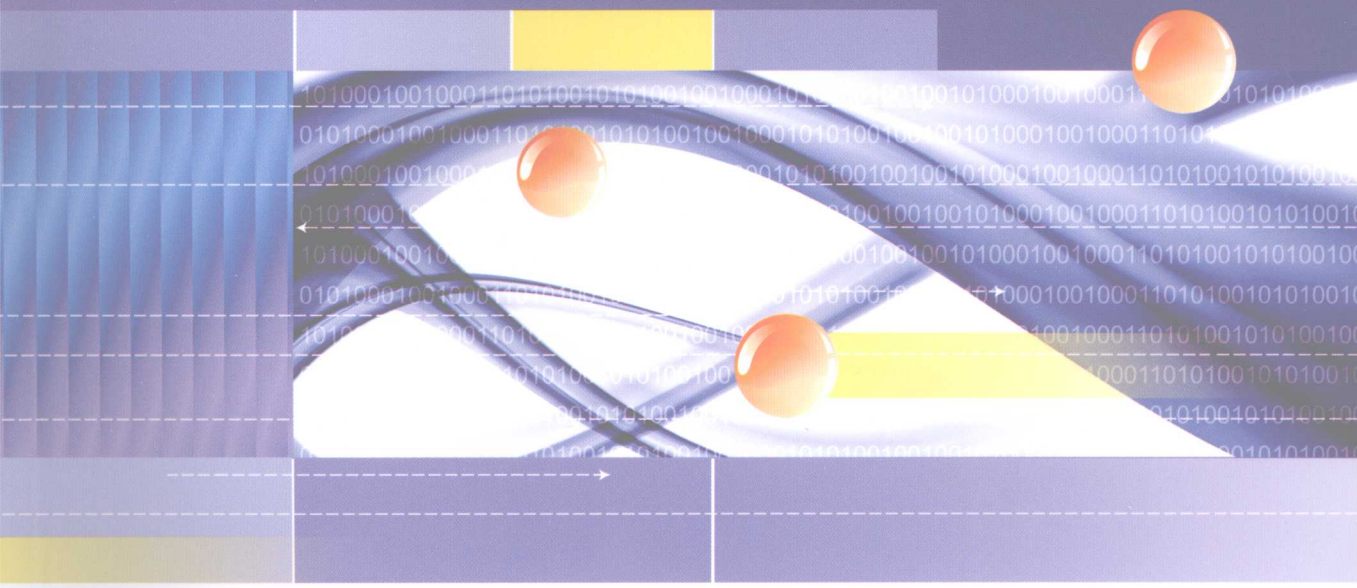
软　件　工　程　系　列　规　划　教　材

面向对象技术与工具

◎ 陈文宇　白忠建　吴　劲　屈　鸿　编著

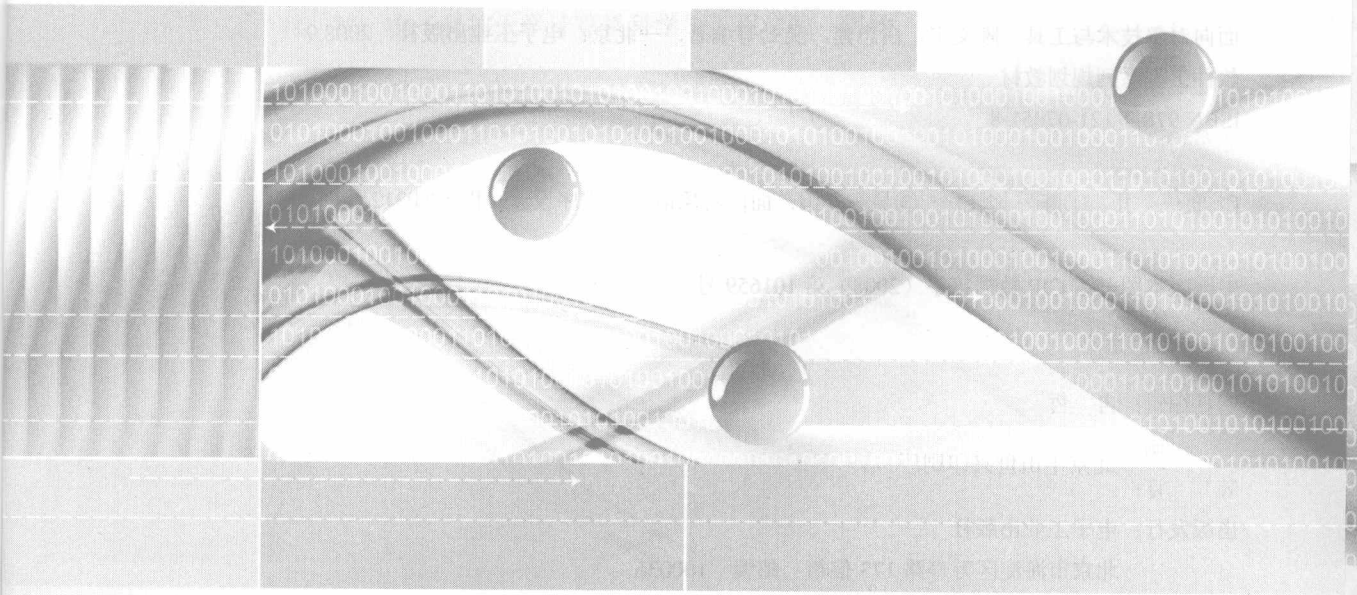


電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

软 件 工 程 系 列 规 划 教 材

面向对象技术与工具

◎ 陈文宇 白忠建 吴劲 屈鸿 编著



電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书内容涉及面向对象的思想、方法和技术及两门著名的面向对象程序设计语言——C++语言和 Java 语言。

全书分为两大部分，共 12 章。第一部分介绍面向对象技术，包括：软件工程概述、软件维护、软件工具与集成化环境、面向对象方法、统一建模语言 UML、软件测试；第二部分介绍面向对象程序设计语言，包括：面向对象程序设计语言的核心概念、C++语言实现数据封装、C++语言实现多态性、C++语言实现继承性、Java 语言基础、Java 语言程序设计。

