

国家级精品课程配套教材

高等院校信息技术规划教材

计算机软件技术基础

周福才 高克宁 李金双 高 岩 编著

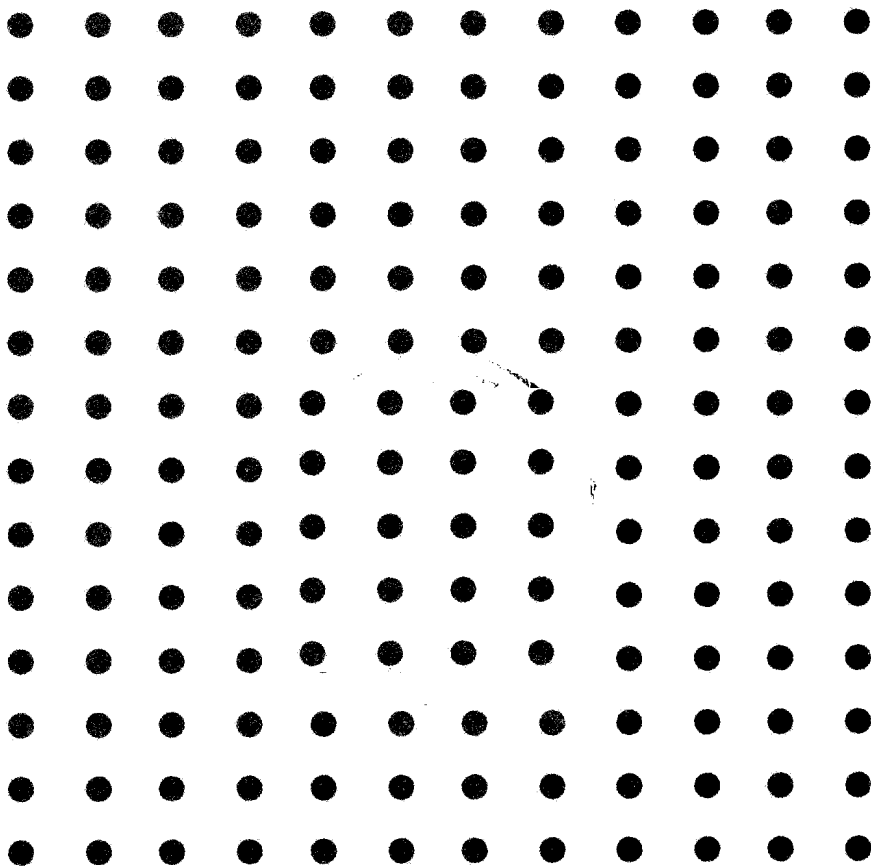


清华大学出版社

国家级精品课程配套教材
高等院校信息技术规划教材

计算机软件技术基础

周福才 高克宁 李金双 高 岩 编著



清华大学出版社
北京

内 容 简 介

本书围绕软件开发所需要的知识,系统地介绍算法与数据结构、数据库技术、操作系统技术、软件设计方法以及个体软件过程管理五方面的内容。本书适用于学习程序设计语言之后,想继续深入地学习软件开发相关技术和方法的读者。并为之配套出版了辅导教材《计算机软件技术基础实验指导》和《计算机软件技术基础习题与解答》。

本书可作为高等院校理工科非计算机专业本科生和研究生教材,也可作为计算机培训教材。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机软件技术基础/周福才等编著. —北京:清华大学出版社,2011.8
(高等院校信息技术规划教材)

ISBN 978-7-302-24391-5

I. ①计… II. ①周… III. ①软件开发—技术 IV. ①TP31

中国版本图书馆 CIP 数据核字(2010)第 259011 号

责任编辑:战晓雷 顾 冰

责任校对:白 蕾

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260

印 张:21.25

字 数:513 千字

版 次:2011 年 8 月第 1 版

印 次:2011 年 8 月第 1 次印刷

印 数:1~3000

定 价:35.00 元

产品编号:032455-01

前 言

针对理工类专业人才计算机软件开发能力培养的需求,作者根据多年教学以及软件开发实践的经验,以算法与数据结构、数据库技术、操作系统技术、软件设计方法和个体软件过程管理等内容为主线,系统地介绍软件开发过程所涉及的基本方法和技术。

