

高等学校培养应用型人才教材——计算机系列



Java 程序设计教程

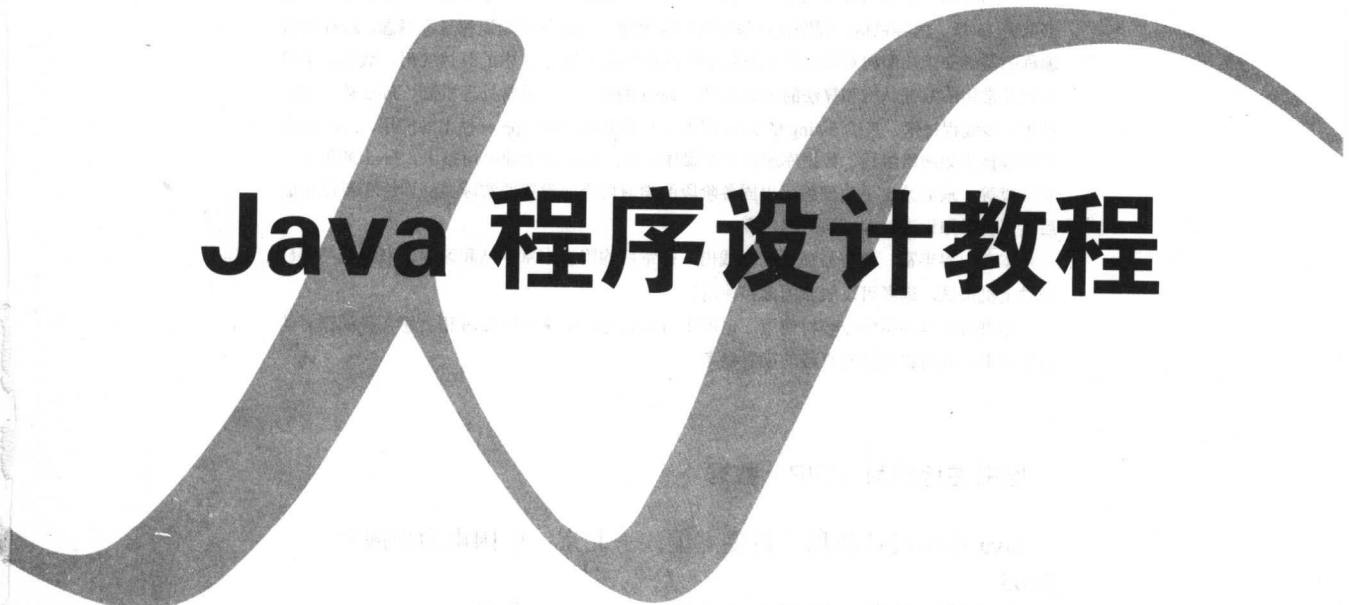
肖 旻 主 编
廖 雷 副主编



中国电力出版社

www.infopower.com.cn

高等学校培养应用型人才教材——计算机系列



Java 程序设计教程

	肖 旻	主 编
	廖 雷	副主编
刘晓璐	杜 青	参 编

中国电力出版社

内 容 提 要

本书为高等学校培养应用型人才教材之一,由具有丰富教学经验的一线教师编写。全书共分15章。内容包括:程序设计与面向对象程序设计的基本知识和基本概念,Java 语言编程的基础知识,面向对象编程中的四个特性及其在Java语言中的具体应用,数组、字符串以及常用的数据结构和算法的Java实现,Java的特色——异常处理机制,Java输入输出技术,多线程技术,基于Swing的Java图形用户界面设计和Applet技术与应用,Java的高级编程技术如网络编程、数据库连接和多媒体应用,Java的企业应用技术,Java的集成开发环境等。最后给出了为理解、巩固各阶段的学习内容而设置的实际技能训练和授课完成后运用所学知识进行综合训练的内容。