本书是在汲取了国内外有关教材精华的基础上，并结合编者多年面向对象技术和面向对象语言教学经验而编写的，内容注重科学性、先进性、强调实用性。

本书是高等学校软件工程、计算机等相关专业研究生和高年级本科生的教材，也可作为广大工程技术人员和科研人员的参考书。

图书在版编目（CIP）数据

面向对象技术与工具 / 陈文宇，白忠建，吴劲等编著. —北京：电子工业出版社，2008.9

软件工程系列规划教材

ISBN 978-7-121-07051-8

Ⅰ. 面… Ⅱ. ①陈… ②白… ③吴… Ⅲ. 面向对象语言－程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2008）第 101659 号

责任编辑：冉 哲

印 刷：
装 订：北京牛山世兴印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：33 字数：840 千字

印 次：2008 年 9 月第 1 次印刷

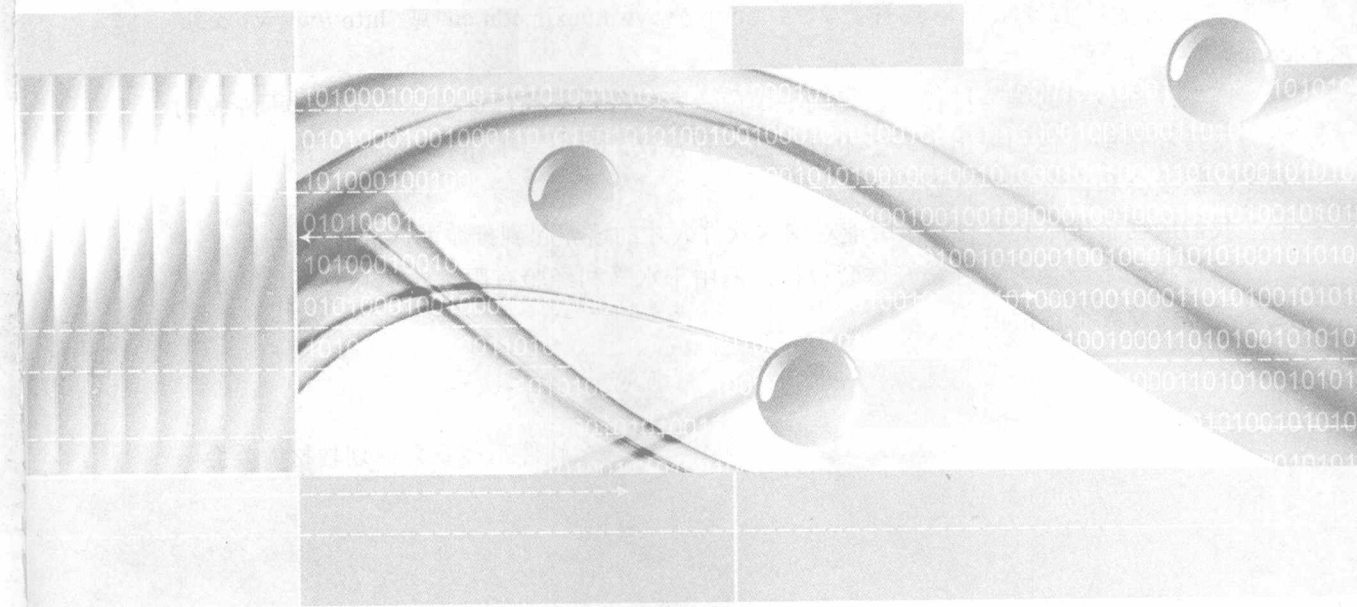
印 数：4000 册 定价：56.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