本书从软件开发能力培养的实际目标出发,注重技术的实用性和典型性,将内容分为基础篇、方法篇和工程篇,既紧跟软件开发的技术前沿,又兼顾传统的方法和技术。

基础篇由第1章至第4章构成,系统地介绍软件开发所需常用算法与数据结构、数据库基础、进程与线程、文件管理、用户界面设计和数据库开发等实用软件开发技术基础。第1章从程序、软件到系统的视角,介绍了随着软件开发规模的不断扩大,由程序设计上升为软件开发所涉及的各种知识;第2章以程序设计能力培养为目标,介绍了常用的算法与数据结构,包括线性表、堆栈、树、图、查找、排序和递归;第3章从数据存储和处理的角度,介绍了数据库相关技术,实现从文件管理到数据库管理的跨越;第4章从软件开发实践出发,介绍了软件开发过程所要掌握的操作系统及其接口方面的知识。

方法篇由第5章至第7章构成,重点介绍基于UML面向对象的软件开发方法,同时兼顾了传统结构化开发方法。第5章从结构化程序设计入手,系统地介绍结构化软件开发方法;第6章在分析了结构化软件开发方法的不足后,系统地介绍主流的面向对象的软件开发方法,主要包括基于UML的面向对象的软件设计建模;第7章系统地介绍软件工程及项目管理方面的知识。

工程篇由第8章和第9章构成。在简述软件工程基本概念的基础上,介绍了个人软件过程管理PSP和组件技术。第8章从软件开发是一个典型工程项目的高度,介绍软件工程基本概念,并从个体软件过程管理能力培养的角度,重点介绍PSP。第9章从软件开发过程的分工与重用角度,介绍软件开发过程中设计的库和组件技术,重点介绍了支持多平台的CORBA组件技术。

本书作为国家级网络精品课“软件技术基础”的配套教材,由周福才、高克宁、李金双、赵长宽、柳秀梅和高岩共同完成。其中,精品课课程负责人周福才教授负责本书内容的组织、统稿和审校。高克宁编写第1、2章,柳秀梅编写第3章,李金双编写第4、6章,高岩编写第5、7章,赵长宽编写第8、9章。

由于作者水平有限,书中难免会有错误或疏漏之处,真诚地欢迎各位专家和读者批评指正,以帮助我们进一步完善教材。

作 者

2011年4月于东北大学

目 录

第 1 篇 基 础 篇

第 1 章 软件开发概述	3
1.1 程序与算法	3
1.1.1 程序.....	3
1.1.2 程序设计语言.....	4
1.1.3 算法.....	6
1.1.4 算法描述语言.....	7
1.1.5 算法设计目标.....	8
1.2 软件.....	10
1.2.1 软件的基本概念	10
1.2.2 软件分类	10
1.2.3 软件开发历史与发展趋势	11
1.2.4 软件危机	12
1.2.5 软件生存周期	13
1.3 软件开发技术基础.....	14
1.3.1 软件开发技术概述	14
1.3.2 数据结构	14
1.3.3 关系型数据库	16
1.3.4 操作系统接口技术	17
1.4 软件工程.....	17
1.4.1 软件工程方法学	17
1.4.2 软件工程建模	18
1.4.3 软件开发过程管理	19
1.5 软件工程技术基础.....	20
1.5.1 软件复用技术	20
1.5.2 组件技术	20
1.5.3 C/S 系统	21
1.5.4 B/S 系统	23
第 2 章 数据结构及算法	25
2.1 数据结构概述.....	25
2.1.1 基本概念	25