本书内容丰富、通俗易懂、深入浅出,每章后均附有本章小结和习题。书中程序都已经过上机调试,读者可以放心阅读和使用。

本书可作为高等学校教材使用,也可作为对面向对象编程和Java语言感兴趣的读者的自学用书,还可供其他有兴趣的读者参考。

图书在版编目(CIP)数据

Java 程序设计教程 / 肖旻主编. —北京: 中国电力出版社,
2003
高等学校培养应用型人才教材——计算机系列
ISBN 7-5083-1522-7

I. J… II. 肖… III. JAVA 语言—程序设计—高等学校
—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 057815 号

中国电力出版社出版、发行
(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)
汇鑫印务有限公司印刷
各地新华书店经售

*

2003 年 9 月第一版 2003 年 9 月北京第一次印刷
787 毫米×1092 毫米 16 开本 23 印张 510 千字
定价 30.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)

高等学校培养应用型人才教材——计算机系列

编 委 会

主任委员:

宗 健 常明华

副主任委员:

顾元刚 陈 雁 杨翠南 林全新 华容茂 曹泰斌
魏国英 邵晓根 庄燕滨 邓 凯 吴国经 常晋义
许秀林 谢志荣 张家超 陶 洪 龚兰芳 刘广峰
丁 雁 方 岩 王一曙

委 员: (以姓氏笔画为序)

丁志云 及秀琴 石振国 李 翊 吕 勇 朱宇光
任中林 刘红玲 刘 江 刘胤杰 许卫林 杨劲松
杨家树 杨伟国 郑成增 张春龙 闵 敏 易顺明
周维武 周 巍 胡顺增 袁太生 高佳琴 唐学忠
徐煜明 曹中心 曾 海 颜友钧

序 言

进入 21 世纪,世界高等教育已从精英教育走向了大众教育。我国也适应这一潮流,将高等教育逐步推向大众化。培养应用型人才已成为国家培养国际人才的重要组成部分,且得到了社会各界的广泛支持。于是一大批有规模、有实力、规范化、以培养应用型人才为己任的高等学校得到了长足发展。这类高校办学的一个显著的特点是按照新时代需求和当地的需要来培养学生,他们重视产学研相结合,并紧密地结合当地经济状况,把为当地培养应用型人才作为学校办学的主攻方向。

这类学校的教学特点是:在教授“理论与技术”时,更注重技术方法的教学。在教授“理论与实践”时,更注重理论指导下的可操作性,更注意实际问题的解决。因此,这些学生善于解决生产中的实际问题,受到地方企事业单位的普遍欢迎。

为满足这类高校的教学要求,达到培养应用型人才的目的,根据教育部有关重点建设项目的要求和相关教学大纲,我们组织了多年在这类高校中从教,并具有丰富工程经验的资深教授、高级工程师、教师来编写这套教材。

在这套教材的编写中,我们提倡“实用、适用、先进”的编写原则和“通俗、精练、可操作”的编写风格,以解决多年来在教材中存在的过深、过高且偏离实际的问题。

实用——本套教材重点讲述本行业中最广泛应用的知识、方法和技能。使学生学习后能胜任岗位工作,切实符合当地经济建设的需要和社会需要。

适用——本套教材是以工程技术为主的教材,所以它适用于培养应用型人才的所有高校(包括本科、专科、技术学院、高职等),既符合此类学生的培养目标,又便于教师因材施教。

先进——本套教材所选的内容是当今的新技术、新方法。使学生在掌握经典的技术和方法之后,可用教材中的新技术、新方法去解决工程中的技术难题,为学生毕业后直接进入生产第一线打下坚实的基础。

通俗——本套教材语言流畅、深入浅出、容易读懂。尽量避开艰深的理论和长篇的数学推导,尽量以实例来说明问题,在应用实例中掌握理论,使学生轻松掌握所学知识技能,达到事半功倍的效果。

精练——本套教材选材精练。详细而不冗长,简略得当,对泛泛而谈的内容将一带而过,对学生必须掌握的新技术、新方法详细讲,讲透、讲到位,为教师创造良好的教学空间和结合当地情况调整教学内容的余地。

可操作——本套教材所有的实例均是容易操作的,且是有实际意义的案例。把这些案例连接起来,就是一个应用工程的实例。通过举一反三的应用,使学生能够在更高层次上创造性地应用教材中的新思想、新技术、新方法去解决问题。