软件工程系列规划教材　编委会



出版说明

为适应我国经济结构战略性调整的要求和软件产业发展对人才的迫切需要，实现我国软件人才培养的跨越式发展，教育部和国家发展计划委员会于2001年联合批准在国内部分高等学校开办示范性软件学院，努力造就一批进入国际前沿、掌握关键技术、擅长顶层设计的技术带头人，培养一批具备不同专业背景且有市场观念的开发管理、工程管理和软件经营等复合型软件人才，形成一支有相当规模和质量、从事软件开发与应用的专业技术人员队伍。

经过多年的软件人才培养和教学实践，依据国内外企业对软件人才的知识和能力需求，以培养高层次、实用型软件人才为目标，我们组织长期从事软件工程硕士教学的专家教授，编写了一套软件工程系列规划教材。该系列教材主要包括《软件开发技术》、《数据库系统及应用》、《操作系统原理与Linux实例设计》、《面向对象技术与工具》及《计算机网络》。本次推出的软件工程系列规划教材内容涵盖软件工程硕士需要掌握的主要知识和基本技术，具有领域宽、实用型强的特点，既可以作为软件学院工程硕士专业基础课教材，也可作为计算机专业高年级本科生和研究生的教材，还可供软件开发和管理人员作为参考书籍。我们还将陆续推出系列教材的习题解答和上机指导及教学用多媒体电子课件，便于教师备课和学生自学，请登录华信教育资源网（http://www.huaxin.edu.cn 或 http://www.hxedu.com.cn）注册下载。

在本系列教材的编写过程中，得到了国内众多软件学院的任课教师和软件工程专家的大力支持和帮助，他们提出了许多中肯的意见和建议，对编写工作起到很大的指导作用，对此，编委会和作者表示衷心的感谢！

我们殷切希望本套教材的出版能对国内软件人才的培养起到推波助澜的作用。

尽管我们希望全力以赴编好这套教材，但由于水平和经验有限，难免存在不足和疏漏之处，恳请读者不吝指正。

软件工程系列规划教材编委会

前言

从20世纪80年代中开始，以Smalltalk为代表的面向对象的程序设计语言相继推出，面向对象的方法与技术得到发展，到90年代，研究的重点从程序设计语言逐渐转移到面向对象的分析与设计，演化为一种完整的软件开发方法和系统的技术体系。与此同时，出现了许多面向对象的开发方法的流派，面向对象的方法逐渐成为软件开发的主流。

面向对象的软件开发方法（Object-Oriented Software Development，OOSD）是20世纪80年代推出的一种全新的软件开发方法。它非常实用而强有力，被誉为90年代软件的核心技术之一。

其基本思想是：对问题领域进行自然分割，以更接近人类通常思维的方式建立问题领域的模型，以便对客观的信息实体进行结构和行为的模拟，从而使设计的软件更直接地表现问题的求解过程。

面向对象的软件开发方法以对象作为最基本的元素，是分析和解决问题的核心。

用面向对象方法开发的软件，其结构基于客观世界的对象结构，因此与传统的软件相比，软件本身的内容结构发生了质的变化，因而复用性和扩充性都得到了提高，而且能支持需求的变化。

