求规格说明的主要内容	24
2.3.2 需求分析的评审	25
2.4 需求规格说明书写作范例	26
习题	28
第 3 章 概要设计	30
3.1 概要设计的任务与步骤	30
3.1.1 概要设计的任务	30
3.1.2 概要设计的过程	31
3.2 软件设计的概念与原则	32

3.2.1	模块化	32
3.2.2	抽象与逐步求精	32
3.2.3	信息隐蔽和局部化	33
3.2.4	模块独立性	33
3.2.5	设计准则	35
3.3	面向数据流的设计方法	37
3.3.1	基本概念	37
3.3.2	系统结构图	38
3.3.3	变换分析	39
3.3.4	事务分析	41
3.3.5	设计优化	42
3.4	概要设计文档评审	43
3.5	概要设计文档写作范例	43
	习题	45
第4章	详细设计	47
4.1	详细设计的任务与原则	47
4.1.1	详细设计的任务	47
4.1.2	详细设计的原则	48
4.2	详细设计的方法	48
4.2.1	程序流程图	48
4.2.2	N-S图	50
4.2.3	PAD图	50
4.2.4	PDL	52
4.3	详细设计规格说明书与评审	53
4.3.1	详细设计说明书	53
4.3.2	详细设计复审	53
4.4	详细设计文档写作范例	54
	习题	55
第5章	面向对象的分析和设计	57
5.1	面向对象方法概述	57
5.1.1	面向对象方法概述	58
5.1.2	面向对象的软件工程	59
5.1.3	面向对象的基本概念和特征	62
5.1.4	面向对象的开发方法	67
5.2	面向对象分析	70
5.2.1	OOA方法概述	70
5.2.2	确定对象	74
5.2.3	确立结构	75
5.2.4	确立主题	76

5.2.5 确定属性	76
5.2.6 确定服务和消息	77
5.3 面向对象设计	78
5.3.1 OOD 方法概述	78
5.3.2 人机交互子系统的设计	78
5.3.3 问题域子系统的设计	79
5.3.4 任务管理部分的设计	81
5.3.5 数据管理部分的设计	82
5.3.6 OOD 的基本准则	82
习题	83
第 6 章 统一建模语言 UML	84
6.1 UML 概述	84
6.1.1 UML 概念	84
6.1.2 UML 的发展过程	85
6.1.3 UML 的主要内容	85
6.1.4 标准建模语言 UML 的应用领域	87
6.2 UML 的概念模型	88
6.2.1 UML 的构造块	88
6.2.2 UML 的规则	91
6.2.3 UML 中的公共机制	92
6.3 UML 的静态建模机制	92
6.3.1 用例图	92
6.3.2 类图	96
6.3.3 对象图	101
6.3.4 包	101
6.3.5 使用类图的几个建议	102
6.4 UML 的动态建模机制	103
6.4.1 对象之间的交互	103
6.4.2 状态图	104
6.4.3 交互图	107
6.4.4 活动图	109
6.4.5 四种图的运用	111
习题	112
第 7 章 面向对象分析与设计范例	113
7.1 系统需求获取	114
7.1.1 案例背景介绍	114
7.1.2 需求获取	114
7.2 分析问题领域	116
7.2.1 高层用例识别	116

7.2.2 用例分析	118
7.3 静态建模	121
7.3.1 确定对象	121
7.3.2 描述对象的属性	122
7.3.3 描述对象之间的关联	123
7.4 动态建模	125
7.4.1 动态用例分析	125
7.4.2 对象状态分析	128
7.4.3 描述对象操作	130
7.5 系统设计	131
7.5.1 系统总体设计	132
7.5.2 系统详细设计	135
习题	137
第8章 程序编码	139
8.1 程序设计语言	139
8.1.1 程序设计语言分类	139
8.1.2 程序设计语言的特点	141
8.1.3 程序设计语言的选择	143
8.2 编码风格	144
8.3 程序调试	146
8.3.1 调试过程	146
8.3.2 调试途径	147
8.4 面向对象程序设计	148
8.4.1 面向对象语言的一些技术特点	148
8.4.2 面向对象语言的选择	151
习题	152
第9章 软件测试	153
9.1 软件测试	153
9.1.1 软件测试概述	153
9.1.2 测试方法	155
9.1.3 测试过程	156
9.2 测试用例设计	160
9.2.1 白盒子法测试用例设计	160
9.2.2 黑盒子法测试用例设计	163
9.3 程序调试	164
9.4 面向对象测试	165
习题	166
第10章 软件维护	167
10.1 软件维护的内容	167