2.1.2	数据结构	26
2.1.3	数据类型与抽象数据类型	28
2.1.4	算法的评价	29
2.2	线性表	32
2.2.1	线性表的逻辑结构	32
2.2.2	顺序表	33
2.2.3	链表	38
2.3	栈和队列	44
2.3.1	栈	44
2.3.2	队列	49
2.4	串与数组	55
2.4.1	串	55
2.4.2	数组和矩阵	60
2.5	树和二叉树	64
2.5.1	树的定义	64
2.5.2	二叉树	67
2.5.3	线索二叉树	72
2.5.4	哈夫曼树	74
2.6	图	78
2.6.1	图的定义	78
2.6.2	图的存储	80
2.6.3	图的遍历	83
2.6.4	图的应用	86
2.7	查找算法	88
2.7.1	基本概念	88
2.7.2	顺序查找	89
2.7.3	折半查找	90
2.7.4	分块查找	91
2.7.5	二叉排序树	93
2.7.6	哈希表查找	96
2.8	排序算法	99
2.8.1	基本概念	99
2.8.2	插入排序	100
2.8.3	选择排序	101
2.8.4	冒泡排序	102
2.8.5	快速排序	104
2.9	递归算法	105
2.9.1	递归的定义	106

2.9.2 递归的应用	107
第3章 数据库管理技术	110
3.1 概述	110
3.1.1 基本概念	110
3.1.2 数据库管理技术发展史	110
3.1.3 关系数据库定义	112
3.1.4 面向对象数据库定义	112
3.1.5 典型商用数据库管理系统	113
3.2 关系数据库规范化理论	115
3.2.1 数据模型	115
3.2.2 规范化理论	119
3.3 关系数据库标准查询语言 SQL	124
3.3.1 数据定义语言 DDL	125
3.3.2 数据操纵语言 DML	128
3.3.3 DCL	132
3.4 数据库设计基本方法	133
3.4.1 需求分析	134
3.4.2 概念结构设计	134
3.4.3 逻辑结构设计	135
3.4.4 物理结构设计	136
3.4.5 数据库的实施及运行维护	136
3.5 数据库保护	137
3.5.1 安全性和完整性	137
3.5.2 并发控制和事务处理	139
3.5.3 数据库备份与恢复	140
第4章 软件开发技术	143
4.1 操作系统概述	143
4.1.1 操作系统定义	143
4.1.2 操作系统的类型	145
4.1.3 典型操作系统	147
4.1.4 操作系统接口开发技术	149
4.2 进程和线程管理	151
4.2.1 进程与线程定义	151
4.2.2 多进程程序开发	151
4.2.3 多线程程序开发	152
4.2.4 进程通信	154
4.3 内存管理技术	155

4.3.1	内存管理概述	155
4.3.2	内存管理函数	156
4.4	文件管理技术	162
4.4.1	文件的定义	162
4.4.2	文件管理函数	162
4.4.3	文件管理程序开发	162
4.5	用户界面设计技术	164
4.5.1	用户界面概念	164
4.5.2	文本界面	164
4.5.3	图形界面的基本要素	165
4.5.4	图形界面的设计原则	165
4.5.5	图形界面开发技术	166
4.6	数据库开发技术	170
4.6.1	SQL 技术	170
4.6.2	ODBC 技术	171
4.6.3	ADO 技术	171
4.6.4	JDBC 和 ORM 技术	171

第 2 篇 方 法 篇

第 5 章	传统的软件开发方法	175
5.1	结构化开发方法概述	175
5.2	可行性研究	175
5.2.1	可行性研究的任务	176
5.2.2	可行性研究的步骤	177
5.2.3	可行性研究报告	178
5.3	需求分析	178
5.3.1	需求分析概述	179
5.3.2	需求分析原则和模型	182
5.3.3	功能建模与数据流程图	183
5.3.4	行为建模与状态变迁图	190
5.3.5	数据字典	191
5.3.6	软件需求说明书	195
5.4	系统设计	196
5.4.1	软件设计概述	197
5.4.2	软件设计原则	197
5.4.3	结构化设计方法	202
5.4.4	软件设计文档	210

