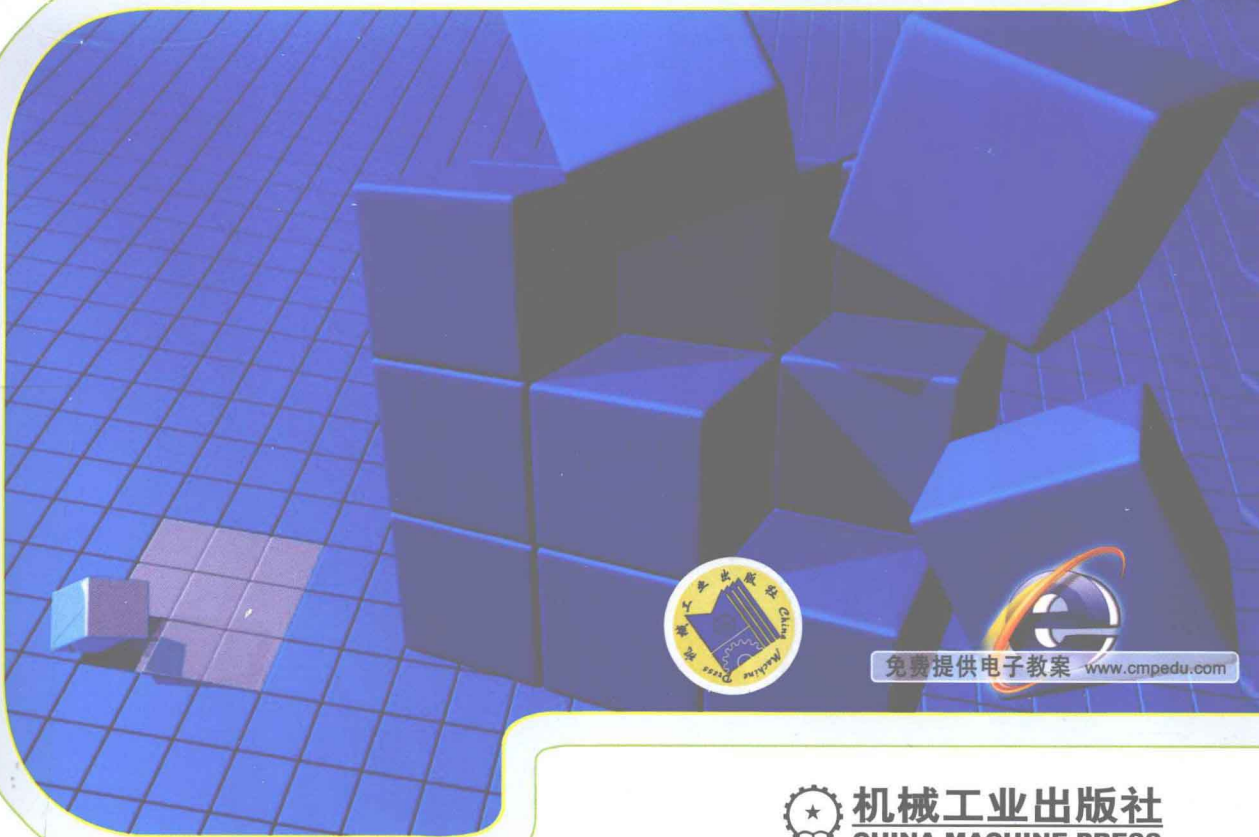




全国高等职业教育规划教材

程序设计基础 (Java语言)

余先锋 编著



免费提供电子教案 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

程序设计基础

(Java 语言)

余先锋 编著



机械工业出版社

本书是以 Java 语言为基础的程序设计入门教材,从有关程序设计的思想和方法入手,介绍程序设计概念及目的。内容包括程序编写的基本要求,程序代码的格式与编写规则、数据的概念以及数据类型的特点,变量、运算符、表达式、方法(函数)等概念及应用,同时以顺序、分支、循环等程序的控制方法为核心,通过数组、字符串、方法等应用,介绍程序控制的方法和技巧,并通过实例介绍穷举、递推、迭代、递归、查找、排序等常用的算法。

本书可作为大学一年级计算机及相关专业学生程序设计入门教材,也可作为程序设计初学者的学习参考书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 1239258369, 电话: 010-88379739)。