全书分为两大部分，共12章。

第一部分介绍面向对象技术，包括：软件工程概述、软件维护、软件工具与集成化环境、面向对象方法、统一建模语言UML、软件测试；第二部分介绍面向对象程序设计语言，包括：面向对象程序设计语言的核心概念、C++语言实现数据封装、C++语言实现多态性、C++语言实现继承性、Java语言基础、Java语言程序设计。

第1～6章由白忠建编写，第7～10章由陈文宇和屈鸿编写，第11、12章由吴劲编写。

电子科技大学许家珆教授为本教材提供了大量素材并授权使用，在此表示衷心的感谢！

另外，谨对参阅文献的作者和翻译人员表示衷心感谢。

本书在电子科技大学计算机科学与工程学院的徐谡书记和秦志光院长的规划下得以编写和出版。

感谢电子工业出版社的王羽佳编辑，为本书的出版做了大量工作，才使得本书得以同广大读者见面，还要感谢电子科技大学计算机科学与工程学院学生创新中心的大眼睛工作室的同学们，为本书进行的校对和格式修订。

在本书编写过程中，四川汶川发生8.0级特大地震，谨以此书告慰逝者，激励生者。

由于作者水平有限，疏漏、欠妥、谬误之处在所难免，敬请读者指正。

编者于电子科技大学

2008年6月

目 录

第 1 章 软件工程概述 …… (1)

1.1 软件工程的产生和发展 …… (1)

1.1.1 软件工程的发展过程 …… (1)

1.1.2 软件危机 …… (2)

1.1.3 软件工程研究的内容 …… (3)

1.2 软件与软件工程过程 …… (4)

1.2.1 软件的概念和特点 …… (4)

1.2.2 软件工程过程 …… (6)

1.3 软件过程模型 …… (6)

1.3.1 瀑布模型 …… (7)

1.3.2 增量模型 …… (7)

1.3.3 螺旋模型 …… (8)

1.3.4 喷泉模型 …… (9)

1.3.5 智能模型 …… (9)

1.4 软件开发方法 …… (10)

1.4.1 结构化开发方法 …… (10)

1.4.2 原型化开发方法 …… (11)

1.4.3 面向对象的软件开发方法 …… (12)

1.5 软件工具与软件开发环境 …… (13)

习题 1 …… (14)

第 2 章 软件维护 …… (15)

2.1 软件维护的基本概念 …… (15)

2.1.1 软件维护的目的 …… (15)

2.1.2 软件维护的类型 …… (15)

2.1.3 软件维护的特性 …… (17)

2.1.4 软件维护的代价 …… (18)

2.2 软件维护的过程 …… (19)

2.3 软件维护技术 …… (21)

2.4 软件可维护性 …… (22)

2.4.1 软件可维护性的定义 …… (22)

2.4.2 提高可维护性的方法 …… (24)

2.5 逆向工程和再工程 …… (28)

习题 2 …… (30)

第 3 章　软件工具与集成化环境 …… (31)
3.1　软件工具 …… (31)
3.1.1　软件开发工具 …… (31)
3.1.2　软件维护工具 …… (34)
3.1.3　软件管理与支持工具 …… (35)
3.2　集成化 CASE 环境 …… (36)
3.2.1　概述 …… (36)
3.2.2　集成化的 CASE 开发环境的要求 …… (39)
3.2.3　集成化的 CASE 开发环境的体系结构 …… (40)
3.3　软件开发工具 Rational Rose …… (45)
3.3.1　Rose 工具简介 …… (45)
3.3.2　业务用例图 …… (46)
3.3.3　用例图 …… (47)
3.3.4　类图 …… (48)
3.3.5　协作图与时序图 …… (49)
3.3.6　活动图 …… (50)
3.3.7　状态图 …… (51)
3.3.8　构件图和部署图 …… (52)
习题 3 …… (53)
第 4 章　面向对象方法 …… (54)
4.1　面向对象方法概述 …… (54)
4.1.1　什么是面向对象方法 …… (54)
4.1.2　面向对象方法的主要特点 …… (55)
4.2　面向对象的基本概念 …… (56)
4.2.1　对象与类 …… (56)
4.2.2　继承 …… (57)
4.2.3　多态性 …… (58)
4.2.4　消息 …… (58)
4.3　面向对象的分析 …… (58)
4.3.1　需求分析中的问题 …… (59)
4.3.2　OOA 的特点 …… (60)
4.3.3　OOA 的基本任务与分析过程 …… (60)
4.4　面向对象的设计 …… (62)
4.4.1　OOD 的准则 …… (62)
4.4.2　OOD 的基本任务 …… (63)
4.5　典型的面向对象方法 …… (64)
4.5.1　Booch 方法 …… (64)
4.5.2　Coad/Yourdon 方法 …… (66)
4.5.3　对象模型技术 OMT …… (69)