本套教材面向培养应用型人才的高等学校,同时亦可作为社会培训高级技术人才的教材和需要加深某些方面知识技能的人员的自学教材。

编委会

前 言

在当今众多的计算机程序设计语言中,面向对象技术正逐步替代传统的面向过程的程序设计技术,成为目前软件的主流开发方法,Java 语言以它独到的面向对象特性在各类编程语言中脱颖而出,Java 语言的快速发展与广泛应用有目共睹。

本书是基于 Java 2 的面向对象程序设计的实用教程,以 Sun 公司的新版本 JDK1.3 技术为背景,融合了面向对象程序设计方法、数据结构、软件工程以及网络应用等知识体系,较为全面地介绍了面向对象的 Java 语言及其应用技术。

本书将面向对象的基本概念和 Java 程序设计语言原理有机地结合起来,通过面向对象的基本概念加深对 Java 程序设计语言基本机制的理解,反之,又通过 Java 语言生动地描绘出面向对象思想的实用价值。同时,依据 Java 的应用型特点,加强实用性,在基本概念、基本知识和一定扩展知识的基础上,作者有意增加了 Java 的应用成分,如网络通信应用、数据库应用、多媒体应用、J2EE 企业应用、Web 技术应用以及电子商务等应用领域的介绍与应用实例,力求展现出 Java 的特色以及最新技术,拓展读者对 Java 高级应用的视野,使得读者能够较为全面地认识与学习 Java。

本书可作为大专院校学生的教材,也可作为对面向对象编程和 Java 语言感兴趣的读者的自学用书。

根据作者的教学体会来看,由于 Java 语言所涉及内容的广泛性,计算机专业的学生在学习 Java 语言时,并不一定会感觉轻松或占有绝对优势,要提高读者的整体学习水平,应重视基本概念、基本理论的讲解,并能与实践相结合。因此,本书起点不高、内容全面、由浅入深、有张有弛,既考虑到初次接触 Java 的读者,又为具有一定编程经验者拓展了深度与广度空间,所以,本书既适合于计算机专业的学生与应用人员使用,也适合作为其他工科类专业学生及应用人员的学习教材。

本书在内容的编排上做了精心的设置与选取,将计算机科学中有关程序设计方法的核心内容融合到本书中,保证了一定的深度与广度。其中,增加了算法与程序设计基础、数据结构、面向对象的软件工程等内容,为读者学习程序设计语言创造了一个融会贯通的环境,并使读者逐渐建立软件开发的完整体系意识,在基本原理和基本方法中真正透彻理解与掌握面向对象程序设计方法,并能够将它应用到工程实践中去。为体现应用型人才培养的指导思想,提高 Java 语言的的实际应用能力,本书除提供了大量生动完整的应用实例以外,特意设置了实际技能训练章节,该实践性环节逐步引导学生综合运用所学知识解决实际问题,探索一个完整应用程序的设计方法,有利于开发应用软件能力以及独立分析、解决问题能力的提高。

全书共分 15 章。第 1 章对程序设计与面向对象程序设计的基本知识与基本概念进行了介绍,本章是学习面向对象程序设计的基础知识。第 2 章介绍了 Java 语言编程的基础知识。第 3 章和第 4 章以 Java 面向对象编程为例,分别介绍了面向对象编程中的四个特性及其在

Java 语言中的具体应用, 将四大特性分为两章讲述用于分散抽象概念与难点。第 5 和第 6 章分别介绍了数组、字符串以及常用的数据结构和算法的 Java 实现。第 7 章介绍 Java 的特色之一异常处理机制。第 8 章介绍 Java 输入输出技术。第 9 章介绍 Java 的多线程技术。第 10 章、第 11 章分别讲述基于 Swing 的 Java 图形用户界面设计和 Applet 技术与应用。第 12 章介绍了 Java 的高级编程技术, 包括网络编程、JDBC 数据库连接和多媒体应用。第 13 章概括了基于 Java 的实用技术, 主要讲述了企业应用技术, 也正是 Java 未来发展的重要应用领域之一。第 14 章介绍了 Java 的集成开发环境。第 15 章给出了为理解巩固各阶段的学习内容而设置的实践上机训练题以及授课完成后运用所学知识进行综合训练的内容。书中程序都经过上机调试, 读者可以放心阅读和使用。