5.5	系统测试与维护	212
5.5.1	软件测试概述	212
5.5.2	软件测试方法	214
5.5.3	测试实施过程	214
5.5.4	系统维护	217
第 6 章	面向对象的软件开发方法	220
6.1	面向对象方法概述	220
6.1.1	传统软件开发方法的问题	220
6.1.2	面向对象技术的由来	222
6.1.3	面向对象的基本概念	223
6.2	统一建模语言——UML 概述	225
6.2.1	用例图	226
6.2.2	类图和对象图	228
6.2.3	交互图	231
6.2.4	状态图	233
6.3	面向对象建模	234
6.3.1	系统、模型和视图	234
6.3.2	数据类型、抽象数据类型和实例	236
6.3.3	类、抽象类和对象	237
6.3.4	事件类、事件和消息	238
6.3.5	面向对象建模过程	239
6.4	UML 建模实例	241
6.4.1	问题描述	241
6.4.2	系统建模	241
第 7 章	软件工程	245
7.1	软件工程概述	245
7.1.1	软件工程原理	245
7.1.2	软件工程基本目标	247
7.2	软件开发方法	248
7.2.1	传统软件开发方法	249
7.2.2	现代软件开发方法	249
7.3	软件生存周期	250
7.4	软件开发模型	251
7.4.1	线性模型系列	252
7.4.2	演化模型系列	253
7.4.3	专用模型系列	258
7.4.4	新型模型系列	258

7.5 软件工程管理	262
7.5.1 软件工程项目管理的任务	262
7.5.2 软件人员组织与管理	263
7.5.3 软件配置管理	265

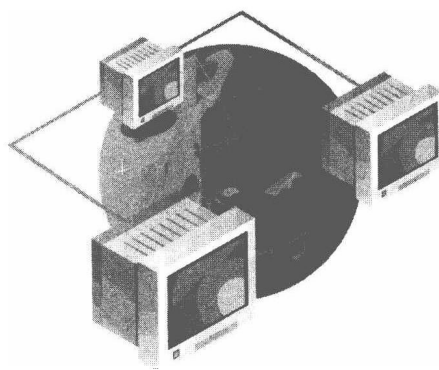
第3篇 工程篇

第8章 个体软件开发过程管理	269
8.1 概述	269
8.2 编码规范定义	270
8.3 ANSI C 程序编码规范	270
8.3.1 代码结构与组织	270
8.3.2 注释	273
8.3.3 标识符命名规范	275
8.3.4 代码风格与排版	277
8.4 软件生命周期模型	278
8.5 CMM 简介	279
8.6 PSP 个体软件开发过程简介	280
8.7 PSP0 级	280
8.7.1 计划过程管理	281
8.7.2 开发过程管理	282
8.7.3 总结过程管理	282
8.7.4 PSP0 过程文档	283
8.7.5 PSP0.1 级	284
8.8 软件开发计划	286
8.8.1 软件开发计划基本内容	286
8.8.2 制定个人软件开发计划	287
8.8.3 PSP 软件开发计划过程	288
8.9 PSP1 级	289
8.9.1 规模估算	289
8.9.2 任务计划	290
8.9.3 进度计划	290
8.10 PSP2 级	291
8.10.1 代码评审	291
8.10.2 设计评审	292
8.10.3 缺陷预防	292
8.10.4 PSP2 改进	293

第 9 章 组件技术	294
9.1 概述	294
9.2 代码重用技术	295
9.2.1 源程序文件.....	295
9.2.2 静态库.....	296
9.2.3 动态链接库.....	298
9.3 组件技术简介	300
9.4 体系结构与组件模型标准	301
9.5 CORBA 技术	302
9.5.1 CORBA 结构基础	302
9.5.2 CORBA 运行机制	303
9.5.3 IDL 约定.....	305
9.5.4 IDL 数据类型.....	306
9.5.5 构建 CORBA 应用程序.....	312
参考文献	325