图书在版编目(CIP)数据

程序设计基础(Java 语言)/余先锋编著. —北京:机械工业出版社,2011.8
全国高等职业教育规划教材
ISBN 978-7-111-34450-6

I. ①程… II. ①余… III. ①Java 语言-程序设计-高等职业教育-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 137630 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:鹿征

责任印制:乔宇

三河市宏达印刷有限公司印刷

2011 年 9 月第 1 版·第 1 次印刷

184mm×260mm·15.75 印张·385 千字

0001-3000 册

标准书号:ISBN 978-7-111-34450-6

定价:29.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近60所高等职业院校的骨干教师对在2001年出版的“面向21世纪高职高专系列教材”进行了全面的修订和增补,并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多个品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

程序设计是计算机应用技术最基本的内容之一，“程序设计基础”是计算机软件及相关专业的必修课，传统上多以 C 语言作为学习程序设计的入门教学语言。以网络为中心的计算机是当今计算机应用的主流，网络程序设计已经成为应用程序设计的主体，而 Java 语言是一种通用的面向对象的编程语言，Java 程序设计是网络应用的重要组成部分，也是目前 Internet 上最主要的程序设计语言之一。因此本书选择 Java 语言作为“程序设计基础”的入门教学语言，使得学生能够直接通过 Java 语言掌握基本的程序设计方法，掌握程序设计的基本能力，为在后续的面向对象程序、Java Web 应用、JavaME 应用等其他 Java 语言相关程序设计课程的学习打下更为直接的基础。

本书作为 Java 语言程序设计的入门教材，充分考虑高职高专学生在学习程序设计的入门阶段的学习能力，从程序设计的基本概念入手，结合示例程序和上机实验，引导学生建立起关于程序设计的思想和方法，充分了解什么是程序设计以及程序设计所要达到的目的。在此基础上，使学生能够较好地掌握程序编写的基本要求，掌握程序代码的编写格式与规则，掌握数据的概念以及数据类型的特点，掌握变量、运算符、表达式、方法（函数）、输入输出方法等的概念及应用，同时将顺序、分支、循环等程序的控制方法作为程序设计基础的核心，通过数组、字符串、方法等的应用，充分掌握程序控制的技巧，完成一般应用程序的设计与实现。同时通过实例介绍穷举、递推、迭代、递归、查找、排序等常用的算法思想与程序实现过程。通过学习和模仿，使学生能够运用 Java 语言和开发工具，完成基本的计算机程序的设计、代码编写、调试运行的过程，并得出正确的结果，最终达到培养学生的逻辑思维能力、学习研究能力、创新能力、分析问题和解决问题的能力，使学生的理论知识和实践技能得到共同发展。

本课程的教学以课内讲授与课外练习、上机实验相结合的方式进行，建议周课时为 6 课时，共 80 课时。每周可安排 4 课时授课，2 课时上机实验，共 12 次实验作业。

书中所有程序对开发工具不做具体要求，任课老师可以根据实际情况选择合适的开发工具，并指导学生使用操作。

编者为本课程设计了一个教学网站，并提供本教材全部的程序源代码、电子教案、教学大纲、授课计划、实验作业文档等。这些资料文档以及本书教学网站的全部素材，均可以从机械工业出版社教学服务网 www.cmpedu.com、编者的教学网站 www.mitu.cn:1019/stu 下载。

编 者

全国高等职业教育规划教材 计算机专业编委会成员名单

主 任 周智文

副 主 任 周岳山 林 东 王协瑞 张福强
陶书中 龚小勇 王 泰 李宏达
赵佩华

委 员 (按姓氏笔画排序)

马 伟	马林艺	万雅静	万 钢
卫振林	王兴宝	王德年	尹敬齐
史宝会	宁 蒙	刘本军	刘新强
刘瑞新	余先锋	张洪斌	张 超
李 强	杨 莉	杨 云	罗幼平
贺 平	赵国玲	赵增敏	赵海兰
钮文良	胡国胜	秦学礼	贾永江
徐立新	唐乾林	陶 洪	顾正刚
康桂花	曹 毅	眭碧霞	梁 明
黄能耿	裴有柱		