10.2	软件维护的特点	169
10.2.1	非结构化维护和结构化维护	169
10.2.2	维护的困难性	170
10.2.3	软件维护的费用	170
10.3	软件维护的实施	171
10.3.1	维护的组织	171
10.3.2	维护的流程	171
10.3.3	维护技术	173
10.4	维护的副作用	174
10.5	软件可维护性	175
10.5.1	影响可维护性的软件属性	175
10.5.2	对可维护性的定量度量	176
10.5.3	提高可维护性的途径	176
10.6	软件再工程	177
	习题	179
第 11 章	配置管理与文档技术	180
11.1	软件配置管理基本概念	180
11.1.1	软件配置管理的概念	180
11.1.2	软件配置项	181
11.1.3	基线技术	182
11.2	软件配置管理的任务	183
11.2.1	配置标识	184
11.2.2	版本控制	185
11.2.3	变更控制	186
11.2.4	配置状态报告	188
11.2.5	配置审核	189
11.3	软件文档技术	189
11.3.1	软件文档的作用和分类	190
11.3.2	对文档编制的质量要求	194
11.3.3	文档的管理和维护	195
	习题	196
第 12 章	软件质量管理	197
12.1	软件质量的定义	197
12.1.1	软件质量定义	197
12.1.2	影响软件质量的因素	198
12.2	软件质量保证	199
12.2.1	软件质量保证策略	199
12.2.2	软件质量保证活动	200
12.3	软件质量保证标准	201

12.3.1	ISO 9000 基本思想	201
12.3.2	ISO 9000-3 标准	202
12.4	软件过程成熟度模型 (CMM)	205
12.4.1	软件机构的成熟性	205
12.4.2	软件过程成熟度模型	205
12.4.3	能力成熟度等级	207
12.4.4	关键过程领域	208
12.4.5	成熟度提问单	209
12.4.6	CMM 的应用	210
	习题	211
第 13 章	软件复用和 CASE 技术	212
13.1	软件复用概述	212
13.1.1	软件复用的概念	212
13.1.2	软件复用的种类、成分和意义	213
13.2	面向对象与软件复用	215
13.3	计算机辅助软件工程 (CASE)	216
13.3.1	计算机辅助软件工程概述	216
13.3.2	集成化 CASE 环境 (I-CASE)	217
	习题	219
附录	计算机软件开发文档编写指南	220
附录 1	可行性研究报告	220
附录 2	项目开发计划	224
附录 3	需求规格说明书	225
附录 4	概要设计说明书	227
附录 5	详细设计说明书	229
附录 6	测试计划	230
附录 7	测试分析报告	232
附录 8	开发进度月报	233
附录 9	项目开发总结报告	233
主要参考文献	235

第 1 章 软件工程概述



知识点

- 软件的定义、特点及分类
- 软件危机的概念
- 软件工程的定义、目标和原则
- 软件生存周期及软件开发模型



难点

- 软件危机产生的原因
- 软件工程的定义、目标和原则
- 软件生存周期各阶段应解决的问题及相应的输出



要求

掌握：

- 软件工程的定义、目标和原则
- 常见的软件开发方法
- 软件生存周期各个阶段应解决的问题和相应的输出

了解：

- 软件的定义、特点及分类
- 软件危机产生的原因及解决的途径
- 几类常见的软件开发模型

计算机系统的发展与微电子技术的进步息息相关。自 1946 年计算机诞生以来，计算机系统已经经历了电子管、晶体管、集成电路、大规模集成电路、超大规模集成电路等多个不同的发展时期。由于微电子学技术的进步，计算机硬件性能/价格比平均每十年提高二个数量级，而且质量稳步提高。与此同时，正在运行使用着的计算机软件的数量正以惊人的速度急剧膨胀，计算机软件成本在逐年上升，而且质量没有可靠的保证，软件开发的生产率也远远跟不上计算机应用的要求，软件已经成为限制计算机系统发展的关键因素。