第1篇

基础篇



软件开发概述

1.1 程序与算法

1.1.1 程序

程序(program)是为实现特定目标或解决特定问题而采用计算机语言编写的、可以连续执行并能够完成一定任务的指令序列的集合。程序具有以下特点。

1. 程序是程序设计语言抽象符号的集合

程序设计语言既有面向机器的汇编语言,又有面向过程和面向对象的高级语言。程序设计的过程就是将待解决的问题转化为程序设计语言的表达式、语句、过程/函数、对象,并进一步编译、链接、执行的过程。

2. 程序是对数据施行算法的过程

使用面向过程程序设计语言提供的符号和语法编写程序,需要按照一定的算法(解题的具体方法)编写程序的执行步骤。同样的数据采用不同的实现算法,所需要的时间和所占据的存储空间的开销大不相同。应根据实际需求,选择最优算法以达到效率和质量上的最佳。

使用面向对象程序设计语言编写程序同样需要算法。面向对象语言提供了更高的抽象层次的对象。从对象的角度而言,程序是对象属性、状态、行为以及对象间关系的描述。其中,数据表示状态和属性,行为方法包含了改变状态的算法。

3. 程序是层次化的调用结构

从计算机系统结构的角度来说,由高到低的程序层次结构为应用软件、工具集、操作系统、计算机硬件。高层程序可以利用低层系统提供的服务为自身服务。例如,应用软件利用操作系统和工具集提供的服务实现应用功能(如字处理软件 Word、VC++ 编程环境等),系统软件工具集利用操作系统提供的服务实现服务功能(如数据库系统等)。当高层程序需要调用低层服务时,高层程序暂停执行继而转向执行低层服务程序,在低层服务程序执行结束后返回其调用层次继续执行。

1.1.2 程序设计语言

程序设计语言实际是编写程序所要遵循的一系列操作规则。按照这些规则,人们编写程序与计算机进行信息交流。因此计算机程序设计语言是人与计算机进行信息交换的语言工具。随着计算机技术的发展,计算机程序设计语言也在不断地发展,形成了功能、特点不同的各类程序设计语言。

1. 程序设计语言的分类

可以从不同的角度对计算机程序设计语言进行分类。从语言的发展历程,可分为机器语言、汇编语言和高级语言。从语言的应用范围,可分为通用语言与专用语言,或细化为系统程序设计语言、科学计算语言、事务处理语言、实时控制语言及解决非确定性问题的语言等。从程序设计的方法,可分为结构化语言、面向对象语言、函数式语言以及逻辑型语言等。

机器语言是由计算机硬件只能识别的 0、1 所组合的二进制机器指令序列组成,是最基本的计算机语言。由于机器指令是特定计算机系统所固有的、面向机器的语言。因此,利用机器语言编写程序需要了解实现程序的计算机体系结构。

对程序设计者而言,机器语言程序难于编写,而且更难以理解、修改和维护。为了提高计算机程序编写的效率,人们采用符号(英文单词)代表机器指令(如用 ADD 代表加法),即出现了汇编语言。汇编语言是机器语言的符号化,它的书写格式在很大程度上仍然依赖于特定的计算机机器指令,不同的计算机在指令长度、寻址方式、寄存器数目、指令表示等方面都不相同,汇编语言同样面向计算机结构。人们将机器语言和汇编语言统称为低级语言。