秘 书 长 胡毓坚

目 录

出版说明

前言

第 1 章 程序设计基本概念 1

1.1 程序设计与程序设计语言 1

1.1.1 程序设计概述 1

1.1.2 程序设计语言 2

1.1.3 程序的编译与解释 3

1.1.4 程序设计的步骤 4

1.1.5 学习程序设计应该注意的几个方面 4

1.2 算法及算法的表示 5

1.2.1 算法的基本概念 5

1.2.2 算法的表示 6

1.2.3 算法的特点 8

1.3 Java 语言的特点 9

1.3.1 Java 语言的特点 9

1.3.2 面向对象 9

1.3.3 Java 程序的编译与运行特点 10

1.3.4 C 语言和 Java 语言的比较 10

1.4 Java 程序的开发环境与运行环境 11

1.4.1 Java 类库 11

1.4.2 Java 运行环境 11

1.4.3 Java 开发环境 12

1.4.4 Java 集成开发环境 13

1.5 关于 Java 知识体系 14

本章小结 14

习题 15

第 2 章 程序的编写与编译运行 16

2.1 Java 程序文件 16

2.2 程序的编译和运行 17

2.2.1 使用集成开发工具编译运行 18

2.2.2 使用命令行编译运行 19

2.3 Java 程序结构与语法规则 20

2.3.1 Java 程序基本结构 21

2.3.2 程序注释 21

2.3.3 程序的语法规则 22

2.4 标识符与关键字 25

2.4.1 标识符 25

2.4.2 关键字 26

本章小结 26

习题 27

第 3 章 变量与数据类型 28

3.1 数的进制 28

3.1.1 原码、反码和补码 28

3.1.2 不同进制数的表示方法 29

3.2 变量和常量 30

3.2.1 变量与常量的声明与初始化 30

3.2.2 变量 30

3.2.3 常量 32

3.2.4 字面值 32

3.3 基本数据类型 33

3.3.1 整型 34

3.3.2 实型 34

3.3.3 字符型 35

3.3.4 逻辑型 39

3.4 基本数据类型的转换 40

3.4.1 数据类型的自动转换 40

3.4.2 数据类型的强制转换 40

3.5 引用数据类型 42

本章小结 42

习题 43

第 4 章 程序的基本结构 45

4.1 程序的基本构成 45

4.2 运算符 46

4.2.1 算术运算符 46

4.2.2 关系运算符 50

4.2.3 逻辑运算符 51

4.2.4	位运算符	53	5.5	循环的中断控制	100
4.2.5	赋值运算符	55	5.5.1	break 语句	100
4.2.6	其他运算符	56	5.5.2	continue 语句	103
4.2.7	运算符的优先级	57	5.5.3	return 语句	104
4.3	表达式和语句	58	5.6	for-each 循环	105
4.3.1	赋值表达式	58	本章小结	106	
4.3.2	算术表达式	59	习题	106	
4.3.3	关系与逻辑表达式	60	第6章 数组	112	
4.3.4	位运算与移位表达式	62	6.1	数组与数组的声明	112
4.3.5	语句	63	6.1.1	数组的特点	112
4.3.6	复合语句	64	6.1.2	一维数组的声明	112
4.4	数据的输出与输入	65	6.1.3	数组的初值	114
4.4.1	数据的输出显示	65	6.1.4	数组的长度	115
4.4.2	从键盘输入数据	66	6.1.5	一维数组应用实例	116
本章小结		68	6.2	二维数组	118
习题		68	6.2.1	二维数组的声明与创建	119
第5章 程序的控制		71	6.2.2	二维数组的初值	119
5.1	顺序程序	71	6.2.3	二维数组的应用实例	121
5.2	分支控制	72	6.3	引用数据类型数组	123
5.2.1	if 分支结构	72	6.4	数组应用实例	125
5.2.2	if... else ...分支结构	73	6.5	数组元素的遍历	127
5.2.3	if ...else if ...结构	74	本章小结	128	
5.2.4	if 语句的嵌套	76	习题	129	
5.2.5	关于复合语句问题	77	第7章 字符串	132	
5.2.6	switch 控制语句	78	7.1	字符串类 String	132
5.2.7	switch 与 if...else...结构对比	82	7.1.1	String 类的创建	132
5.2.8	条件运算符的应用	82	7.1.2	String 类的常用方法	133
5.2.9	分支控制应用实例	84	7.1.3	String 字符串与基本数据的相互 转换	140
5.3	循环控制	86	7.1.4	String 字符串与字符数组的 转换	141
5.3.1	循环结构的组成部分	86	7.1.5	String 字符串数组	143
5.3.2	for 循环	87	7.2	字符串类 StringBuffer	144
5.3.3	while 循环	90	7.2.1	StringBuffer 类的创建	144
5.3.4	do...while 循环	93	7.2.2	StringBuffer 类的常用方法	145
5.3.5	循环基本要素与实现方法	95	7.2.3	String 字符串与 StringBuffer 字符串 的区别	148
5.3.6	循环体中的局部变量	97	本章小结	148	
5.4	多重循环	97			
5.4.1	多重循环的基本结构	97			
5.4.2	多重循环的控制技巧	99			