4.5.4 OOSE 方法 …… (75)
习题 4 …… (76)
第 5 章 统一建模语言（UML） …… (77)
5.1 UML 概述 …… (77)
5.1.1 UML 的形成 …… (77)
5.1.2 UML 的特点 …… (78)
5.1.3 UML 建模及其构成 …… (78)
5.1.4 UML 的图形表示 …… (80)
5.1.5 通用模型元素 …… (81)
5.2 建立用例模型 …… (83)
5.2.1 需求分析与用例建模 …… (84)
5.2.2 确定执行者 …… (85)
5.2.3 确定用例 …… (86)
5.2.4 建立用例之间的关系 …… (88)
5.2.5 用例建模实例 …… (89)
5.3 建立静态模型 …… (92)
5.3.1 类图 …… (92)
5.3.2 包图 …… (102)
5.4 建立动态模型 …… (103)
5.4.1 消息 …… (103)
5.4.2 状态图 …… (104)
5.4.3 顺序图 …… (107)
5.4.4 合作图 …… (109)
5.4.5 活动图 …… (111)
5.5 建立实现模型 …… (114)
5.5.1 构件图 …… (114)
5.5.2 配置图 …… (115)
5.6 统一过程及其应用 …… (117)
5.6.1 UML 与 RUP …… (117)
5.6.2 RUP 的特点 …… (118)
5.6.3 RUP 的二维开发模型 …… (119)
5.6.4 RUP 的迭代开发模式 …… (121)
习题 5 …… (122)
第 6 章 软件测试 …… (124)
6.1 软件测试概述 …… (124)
6.1.1 软件测试的基本概念 …… (124)
6.1.2 软件测试的特点和基本原则 …… (126)
6.1.3 软件测试过程 …… (129)
6.1.4 静态分析与动态测试 …… (131)

6.2 软件测试的策略 …… (133)
6.2.1 单元测试 …… (133)
6.2.2 集成测试 …… (136)
6.2.3 确认测试 …… (138)
6.2.4 系统测试 …… (139)
6.2.5 α测试和β测试 …… (140)
6.2.6 综合测试策略 …… (141)
6.3 软件调试 …… (141)
6.3.1 软件调试过程 …… (142)
6.3.2 软件调试策略 …… (142)
6.4 面向对象的测试 …… (144)
6.4.1 面向对象测试的特点 …… (145)
6.4.2 面向对象测试的类型 …… (146)
6.4.3 分析模型测试 …… (148)
6.4.4 面向对象的测试用例 …… (153)
习题 6 …… (154)
第 7 章 面向对象程序设计语言的核心概念 …… (155)
7.1 面向对象的目标 …… (155)
7.2 面向对象的核心概念 …… (157)
7.2.1 数据封装 …… (157)
7.2.2 继承 …… (158)
7.2.3 多态性 …… (159)
7.3 按对象方式思维 …… (161)
7.4 面向对象的思想和方法 …… (163)
7.4.1 面向对象是一种认知方法学 …… (163)
7.4.2 面向对象与软件 IC …… (163)
7.4.3 面向对象方法与结构化程序设计方法 …… (166)
7.4.4 对象是抽象数据类型的实现 …… (167)
7.5 面向对象的程序设计语言 …… (168)
第 8 章 C++语言实现数据封装 …… (173)
8.1 类的定义 …… (173)
8.2 类的成员 …… (175)
8.2.1 数据成员 …… (175)
8.2.2 成员函数 …… (176)
8.2.3 静态成员 …… (178)
8.2.4 类外访问成员的方法 …… (182)
8.3 C++语言的类 …… (182)
8.4 数据封装和信息隐蔽的意义 …… (183)
8.5 构造函数 …… (184)