由于汇编语言不便于进行数学描述,且不可移植,因此人们开发出功能更强、抽象程度更高、面向各类应用的程序设计语言,称为高级语言。

高级语言作为面向计算过程和面向问题的语言,只与解题的实现步骤有关。用高级语言编写的程序称为源程序,源程序不能直接在计算机上运行,需要将高级语言翻译为机器语言,才能在机器上运行。把高级语言程序翻译为机器语言程序有编译和解释两种方式。编译程序(编译器)将源程序翻译成目标程序,然后在机器上运行目标程序;解释程序(解释器)直接解释执行源程序。两种处理方式的区别在于:编译方式下的编译器将源程序翻译成独立的目标程序,机器运行的是与源程序等价的目标程序,源程序和编译程序都不再参与目标程序的执行过程;而在解释方式下,解释器在解释源程序时不生成独立的目标程序,解释程序和源程序都要参与到程序的运行过程中,运行程序的控制权在解释器。

结构化语言是基于操作过程的语言,著名的计算机科学家 N. 沃思曾给出结构化的程序公式:程序=算法+数据结构。数据结构是对数据形式的表示或描述(数据流),即指定程序所使用数据之间的关系和存储方式。算法是针对数据所进行的操作(控制流),操作的步骤即程序设计的算法。该公式说明了结构化程序的两大要素是数据结构和算法,二者相辅相成,缺一不可。没有数据,算法就没有运算处理的对象。第一个结构化语言是

FORTRAN 语言, C 语言是典型的结构化语言, 它体现了结构化程序设计方法的关键思想。

面向对象语言运用对象、类、继承、封装、聚合、消息传递、多态性等概念构造程序系统。将具有相同属性、操作和访问机制的多个对象抽象为一个对象类。对象的实体中封装了属性和操作, 用户可以发送消息查询或修改对象的属性。代表性的面向对象语言有 C++、Java 等。

函数式语言是一类以 λ 演算为基础的语言, 代表语言为 LISP, λ 演算的函数可以接受函数当作输入(引数)和输出(传出值)。在函数式编程语言中强调函数的计算比指令的执行重要, 并且函数的计算可随时调用。

逻辑型语言是一类以形式逻辑为基础的语言, 代表语言为 PROLOG。PROLOG 建立在关系理论和一阶谓词理论的基础上, 是一系列事实、数据对象和事实间关系规则的集合。

2. 程序设计语言的基本元素

程序设计语言的基本成分包括数据、运算、控制、数据的输入/输出和函数(或过程)。

数据是程序操作的对象, 具有存储类型、数据类型、名称、作用域以及生存期等属性。使用数据时要为其分配存储空间, 存储类型说明数据在内存中的位置; 数据类型说明数据占用内存的字节个数以及存放形式; 作用域说明程序可以使用数据的范围大小; 生存期说明数据占用内存的时间长短。根据上述属性, 程序中的数据可分为常量和变量、全局变量和局部变量、动态变量和静态变量。大多数程序设计语言的数据类型包括基本类型、构造类型、用户定义类型或其他类型。

数据运算必须明确运算使用的运算符以及运算规则。为了明确运算结果, 对运算符规定了优先级和结合性, 运算符的使用还与数据类型密切相关。

控制表明程序语句运行的次序关系, 使用控制语句构造程序的流程控制结构。理论上已证明, 任何一个可计算问题的程序都可以由顺序、分支和循环三种基本结构构成。三种基本结构如图 1.1 所示。

顺序结构表示程序执行的序列。程序从第一个操作开始, 按顺序依次执行其后的操作, 直到序列的最后操作为止; 分支结构表示在多种条件(两个或多个)中选择其中一个分支序列执行, 条件成立与否关系到执行不同的语句序列; 循环结构表示在满足循环条件时重复执行一段语句序列。三种基本结构相互嵌套, 可以实现更加复杂的程序。

一个完整的程序由一系列子处理程序段组成(如 C 语言中的函数及其他语言中的过程)。以 C 语言为例, C 程序由一个或多个函数组成, 函数是 C 程序模块的主要成分, 是一段具有独立功能的程序。每个函数都有一个确定的名字, 其中名字为 main 的函数是程序的入口函数。函数使用时要进行函数定义、函数声明及函数调用。