习题	149	9.6 查找算法	194
第8章 方法	152	9.6.1 线性查找	194
8.1 方法的设计和使用	152	9.6.2 二分法查找	195
8.1.1 方法的声明	152	9.6.3 费氏查找	197
8.1.2 方法调用、方法参数与返回值	154	9.6.4 插值查找	198
8.1.3 类方法和实例方法	156	9.7 排序算法	200
8.1.4 关于 main() 方法	157	9.7.1 交换排序	201
8.1.5 方法应用实例	157	9.7.2 插入排序	203
8.1.6 方法的嵌套调用	158	9.7.3 选择排序	205
8.2 变量的作用域与参数传递的方式	159	9.7.4 希尔排序	206
8.2.1 类变量与实例变量	159	9.7.5 快速排序	208
8.2.2 局部变量	160	9.7.6 字符串数组的查找与排序	210
8.2.3 基本数据类型参数传递	162	9.7.7 使用 Arrays 类的 sort 方法 排序	213
8.2.4 引用数据类型参数传递	162	本章小结	214
8.3 常用 Java 类库的方法及使用	164	习题	215
8.3.1 系统类 System 中常用的方法	164	附录 A 上机实验	217
8.3.2 字符串类 String、StringBuffer 中常用的方法	166	实验 1 简单程序的编辑、编译和运行	217
8.3.3 数学 (Math) 类中的常用方法	168	实验 2 数据类型的定义与输出、数据类型的转换	218
8.3.4 数字格式化类 NumberFormat 中常用的方法	170	实验 3 表达式与语句	220
8.3.5 日期时间类中常用的方法与应用	171	实验 4 分支程序设计	222
本章小结	177	实验 5 循环程序设计	225
习题	178	实验 6 数组应用设计	227
第9章 算法与程序技巧	181	实验 7 循环程序与数组	228
9.1 程序算法及特点	181	实验 8 字符串应用	230
9.1.1 算法概述	181	实验 9 方法的应用	232
9.1.2 程序算法的特点	182	实验 10 程序综合设计	234
9.1.3 常用算法及分类	182	实验 11 程序算法应用 (1)	236
9.2 穷举算法	183	实验 12 程序算法应用 (2)	238
9.3 递推算法	186	附录 B ASCII 码表	240
9.4 迭代算法	188	参考文献	241
9.5 递归算法	190		

第1章 程序设计基本概念

本章概要

- 程序设计概述
- 程序设计语言
- 学习程序设计的基本要求
- 算法的基本概念
- Java 程序语言的特点及开发与运行环境的建立

学习要点

了解程序设计在计算机应用中的作用，掌握程序语言的概念，了解程序设计的基本步骤，了解程序算法的基本概念，掌握 Java 语言的特点及运行机制，掌握 Java 程序开发与运行环境的建立的过程和方法。

1.1 程序设计与程序设计语言

由计算机带来的信息革命，把人类带到到信息化社会，而信息化的社会对人才素质和知识结构都提出了更高的要求。对于高等教育以及职业教育的各个学科，计算机的作用已经不仅仅是一种工具，而是各个学科的一个重要的组成部分。随着信息技术的不断发展，正确熟练地使用计算机软硬件成为当代大学生必须掌握的一项基本技能。其中，学会计算机程序设计，是计算机及相关专业学生掌握计算机应用技术的最基本内容之一，也是计算机及相关专业学生必须掌握的基本技能。