8.5.1 构造函数的作用 …… (184)
8.5.2 构造函数的定义 …… (185)
8.5.3 重载构造函数 …… (187)
8.6 复制构造函数 …… (188)
8.7 析构函数 …… (193)
8.8 对象的创建、释放和初始化 …… (195)
8.9 对象和指针 …… (197)
8.9.1 this 指针 …… (197)
8.9.2 指向类对象的指针 …… (200)
8.9.3 指向类的成员的指针 …… (201)
8.10 友元关系 …… (204)
8.10.1 友元函数 …… (204)
8.10.2 友元类 …… (206)
8.10.3 友元关系的总结 …… (207)
8.11 与类和对象相关的问题 …… (208)
8.11.1 类类型作为参数类型 …… (208)
8.11.2 一个类的对象作为另一个类的成员 …… (210)
8.11.3 临时对象 …… (211)
习题 8 …… (212)

第 9 章 C++语言实现多态性 …… (213)

9.1 重载运算符 …… (213)
9.1.1 运算符重载的语法形式 …… (216)
9.1.2 重载运算符规则 …… (217)
9.1.3 一元运算符和二元运算符 …… (219)
9.1.4 重载“++”和“—”的前缀和后缀方式 …… (227)
9.1.5 重载赋值运算符 …… (232)
9.1.6 重载运算符“()”和“[]” …… (235)
9.1.7 重载输入运算符和输出运算符 …… (240)
9.1.8 指针悬挂问题 …… (242)
9.2 C++语言的类型转换 …… (246)
9.2.1 标准类型转换为类类型 …… (247)
9.2.2 类类型转换函数 …… (249)
9.3 实例——复数类重载运算符 …… (260)
习题 9 …… (264)

第 10 章 C++语言实现继承性 …… (266)

10.1 继承和派生 …… (266)
10.1.1 为什么要使用继承 …… (266)
10.1.2 派生类的声明和继承方式 …… (272)
10.1.3 基类对象的初始化 …… (281)

10.2 多继承 …… (288)
10.2.1 多继承的概念 …… (288)
10.2.2 虚基类 …… (291)
10.3 继承的意义 …… (297)
10.3.1 模块的观点 …… (298)
10.3.2 类型的观点 …… (298)
10.4 虚函数 …… (299)
10.4.1 静态多态性 …… (300)
10.4.2 基类和派生类的指针与对象的关系 …… (301)
10.4.3 虚函数与多态性 …… (303)
10.5 纯虚函数和抽象类 …… (313)
10.6 虚函数实例——Figure 类 …… (314)
10.7 类属 …… (321)
10.7.1 无约束类属机制 …… (321)
10.7.2 约束类属机制 …… (322)
10.8 模板的概念 …… (323)
10.8.1 函数模板与模板函数 …… (323)
10.8.2 类模板与模板类 …… (326)
10.9 实例——一维数组 …… (331)
10.10 堆栈、队列的应用 …… (339)
习题 10 …… (342)
第 11 章 Java 语言基础 …… (344)
11.1 Java 语言的发展历程 …… (344)
11.2 Java 语言的特点 …… (346)
11.2.1 简捷性 …… (346)
11.2.2 面向对象 …… (346)
11.2.3 动态性 …… (348)
11.2.4 安全性 …… (349)
11.2.5 平台无关性和可移植性 …… (349)
11.2.6 高性能 …… (349)
11.2.7 多线程 …… (350)
11.2.8 分布式 …… (350)
11.2.9 健壮性 …… (350)
11.3 Java 语言的开发工具包 …… (351)
11.3.1 JDK 的下载、安装和设置 …… (351)
11.3.2 JDK 的简介 …… (352)
11.4 Java 程序的基本结构 …… (355)
11.5 Java 程序开发实例 …… (356)
11.5.1 一个简单的 Java Application 程序 …… (356)
11.5.2 一个简单的 Java Applet 程序 …… (357)