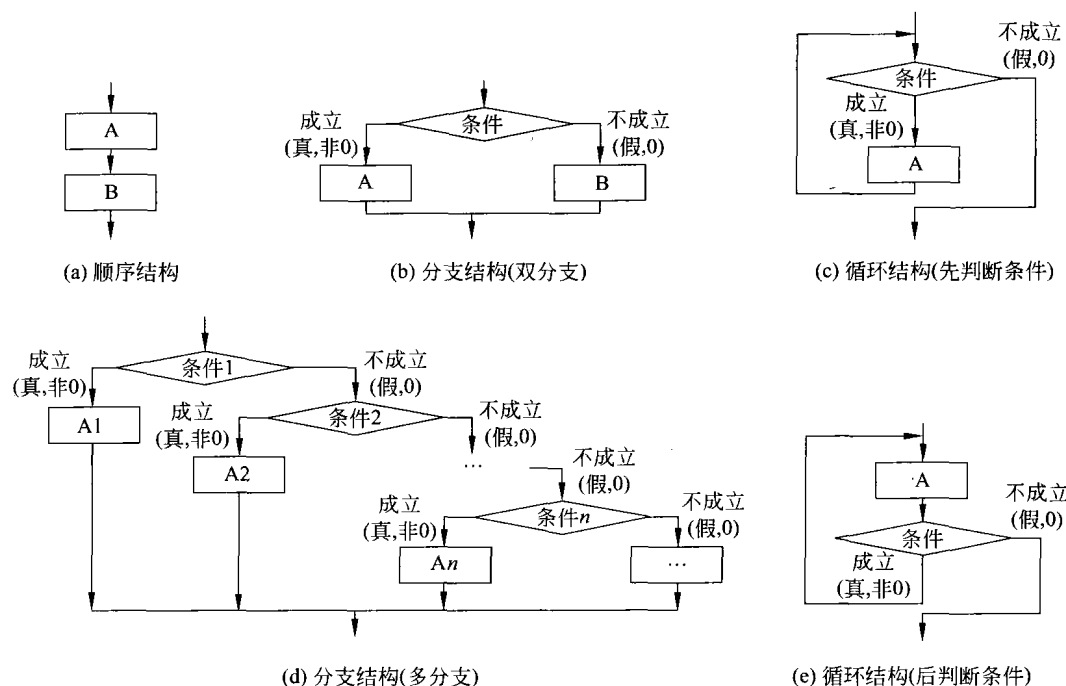


图 1.1 顺序、分支、循环结构示意图

1.1.3 算法

当利用计算机解决一个具体问题时,首先需要从具体问题中抽象出一个适当的数学模型,其次设计一个解决此数学模型的算法(algorithm),最后利用计算机语言编出实现算法的程序,并进行测试、调整,直至得到最终结果。

数学模型的抽象是获得正确结果的基础和保障,寻求数学模型的实质是通过对实际问题的分析,从中提取出操作对象,找出操作对象之间所隐含的关系,并采用数学的语言加以描述。

算法是为了解决一个特定问题而采取的确定的、有限的、按照一定次序进行的、缺一不可的执行步骤。从算法的应用领域,算法分为数值算法和非数值算法。数值算法主要进行数学模型的计算,科学和工程计算方面的算法都属于数值算法,如求解微/积分、代数方程等数值计算问题。非数值算法主要进行比较和逻辑运算,数据处理方面的算法都属于非数值算法,如各种排序、查找、插入、删除、遍历等非数值运算问题。

从算法的实现技术划分,算法可分为递归算法和非递归算法。理论上,任何递归算法都可以通过循环、堆栈等相关技术转化为非递归算法(递推或迭代)。

作为对特定问题处理过程的精确描述,算法应该具备以下特性:

1. 有穷性

有穷性是指解决问题应在“合理的限度之内”,即一个算法应包含有限次的操作步骤,