1.1.1 程序设计概述

程序设计 (Programming) 是指设计、编制、调试程序的方法和过程。计算机的所有操作都是根据一系列的命令来完成的，当我们使用计算机进行有关活动、完成指定任务时，必须通过一种计算机可以接受的语言与计算机进行交流，使得计算机能按照人们的意愿完成相应的操作和功能。与人类语言一样，计算机语言也是由一些根据特殊规则构成的字符组成。这些字符按一定的格式要求，被依次排列连接组成命令或语句。这种针对计算机的操作，由一系列的命令或语句组成的序列叫做程序，而编写程序的过程叫做程序设计。程序员通过程序设计语言准确地表达他们所想要表达的目的。对那些从事计算机应用的人来说，懂得程序设计语言是十分重要的，因为所有计算机的计算、控制与应用都需要程序设计语言才能够完成。

程序设计也叫做软件开发。计算机软件都是通过程序设计所产生的，程序是软件的本体，软件的质量主要通过程序编写的质量来体现。因此，程序设计对于计算机软件是非常重要的，程序的功能和设计质量，直接影响着计算机的功能、应用范围和应用能力。

程序设计的内容由多方面组成，涉及有关的基本概念、程序语言、开发工具、数学模型、设计方法等诸多方面。

1.1.2 程序设计语言

程序设计语言 (Programming Language) 就是用于编写计算机程序的语言，通常简称为编程语言，是一组用来定义计算机程序的字符记号及语法规则。程序设计语言是一种被标准化的专用语言，用来向计算机发出指令。在程序设计过程中，程序员根据所要设计的软件功能的要求，确定程序的算法，然后根据算法将这些字符记号根据约定的语法规则进行有序组合就形成了程序。程序设计语言使得程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的动作，从而完成对计算机指定的操作。

程序设计语言是专门使用在计算机上的，但它们也可以用来定义算法或者数据结构。

1. 程序设计语言的 3 个因素

程序设计语言包含 3 个因素：语法、语义和语用。

- 语法：表示程序的结构或形式，表示构成语言的各个字符记号之间的组合规律。
- 语义：表示程序的含义，表示按照各种方法所描述的各个记号的特定含义。
- 语用：表示程序与使用者的关系。

2. 程序语言的基本构成

程序语言的种类有很多，不同的程序语言有着很大的差别。一般情况下，不同的程序语言其基本构成是一样的。程序语言的基本构成有 4 个方面。

- 1) 数据部分：用以描述程序中所涉及的数据。
- 2) 运算部分：用以对程序中所包含的数据进行运算。
- 3) 控制部分：用以实现对程序流程的控制。
- 4) 数据传输部分：用以实现程序中数据的传输，包括要进行运算处理的数据和运算后所产生的数据的传输。

以设计一个对学生成绩进行汇总统计的程序为例，以上 4 个部分内容如下。

- 1) 数据部分：一个或若干班级学生的学号、姓名、课程名称、各门课程的成绩数据等。
- 2) 运算部分：计算最高成绩、最低成绩、总分、课程平均分、班级平均分，优、良、及格、不及格人数及比例等。
- 3) 控制部分：如根据班级人数，使用循环方法对每一个学生的成绩进行计算；使用条件判断方法，根据分数判断该学生的成绩是否及格等。
- 4) 数据传输部分：输入学生成绩数据、将学生成绩及统计结果进行显示打印，保存到文件或指定的数据库当中等。

3. 程序语言的分类

程序语言根据不同的应用，有不同的分类。

(1) 按语言级别分类

按照语言的级别分类，可分为低级语言和高级语言。

低级语言包括机器语言和汇编语言。它的特点是与特定的机器有关，运行效率高，但使用复杂、烦琐、费时、易出差错。机器语言是表示成二进制码形式的机器基本指令集，或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果。

高级语言的表示方法比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，其语法表示与日常的语言比较接近，易学、易用、易维护。由于机器只能接受低级语言，因此高级语言程序必须翻译成相应的低级语言程序后，才能够在计算机上运行。一个高级语言程序单位需要对应多条机器指令，运行效率要比低级语言低。Java 程序语言是一种高级语言。