11.5.3 Java Applet 图形界面的输入/输出 …… (359)
11.5.4 Java Application 图形界面的输入/输出 …… (361)
11.6 Java 符号集 …… (363)
11.7 数据的简单类型 …… (364)
11.8 常量 …… (364)
11.9 变量 …… (365)
11.10 运算符与表达式 …… (366)
11.10.1 赋值运算与类型转换 …… (366)
11.10.2 算术运算符 …… (367)
11.10.3 关系与逻辑运算 …… (369)
11.10.4 位运算 …… (370)
11.10.5 其他运算符 …… (371)
11.10.6 优先级 …… (371)
11.11 流程控制语句 …… (371)
11.11.1 分支语句 …… (372)
11.11.2 循环语句 …… (375)
11.11.3 跳转语句 …… (378)
习题 11 …… (379)
第 12 章 Java 语言程序设计 …… (380)
12.1 Java 的类和对象 …… (380)
12.1.1 系统定义的类 …… (380)
12.1.2 用户程序自定义类 …… (381)
12.1.3 创建对象与定义构造函数 …… (383)
12.1.4 类的修饰符 …… (386)
12.2 域和方法 …… (386)
12.2.1 域 …… (386)
12.2.2 方法 …… (389)
12.3 访问控制符 …… (392)
12.4 继承 …… (393)
12.4.1 派生子类 …… (394)
12.4.2 域的继承与隐藏 …… (396)
12.4.3 null、this 与 super …… (400)
12.5 多态性 …… (402)
12.5.1 方法的继承 …… (402)
12.5.2 覆盖 …… (403)
12.5.3 重载 …… (403)
12.5.4 构造函数的继承与重载 …… (406)
12.6 上转型对象 …… (407)
12.7 接口 …… (408)
12.7.1 接口的声明 …… (409)

12.7.2 接口的实现 …… (410)
12.7.3 接口的回调 …… (410)
12.7.4 接口作为参数 …… (411)
12.8 包 …… (412)
12.8.1 创建包 …… (413)
12.8.2 包的引用 …… (413)
12.9 数组 …… (414)
12.9.1 数组声明 …… (414)
12.9.2 数组元素的引用及初始化 …… (415)
12.10 字符串 …… (417)
12.10.1 String 类 …… (418)
12.10.2 StringBuffer 类 …… (421)
12.10.3 Java Application 命令行参数 …… (423)
12.11 语言基础类库 …… (423)
12.11.1 Object 类 …… (423)
12.11.2 数据类型类 …… (424)
12.11.3 Math 类 …… (425)
12.11.4 System 类 …… (425)
12.12 Applet 类与 Applet 程序 …… (425)
12.12.1 Applet 类 …… (425)
12.12.2 HTML 文件的参数传递 …… (428)
12.13 异常处理 …… (429)
12.13.1 Java 语言异常处理的特点 …… (429)
12.13.2 异常类的层次 …… (433)
12.13.3 抛出异常 …… (435)
12.13.4 异常处理 …… (436)
12.13.5 嵌套的异常处理 …… (437)
12.14 Java 多线程机制 …… (437)
12.14.1 基本概念 …… (437)
12.14.2 多线程实现方法 …… (441)
12.15 输入/输出流类 …… (447)
12.15.1 文件系统 …… (447)
12.15.2 读/写文件 …… (452)
12.15.3 抽象流类 …… (458)
12.15.4 文件输入/输出流类 …… (459)
12.15.5 加强输入/输出流类 …… (462)
12.15.6 其他输入/输出流类 …… (463)
12.15.7 Reader 和 Writer …… (470)
12.16 网络编程 …… (471)
12.16.1 InetAddress 类 …… (471)

12.16.2 Socket 类和 ServerSocket 类 …… (473)
12.16.3 DatagramPacket 类和 DatagramSocket 类 …… (478)
12.16.4 URL 类和 URLConnection 类 …… (481)
12.17 图形用户界面的设计与实现 …… (484)
12.17.1 图形用户界面的构成 …… (484)
12.17.2 布局管理 …… (485)
12.17.3 组件和事件处理 …… (491)
12.17.4 Java Swing 基础 …… (506)
习题 12 …… (508)
参考文献 …… (510)

第 1 章　软件工程概述

1.1　软件工程的产生和发展

软件工程（Software Engineering）是在克服 20 世纪 60 年代末所出现的“软件危机”的过程中逐渐形成与发展的。自 1968 年在北大西洋公约组织（NATO）举行的软件可靠性学术会议上正式提出“软件工程”的概念以来，在这 40 年的时间里，软件工程在理论和实践两方面都取得了长足的进步。

软件工程是一门指导计算机软件系统开发和维护的工程学科，是一门新兴的边缘学科，它涉及计算机科学、工程科学、管理科学、数学等多学科。软件工程的研究范围广，不仅包括软件系统的开发方法和技术、管理技术，还包括软件工具、环境及软件开发的规范。

软件是信息化的核心，国民经济、国防建设、社会发展及人民生活都离不开软件。软件产业关系到国家经济发展和文化安全，体现了国家综合实力，是决定 21 世纪国际竞争地位的战略性产业。因此，大力推广应用软件工程的开发技术及管理技术，提高软件工程的应用水平，对促进我国软件产业与国际接轨，推动软件产业的迅速发展起着十分重要的作用。