本书由肖旻主编、廖雷副主编, 刘晓璐、杜青参编, 王志坚教授主审。各章编写情况如下: 肖旻编写第 1 章、第 3 章、第 4 章、第 7 章、第 13 章、附录 A、附录 B、附录 C; 廖雷编写第 2 章、第 5 章、第 9 章; 刘晓璐编写第 10 章、第 11 章、第 12 章; 杜青编写第 6 章、第 8 章、第 14 章及第 13 章的第 2 至 4 节。河海大学计算机及信息工程学院的王志坚教授、博导在百忙之中抽出时间, 认真细致地审阅了全书, 并提出了许多宝贵意见, 使得本书更趋完善, 作者深表感谢。

在本书的编写和出版过程中, 得到了中国电力出版社的指导与支持、作者所在学校及教务处、高教研究所、计算机工程系、教研室领导和同事的关心与帮助, 在此一并致谢。同时, 感谢邓晶、阚建飞、陈建红和屠立中老师的支持与帮助, 参与该课程的梁铁石、何娟、邵小巍、陈新会等同学也在本书的修改、完善过程中提出了许多有益的建议。

由于作者水平有限, 加之时间仓促, 书中错、漏之处在所难免, 欢迎广大同行和读者批评指正, 作者的电子信箱是: xiaomin_xy@sohu.com。

编 者

目 录

序 言 前 言

第 1 章 面向对象程序设计概述	1
1.1 程序设计基础	1
1.2 程序设计语言	4
1.3 程序设计方法	7
1.4 面向对象的基本概念及面向对象程序设计的特性	9
1.5 面向对象的软件开发过程	12
1.6 面向对象软件开发方法的特点	15
1.7 本章小结	15
习题	16
第 2 章 Java 语言基础	17
2.1 Java 发展概述	17
2.2 Java 语言的特点	17
2.3 Java 的运行环境与 Java 虚拟机	19
2.4 Java 的实现	20
2.5 一个简单的 Java 应用 (Application) 程序	23
2.6 一个简单的 Java 小应用程序	23
2.7 Java 程序的构成	24
2.8 Java 语言的词法单位	25
2.9 数据类型、常量与变量	26
2.10 常量与变量	27
2.11 运算符与表达式	28
2.12 流程控制语句	32
2.13 控制语句应用实例	39
2.14 本章小结	40
习题	41
第 3 章 类与对象	42
3.1 抽象与封装	42
3.2 对象	43
3.3 类	46

3.4 成员变量的定义和使用	49
3.5 方法的定义和实现	53
3.6 方法的调用	57
3.7 构造方法 (constructor)	60
3.8 静态 (static) 方法	62
3.9 main()方法	63
3.10 结束方法	63
3.11 本地方法	64
3.12 同步方法	64
3.13 本章小结	64
习题	64
第4章 继承与多态	66
4.1 继承	66
4.2 属性和方法的继承	67
4.3 属性隐藏和方法的覆盖	68
4.4 super 的使用	70
4.5 构造方法的继承	72
4.6 Java 的继承原则与访问控制	73
4.7 继承综合实例	74
4.8 多态	77
4.9 重载	77
4.10 运行时多态	80
4.11 抽象类与抽象方法	80
4.12 最终类与最终方法	82
4.13 接口	84
4.14 包	89
4.15 JAR 文件	93
4.16 内部类	94
4.17 本章小结	96
习题	97
第5章 数组、字符串与类库	98
5.1 数组	98
5.2 字符串	104
5.3 Java 类库介绍	108
5.4 本章小结	113
习题	113