在计算机系统发展的过程中，早期所形成的一些错误概念和做法曾严重地阻碍了计算机软件的开发。用错误方法开发出来的许多大型软件由于无法维护只好提前报废，造成大量人力、物力的浪费。西方计算机科学家把在软件开发和维护过程中遇到的一系列严重问题统称为“软件危机”。在 20 世纪 60 年代后期人们开始认真研究解决软件危机的方法，从而形成了计算机科学技术领域中一门新兴的学科——计算机软件工程学，

通常简称为软件工程。在以计算机为核心的信息化社会中，信息的获取，处理、交流和决策无一不需要大量高质量的软件，因此软件工程一直是人们研究的焦点。

1.1 软 件

1.1.1 软件的定义

软件是计算机系统中与硬件相互依存的部分，它是包括程序、数据及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；数据是程序所处理信息的数据结构；文档是与程序开发、维护和使用相关的各种图文资料。

1.1.2 软件的特点及最新发展

为全面、正确地理解计算机系统及其软件，我们必须了解软件的以下特点。

1. 抽象性

软件是一种逻辑实体，而不是具体的物理实体。这种抽象性是软件与硬件的根本的区别。软件一般寄生在诸如纸、内存储器、磁带、磁盘或光盘等载体上，我们无法观察到它的具体形态，而必须通过对它的运行来分析了解它的功能和特征。

2. 无明显的制造过程

软件的生产与其他硬件的生产不同，它没有明显的制造过程。在硬件的制造过程中必须对每一个制造环节都进行质量控制，以保证整个硬件产品的质量，并且每一个硬件都几乎付出与样品一样的生产资料成本。而软件是将人类的知识和技术转化成产品，软件产品的开发成本几乎全部用在样品的开发设计上，其制造过程则非常简单，人们可以用很低的成本进行软件产品的复制，因此也产生了软件产品的保护问题。软件产品保护这个问题已引起国际上的普遍重视，为了保护软件开发者的根本利益，除国家在法律上采取有力的措施之外，开发者在技术上也采取了各种措施，防止对软件产品的随意复制。

3. 无磨损、老化的问题

在软件的运行和使用期间，没有像硬件那样的磨损、老化问题。任何机械、电子设备在运行和使用的过程中，其失效率大致遵循如图 1.1（硬件失效率曲线）中所示的 U 型曲线（即浴盆曲线）。软件的情况则与此不同，它不存在磨损和老化的问题，然而它却存在退化的问题，设计人员必须不断地修改软件，如图 1.2（软件失效率曲线）所示。