(2) 按语言应用范围分类

按照语言的应用范围分类，可分为通用语言和专用语言。

通用语言可以编写不同类型的计算机应用程序，如 Java 语言可以编写计算、多媒体、数据库应用、通信等不同类型的程序，是一种通用语言。

目标单一的语言称为专用语言，如 SQL 是专门实现数据库操作的语言。

还有一些语言也带有专用语言的特点，如 ASP、JSP 语言是用于 Web 网站开发设计的语言。

(3) 按语言特性分类

按照语言的特性分类，可分为面向过程和面向对象语言。

面向过程语言是以过程或函数为基础的，程序中数据和对数据的操作是分离的。这种语言对底层硬件、内存等操作比较方便，适合接口程序、驱动程序、操作系统、低层开发应用程序、游戏等软件的设计。使用面向过程语言，程序的编写、调试、维护等会比较麻烦。早期的程序语言一般是面向过程的，如 C 语言、Pascal 语言等。

面向对象语言就是面向具体的程序应用功能，程序所要完成的操作都以对象为基础，而对象则包含了对对象的数据以及对数据的操作。面向对象语言是由面向过程语言发展而来的，是当今程序设计语言的主流。面向对象程序语言有 Java、C++、C# 等。

1.1.3 程序的编译与解释

多数情况下程序员都使用高级语言编写程序。这种由程序员按照一定的程序设计语言规范书写的文本文件叫做源代码或源程序代码。但由于计算机只能认识和接受机器语言，因此，需要通过编译的过程把高级语言翻译成计算机可以识别的机器语言。将用高级语言编写的程序（源程序）整体翻译成计算机可以接受和识别的机器语言（目标程序），这个翻译过程称为编译。编译的过程是通过一个编译器（也就是编译软件）对源程序进行翻译，然后生成和输出计算机可以识别的机器语言文件。这种编译后所生成的文件称为目标程序。

经过编译所生成的目标程序文件一般有两类：可执行文件和字节码文件，可执行文件可以由计算机的中央处理器直接运行，如 C 语言程序经编译生成的是可执行文件，文件的扩展名为 .exe，可以直接运行。字节码文件也称为二进制码文件，字节码文件不能直接运行，需要通过指定的模拟器来执行。Java 程序编译后生成的就是字节码文件，文件的扩展名为 .class，需要有 Java 虚拟机（JVM）对字节码文件进行解释执行。

编译后的程序脱离了源程序独立地执行。如果对源程序进行了修改，则需要对源程序重新进行编译。

还有一种情况，源程序代码不需要单独的编译过程，而是在程序开始运行时才进行即时翻译，也就是边编译，边运行。这种翻译机制就被称做解释。使用解释方式运行的程序，其运行速度往往比编译的程序要慢，但一般都更具灵活性，因为它们能够与执行环境互相作用，更方便于程序的调试。BASIC 语言就是一种解释语言。

也有程序语言采用解释和编译混合形式的。

1.1.4 程序设计的步骤

完成一个计算机程序的设计，主要有以下几个步骤。

- 1) 问题分析：对所要解决的问题事务进行分析，确定解决问题的思路 and 过程。
- 2) 设计算法：根据对所要解决的问题的分析，确定所需的算法和程序结构。
- 3) 建立流程图：根据算法和程序结构画出程序流程图。良好的程序流程图可以使得程序流程清晰，逻辑严谨。流程图也是软件工程中重要的文档资料。
- 4) 程序编写：按流程图编写程序。
- 5) 编译：对完成的程序代码进行编译。在编译过程中，修正程序中的错误。
- 6) 检测和调试：运行编译好的程序，检测和分析程序运行结果，对程序运行过程中出现的问题进行分析，进一步修改和完善程序。
- 7) 程序改进和再设计：程序在完成后的使用过程中，可能会出现一些未曾预料的情况，或未能达到预先的设计要求，这时应该对程序进行改进和再设计。
- 8) 编写有关的程序文档：在程序设计前、设计过程中以及程序完成后，应该及时按软件设计的规范编写程序文档。

程序设计流程如图 1-1 表示。

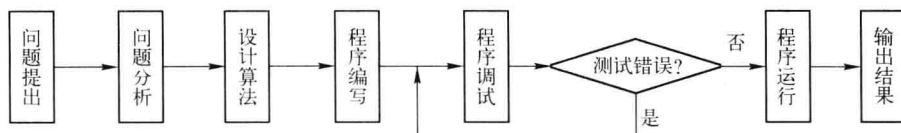


图 1-1 程序设计流程

1.1.5 学习程序设计应该注意的几个方面

要成为一个合格的程序员，应该具有良好的素质。这些素质要求包括如下几个方面。

1) 掌握好程序设计规范。程序设计规范是进行程序设计的具体规定，而软件开发是一种工程性的工作，因此必须严格按照程序规范进行代码编写。遵循程序设计规范，培养标准化的代码编写习惯，对所设计的程序的功效以及软件的可靠性、易读性和易维护性都起着重要的作用。