第 6 章 常用算法及数据结构的 Java 实现	115
6.1 经典算法	115
6.2 线性数据结构	123
6.3 二叉树	131
6.4 本章小结	138
习题	138
第 7 章 Java 的异常处理	139
7.1 异常概念	139
7.2 Java 的异常处理机制	139
7.3 异常类的层次结构	140
7.4 抛出异常	142
7.5 处理异常	144
7.6 try-catch-finally 语句的嵌套	146
7.7 用户自定义异常	147
7.8 本章小结	148
习题	148
第 8 章 Java 的输入输出与文件处理	150
8.1 Java 输入输出类库	150
8.2 InputStream 和 OutputStream	151
8.3 Reader 和 Writer	159
8.4 文件的处理与随机访问	161
8.5 本章小结	165
习题	166
第 9 章 Java 多线程机制	167
9.1 线程概念	167
9.2 创建线程	168
9.3 如何实现多线程	169
9.4 线程的同步与死锁	173
9.5 本章小结	176
习题	177
第 10 章 Java 图形用户界面的设计与实现	178
10.1 图形用户界面 GUI 概述	178
10.2 Java GUI 的 AWT 组件介绍	180
10.3 Java 事件处理的基本原理	184
10.4 Swing 组件包及常用组件的使用	188

10.5 GUI 应用实例	224
10.6 本章小结	227
习题	227
第 11 章 Applet 技术及应用	229
11.1 Java Applet 基础	229
11.2 Applet 基本工作原理	230
11.3 Applet 的生命周期	232
11.4 HTML 文件中的常用标记	234
11.5 Applet 用户界面设计	236
11.6 小应用程序的安全性	243
11.7 Applet 小应用程序实例	244
11.8 本章小结	248
习题	248
第 12 章 Java 高级编程	250
12.1 数据库操作	250
12.2 Java 数据库连接 (JDBC)	253
12.3 Java 网络编程	265
12.4 Java 多媒体编程	285
12.5 本章小结	290
习题	290
第 13 章 基于 Java 的实用技术	291
13.1 概述	291
13.2 JSP	292
13.3 Java Servlet	301
13.4 EJB	303
13.5 J2EE	307
13.6 J2EE 应用开发和部署过程	310
13.7 J2EE 企业应用	312
13.8 RMI	314
13.9 CORBA	315
13.10 UML	317
13.11 Java 的未来发展	319
13.12 本章小结	319
习题	319
第 14 章 Java 面向对象编程环境	321
14.1 Java 开发环境概述	321

14.2 Visual J++6.0 集成开发环境介绍	322
14.3 本章小结	330
第 15 章 实训	331
实训 1 Java 程序的编辑、编译和运行	331
实训 2 Java 流程控制结构——分支结构	331
实训 3 Java 流程控制结构——循环结构	332
实训 4 类的定义与对象的使用	332
实训 5 方法和构造	333
实训 6 类的继承	334
实训 7 Java 的多态性	335
实训 8 抽象类与接口的实现	336
实训 9 数组和字符串	337
实训 10 创建链表及排序	337
实训 11 Java 的异常处理	338
实训 12 文件输入输出操作	339
实训 13 多线程的实现	339
实训 14 Java 布局设计	340
实训 15 Java 事件处理	341
实训 16 Applet 小应用程序的编写	342
实训 17 Java 的网络和数据库编程	343
实训 18 编写一个 JSP 页面	344
实训 19 熟悉 Visual J++6.0 集成开发环境	344
实训 20 综合实践技能训练	345
附录 A Java 关键字	347
附录 B Java 的 JDK 开发环境配置	349
附录 C 有关 Java 资源获取的 URL	350
参考文献	351

第 1 章 面向对象程序设计概述

在当今计算机应用开发领域中，面向对象技术正逐步取代面向过程的程序设计技术，成为当今程序设计领域的主流趋势。本章将面向对象程序设计中的基本问题即有关程序设计与面向对象概念进行结合，重点介绍程序设计与面向对象程序设计的基本概念、基本理论和基本方法，并将面向对象的软件开发方法原则、开发流程作了系统概括，为读者今后更好地从事程序设计和软件开发打下一定的基础。

1.1 程序设计基础

进入程序设计之前，需要对程序设计中所涉及的基本问题和基本概念进行了解。本节主要讨论程序设计基础中算法的概念、程序的概念、算法与程序设计关系和程序开发步骤等基本内容。

1.1.1 算法的基本概念

算法是程序设计的精髓。程序设计的实质就是构造解决问题的算法，并将其解释为计算机语言。通常，我们把解决问题的方法和步骤称为算法。它是对特定问题求解步骤的详细描述。从计算机应用角度来看，用计算机所能实现的操作或指令，来描述问题的求解过程的方法和步骤称为计算机算法。计算机算法根据计算机应用特点可以分为两类：即解决数值问题的算法和解决非数值问题的算法。对于同一个问题的求解，可以有多种不同的算法，应根据算法的正确性、运行效率和资源占用来选取合理的算法。