4. 对硬件系统的依赖性

软件的开发和运行往往受到计算机系统的限制，对计算机系统有着不同程度的依赖性。为了减少这种依赖性，在软件开发中提出了软件的可移植问题。

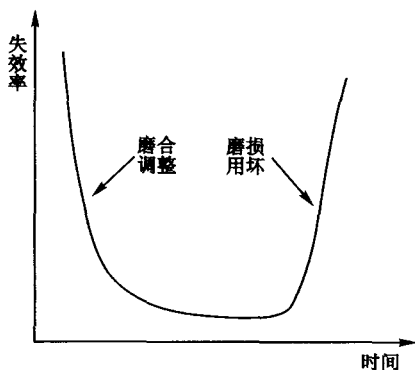


图 1.1 硬件失效率曲线

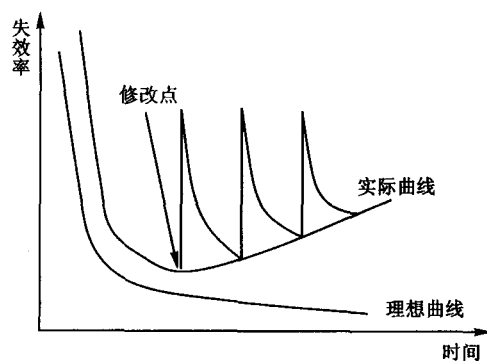


图 1.2 软件失效率曲线

5. 软件开发尚未完全摆脱手工艺的方式

6. 复杂性

软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性，也可能来自程序逻辑结构的复杂性。

7. 成本昂贵

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，它投入的成本是较高的。

8. 社会性

相当多的软件工作涉及到各种社会因素，许多软件的开发和运行涉及机构设置、体制运作及管理方式等问题，甚至涉及到人们的观念和和心理，这些因素直接影响到软件项目的成败。

自 20 世纪 40 年代世界上第一台计算机问世以来，软件经历了程序设计、程序系统及软件工程三个阶段的发展。程序设计阶段（20 世纪 50 年代至 60 年代）的软件指的是程序，程序的开发采用个体工作的方式，开发工作主要依赖于开发人员的个人技能和程序设计的技巧，软件的质量得不到保证，缺乏与程序有关的文档。程序系统阶段（20 世纪 60 年代至 70 年代）的软件是指程序和说明书，软件开发采用开发小组工作的方式，开发工作主要依赖于开发小组的水平，文档资料的不齐全给软件维护带来了很大的难度，软件技术的发展不能满足需要。软件工程阶段（20 世纪 70 年代以后）的软件则是指程序、文档、数据，由开发小组及大中型软件开发机构承担开发软件的任务，开发工作主要依赖于整个机构的管理水平，采用多种开发技术。

目前在很多的应用领域，人们开始采用面向对象的软件开发技术。专家系统、人工智能软件开始走向实际应用。软件技术呈现国际化、网络化、服务化等多种发展趋势。互联网作为二十世纪最重要的科技成果之一，给人类生活 and 经济发展都带来了深远的影响，它所展现出的勃勃商机，吸引了众多厂商围绕互联网开发软件，与分布计算、网络

和互联网相关的软件技术成为软件领域的主要技术热点。此外，自由软件潮流、智能化、简易化、多样化等趋势正极大地拓展软件产业的发展空间，派生出许多具有成长潜力的新兴领域。

1.1.3 软件的分类

20世纪40年代以来，尽管人们开发了大量的软件，积累了丰富的软件资源并使之广泛应用于各个领域，但软件的品种、质量和价格方面仍然满足不了人们日益增长的需要。随着软件复杂性和交互性的增加，我们难以对目前应用着的软件进行一个标准化的分类，这里只是简单地介绍计算机软件在计算机系统、实时系统、商业管理、科学和工程计算、嵌入式系统、人工智能等方面的应用。

1. 系统软件

系统软件是指能与计算机硬件系统紧密配合，使计算机系统的各个部件、相关软件和数据协调高效地工作的软件。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、编译程序、数据库管理程序、文件编辑系统、系统检查与诊断软件等。

2. 实时软件

监视、分析和控制现实世界中发生的事件，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作出反应的软件，称之为实时软件。实时软件包括四个组成部分：数据采集器（负责从外部环境中获取和格式化信息）、分析器（负责将信息转换成应用所需要的形式）、输出/控件器（负责响应外部环境）、管理器（负责协调系统各个部件工作，使系统能保持在一个可接受的响应时间内给实时响应）。实时系统必须在严格的时间范围内响应，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。

3. 商业软件

商业软件可以访问一个或多个商业信息的大型数据库，将已有的数据重新构造，变换成一种能够辅助商业操作和管理决策的形式。商业信息处理是当今最大的软件应用领域，典型的商业软件有银行储蓄软件、电子商务软件、仓库管理软件、ERP软件等。

4. 科学和工程计算软件

科学和工程计算软件的特征是“数值分析算法”，它主要用于科学和工程的计算。如天气预报、弹道计算、石油勘探、地震数据处理、计算机系统仿真、计算机辅助设计等。

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、