1.1.1　软件工程的发展过程

软件工程的产生和发展是与软件的发展过程紧密相关的。自从第一台电子计算机诞生以来，就开始了软件的生产，“软件工程”提出至今，它的发展经历了 4 个重要阶段。

1．第一代软件工程

20 世纪 60 年代末，软件生产主要采用“生产作坊方式”。随着软件需求量、规模及复杂度的迅速增大，生产作坊的方式已不能够适应软件生产的需要，出现了所谓“软件危机”（Software Crisis），其表现为软件生产效率低，大量质量低劣的软件涌入市场或在开发过程中夭折。软件危机不断扩大，对软件生产已经产生了严重危害。

为了克服软件危机，在 NATO 举行的软件可靠性会议上第一次提出“软件工程”这一名词，将软件开发纳入了工程化的轨道，基本形成了软件工程的概念、框架、技术和方法。这阶段又称为**传统的软件工程**。

2．第二代软件工程

从 20 世纪 80 年代中期开始，以 Smalltalk 为代表的面向对象的程序设计语言推出，面向对象的方法与技术得到发展。从 20 世纪 90 年代起，研究的重点从程序设计语言逐渐转移到面向对象的分析与设计，演化为一种完整的软件开发方法和系统的技术体系。与此同时，

出现了许多面向对象的开发方法的流派，面向对象的方法逐渐成为软件开发的主流。所以这一阶段又称为**对象工程**。

3. 第三代软件工程

随着软件规模的不断增大，开发人员也随之增多，开发时间相应持续增长，加之软件是知识密集型的逻辑思维产品，这些都增加了软件工程管理的难度。人们在软件开发的实践过程中认识到：提高软件生产率，保证软件质量的关键是对“软件过程”的控制和管理，是软件开发和维护中的管理和支持能力。提出对软件项目管理的计划、组织、成本估算、质量保证、软件配置管理等技术与策略，逐步形成了**软件过程工程**。

4. 第四代软件工程

从 20 世纪 90 年代起，基于构件（Component）的开发方法取得重要进展，软件系统的开发可通过使用现存的可复用构件组装完成，而无须从头开始构造，从而达到提高效率和质量，降低成本的目的。软件复用技术及构件技术的发展，对克服软件危机提供了一条有效途径。将这一阶段称为**构件工程**。

1.1.2 软件危机

1. 软件危机的产生

“软件危机”（Software Crisis）的出现是由于软件的规模越来越大，复杂度不断增大，而软件需求量也不断增大，“生产作坊式”的软件开发模式及技术已不能满足软件发展的需要。

软件开发过程是一种高密集度的脑力劳动，需要投入大量的人力、物力和财力，如果软件开发的模式及技术不能适应软件发展的需要，将致使大量质量低劣的软件产品涌向市场，有的甚至在开发过程中就夭折了。国外在开发一些大型软件系统时，遇到了许多困难，有的系统最终彻底失败了；有的系统则比原计划推迟了好多年，而且费用大大超过了预算；有的系统不能符合用户当初的期望；有的系统则无法进行修改维护。典型的例子如下。

IBM 公司的 OS/360，共约 100 万条指令，花费了 5000 人年，经费达数亿美元，而结果却令人沮丧，错误多达 2000 个以上，系统根本无法正常运行。OS/360 系统的负责人 Brooks 这样描述开发过程的困难和混乱：“就像野兽在泥潭中做垂死挣扎，挣扎得越猛，陷得越深，最后没有一个野兽能够逃脱淹没在泥潭中的命运……”

1967 年，前苏联“联盟一号”载人宇宙飞船，由于其软件设计时忽略了一个小数点，导致返航时打不开降落伞，当进入大气层时因摩擦力太大而烧毁，造成机毁人亡的巨大损失。

还有，可以称为 20 世纪世界上最精心设计，并花费了巨额投资的美国阿波罗登月飞行计划使用的软件，也仍然没有避免出错，例如，阿波罗 8 号由于太空飞船的一个计算机软件错误，造成存储器的一部分信息丢失；阿波罗 14 号在飞行的 10 天中，出现了 18 个软件错误。

2. 软件危机的表现

20 世纪 60 年代末期所发生的软件危机，反映在软件可靠性没有保障、软件维护工作量