一个合理正确的算法，将为使用计算机解决问题打下良好的基础。对于任何一个正确的算法应具有以下五个重要的特征：

(1) 有穷性：算法在执行有限步之后应能够终结，并且在一个合理的时间限度内执行完毕。

(2) 确切性：算法的每一步骤必须有确切的定义，不应该有模棱两可的情况。

(3) 输入性：执行算法时，可以从外部获取数据，也可以由系统内部自动产生必要的数据，因此，一个算法可以有多个或 0 个输入。

(4) 输出性：一个算法必须有一个或多个输出，以反映对输入数据进行处理后的结果，否则，没有输出的算法是毫无意义的。

(5) 可行性：算法中的每个步骤都必须是计算机能够有效执行的、可实现的，并能够得到确定的结果。

1.1.2 算法的表示

算法需要用一种语言来描述,同时,为满足不同需求算法可以有各种描述方法。为将算法表达或描述出来,需要使用算法描述工具。较好的描述工具应是能将算法的操作步骤执行顺序既描述清楚,又要直观明确,而且便于修改和扩充。常用的算法描述工具包括:规范化自然语言、图形工具(流程图、N-S图等)、计算机语言和伪语言。

1. 自然语言

所谓自然语言,就是我们日常学习生活中使用的交流语言。例如,使用自然语言描述计算 $N=10!$ 的算法,描述如下:

S1: 使 N 等于 1, 即 $1 \Rightarrow N$;

S2: 使 I 等于 2, 即 $2 \Rightarrow I$;

S3: 使 N 等于 N 乘以 I , $N \times I \Rightarrow N$;

S4: 使 I 等于 I 的原值加 1, 即 $I+1 \Rightarrow I$;

S5: 如果 I 不大于 10, 转至 S3, 否则, 转至 S6;

S6: 输出 N 值, 算法结束。

可以看出,用自然语言描述算法,更接近人们的阅读习惯,通俗易懂,但文字冗长,容易产生“歧义性”。在程序设计中很少采用自然语言来描述算法。

2. 流程图

除了文字可以描述算法以外,我们经常使用图形表示算法,其中使用最为普遍的是流程图。在程序设计中,流程图可以灵活地表现判断条件、选择分支、循环执行等过程,能够较直观地表示算法的操作顺序和三种基本控制结构即顺序结构、选择结构和循环结构。实际上,不仅程序设计中使用流程图,而且在日常生活中的许多问题的具体实施过程也可以使用流程图理清思路,提高效率。同时流程图的概念与方法也可以应用于其他学科领域,可以将拟定的实施步骤或考虑问题的思路画成流程图,成为进行系统设计和整体实现的框架工具。建议读者在学习使用流程图进行程序设计后,能将它应用于各自的相关学科及领域。图 1-1 就是使用流程图表示计算 $10!$ 的算法。

最常用的流程图符号如图 1-2 所示。

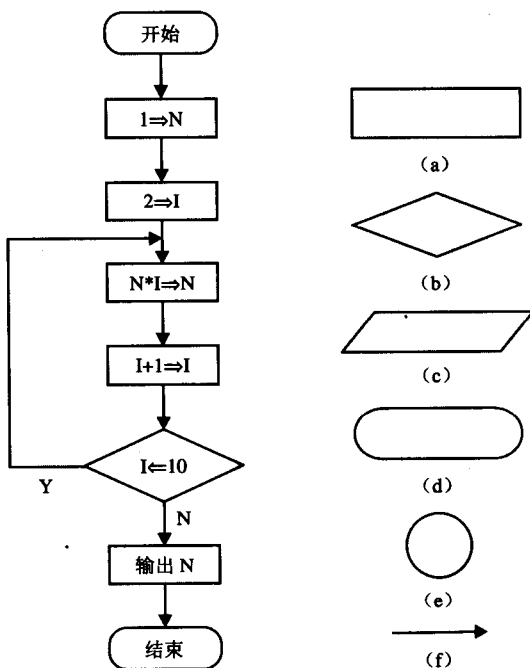


图 1-1 计算 $N=10!$ 的算法 图 1-2 常用流程图符号

其中:

- (a) 矩形框 (处理框): 各种形式的数据处理。
- (b) 菱形框 (判断框): 根据条件满足与否, 在可选的路径中, 判断选择某一路径。
- (c) 平行四边形框: 表示数据的输入与输出。
- (d) 端框 (起止框): 表示流程的开始与结束。
- (e) 圆圈 (连接点): 表示与流程图其他部分相连接的连接点。
- (f) 流程线: 表示流程的路径与方向。

3. 伪代码描述法

伪代码 (pseudo code) 是使用介于自然语言和计算机程序设计语言之间的文字和符号来描述算法, 即用计算机程序设计语言中具有的关键字和汉字相结合的方法表示算法的操作流程。用伪代码表示算法, 并无固定、严格的语法规则, 只要求把意思表达清楚, 但书写的格式要写成清晰易懂的形式, 它可以很方便地向计算机语言过渡。

例如: “打印 x 的绝对值” 的算法。

```
if  $x \geq 0$  then 输出  $x$ 
else 输出  $-x$ 
```

1.1.3 程序概念

如果没有程序, 计算机则不能实现任何功能。程序是计算机的一组指令, 为完成特定的功能, 通常使用计算机语言来描述算法以实现相应功能, 经过编译和执行最终完成程序设计的动作。

1.1.4 算法与程序

我们知道, 利用计算机解决各种问题时, 算法是至关重要的, 但它不能作为计算机的直接处理对象, 而根据算法编写出相应的高级语言程序则可以在计算机上高效运行。算法除了不能在计算机上运行以外, 其中仅给出了处理问题的步骤, 而且在描述中也没有严格的语法限制; 在程序中, 除了要严格遵守相应语言规定的语法格式外, 正如计算机科学家沃思提出的“程序=算法+数据结构”这一著名公式所示, 不仅要给出对问题处理的步骤方法 (即算法), 而且要对程序中所涉及的大量处理对象 (数据和数据结构) 进行定义说明。所以程序还涉及一个重要抽象内容就是组织处理操作对象的有关数据结构内容。为了使计算机真正解决实际问题, 需在设计一个正确算法的基础上, 用高级语言将算法表达成可以高效运行的程序。程序设计则是在吸收了算法与数据结构成分的基础上, 遵循一定的程序设计方法, 按照软件工程思想及方法进行软件开发。因此, 程序实际上包含了算法、数据结构、计算机语言和程序设计方法四个要素。本书学习使用 Java 语言进行程序设计, 为加深读者对程序及程序设计的深入理解和掌握, 书中设定专门的章节讲述程序设计中的经典算法和数据结构的 Java 实现。在使用 Java 语言进行程序设计中, 定义和使用最多、最主要的数据类型是“类”, 在类中既包含数据, 又包含数据操作的“方法”。在设计相对复杂的方法时, 需要先设计相应算法, 然后将算法用 Java 语言表达出来, 完成了由算法向程序设

计的转化。

1.1.5 程序设计的一般步骤

一般的程序设计，大致要经历以下几个步骤：

- (1) 根据任务，确定要解决的问题以及最终应达到的要求。
- (2) 分析问题，构造数学模型。为简化自然界与社会生活中的复杂现象，应先构造相应的物理模型，在此基础上，使用数学语言描述它，即建立数学模型，有利于使用计算机解决实际问题。
- (3) 选择计算方法。有时数学模型比较复杂，还需进行一定的近似处理，此时要确定使用哪一种近似方法，“计算方法”就是专门研究数值计算的近似方法的一门学科。
- (4) 确定算法，画出流程图。
- (5) 确定使用的某种语言，编写程序。
- (6) 进行程序调试。
- (7) 正式运行。运行中一般需反复调试，得出正确运算结果。
- (8) 整理资料，写成技术文档。对于较完整、复杂的任务，在完成运算后，应写出相应技术文档，如程序说明书，其中应包括任务要求、原始数据、算法流程图、程序清单、运行结果记录、操作使用说明等多项内容。

在进行程序设计时，应注意在达到可靠正确等基本要求的基础上，应力求简短、清晰易读，尽量少占用内存，提高运行效率，还应使程序易于修改、调试，具有较好的通用性和可移植性。

1.2 程序设计语言

程序设计语言是人机对话的媒介。人们把需要由计算机完成的工作写成一种计算机能够直接或间接接受的程序语言，这类语言称为程序设计语言。用程序设计语言和计算机对话编写程序的过程是一项人类特定的智力活动，也正是我们所要学习、掌握的一项技能。

程序设计语言是为了使人们更方便地设计程序（或者说开发软件）而进行一系列规则定义后，为人们所提供的一种程序设计环境，通常它们都表现为一种软件。在这种软件的开发环境里，人们可以利用它所提供的各种工具方便地进行程序设计。例如：程序员就是使用C语言或Java语言等各种程序设计语言来编写程序。

1.2.1 程序设计语言的发展

计算机语言的变迁和人类语言的发展一样，都经历了各自的不同阶段。人类从呱呱坠地的那一天起，就开始在适应自然界的过程中，逐步了解，逐步学习，逐步发展人类的语言体系，因为这是在这个世界上生存和发展的需要。而程序设计语言的发展，也是人类为适应未来世界计算机的应用而在不断演变、不断发展的一个过程。随着计算机的飞速发展，

程序设计语言也迅速发展起来。正如计算机硬件发展一样,程序设计语言也从简单到复杂、从低级到高级经历了从机器语言、汇编语言到高级语言的发展阶段,即从与机器密切相关的语言,向着独立于机器的面向问题的语言变迁。

1. 机器语言与汇编语言

在计算机问世的初期,是用二进制代码来表示计算机指令系统,机器指令的集合即为机器语言。因此机器语言是使用二进制代码编写程序、能被机器直接识别的语言。机器语言面向机器,执行效率高,但编程相当烦琐、可读性差,不能通用,阻碍了计算机的推广使用。为此,人们使用简单形象的特定助记符号来代替机器指令编写程序,这样的语言就称为汇编语言。用汇编语言编写的程序必须经过汇编程序的翻译,转换为机器语言表示的目标程序。汇编语言和机器语言都依赖于具体的计算机硬件,所以都是面向机器的语言,被称为“低级语言”。

2. 高级语言

由于汇编语言仍然是面向机器的语言,在使用和推广上仍有很大困难。经过计算机科学工作者的不懈努力,从20世纪50年代末开始,研制了一批中、高级语言。高级语言是最接近人类自然语言的计算机程序设计语言。我们通常所说的计算机程序设计语言都是指高级语言,它的程序语句接近自然语言,因此源程序易懂、易写、易改,它与计算机硬件关系不太紧密,因此程序便于移植。例如C、Visual Basic、Java等语言均为当前较流行的高级语言。

1.2.2 几种常见的编程语言及编程工具介绍

1. 传统的高级语言

主要包括于20世纪50年代末和60年代初研制出的BASIC、FORTRAN、ALGOL 60和COBOL等语言,它们在计算机语言发展史中曾经占有过重要的地位。

(1) FORTRAN 语言。FORTRAN 语言是一种适用于科学计算的高级程序设计语言。广泛运用于科学与工程计算领域。FORTRAN 语言与人们习惯使用的语言(如英文)和数学语言比较接近,它允许使用者将程序编写成类似数学式子的形式,还提供了丰富的标准程序库,受到广大科技工作者的青睐。它是第一个被广泛使用的高级语言。从FORTRAN 77到FORTRAN 90经历了两个标准。由于FORTRAN 语言较多地使用GO TO语句进行程序流程跳转,所以结构化不突出。

(2) COBOL 语言。COBOL 语言是英文Common Business-Oriented Language的缩写,即“面向商业的通用语言”。1960年在美国问世,是使用最广泛的商用数据处理语言。

2. 通用的结构化程序设计语言

通用的结构化程序设计语言由传统的高级语言发展而来,引入了结构化程序设计思想。有代表性的语言包括True BASIC、PASCAL、C、Ada等语言。

(1) PASCAL 语言。PASCAL 语言曾作为计算机科学的的教学语言,它是标准的结构化