2) 注意培养良好的分析问题和解决问题的能力。程序设计是一种高级的逻辑思维活动，程序员应该具有一定的系统分析和程序设计的能力，将现实中的问题进行整理抽象，转变为程序代码，实现对现实问题的解决。将问题化为程序，前提是对程序语言的熟练掌握。

3) 严谨的工作态度和缜密的思维方法。程序设计是一项十分严谨的过程，需要程序员有着极大的耐心和细心。除了必须严格按照程序规范、语法要求进行代码设计，还必须对程序所要解决的问题进行细致的分析，充分考虑到各种可能发生的情况和细节，并在程序中予以解决。学习程序设计除了使自己掌握一种技能以外，额外的收获是可以很好地训练自己的严谨的工作态度和缜密的思维方法。

4) 培养阅读程序代码的能力。无论是程序语言学习还是进行程序设计，程序员都需要培

养良好的阅读程序代码的能力。在学习以及进行程序设计的过程中，很多时候都需要阅读和参考别人已经设计好的例程，只有这样才能开阔自己的视野，掌握不同的程序设计技巧。在学习程序设计的初期，读懂别人的程序，并能够进行模仿设计，是一个很重要的学习过程。阅读大量别人的程序并予以吸收掌握，可以少走许多弯路，较快地提高自己程序设计水平。

5) 先精后博。先精通一种程序语言，然后再根据需去学习其他程序语言。程序语言种类繁多，功能各不相同，但各种程序设计语言是可以触类旁通的。学习程序设计语言，忌讳多而不精。如果能够熟练和精通一种程序语言，再学习和掌握其他语言是比较容易的。

6) 培养良好的团队精神和协作能力。现在多数计算机软件都比较庞大，不是靠一两个人能够完成的，往往需要通过一个团队来实现，这就需要程序员具备良好的团队协作精神。

7) 要有总结的能力。在学习过程中，要学会总结，对所掌握的知识点、程序规范、语法规则，以及设计过程中对问题的分析和解决方法、阅读别人程序后的心得体会等，都应该进行总结归纳，并形成文档。另外，在完成一个程序设计的过程中，根据程序设计的规范，写好各种软件文档，也是一种很好的总结过程。

8) 掌握算法。学习程序设计，无论使用哪一种程序语言，计算机学科中算法和数据结构是最重要的，程序语言只是最基本的工具，算法和数据结构才是程序设计的核心。通过本课程的学习，掌握了基本程序语言应用和基本的算法后，还需要不断地学习，学好数据结构，掌握数据库知识，掌握更多的程序算法，以提高自己的程序设计能力和水平。

1.2 算法及算法的表示

程序语言只是设计程序的工具，而程序设计的核心是算法。掌握程序算法，才是学习计算机程序设计最重要的内容。

1.2.1 算法的基本概念

解决任何问题，都必须掌握问题的规律和本质，根据一定的方法和步骤进行。使用计算机程序来实现问题的求解，也应该根据实际问题采取科学的、有效的方法和步骤，这种科学有效的方法和步骤就是算法。

解决一个问题，可以有不同的方法和步骤，所以同一问题的程序算法不是唯一的。不同的算法也有优劣之分，方法的繁简也各不相同。确定解决问题的算法，首先要求算法应该是正确的，同时要求算法尽可能简单，用较少的步骤，较高的效率，达到最佳的效果。

程序设计的算法分为数值运算和非数值运算两种。数值运算应用于数值求解，多为各种数学运算。人们对数值计算领域有很深入的研究，各种数值计算都有现成的数学模型和比较成熟的算法，有许多现成的算法程序可供使用。对于初学者来说，研究和讨论算法，是对程序设计能力的一种很好的训练。

程序设计中，应用更多的是非数值运算。非数值运算主要是应用于各种事务处理的计算机管理应用。由于计算机管理应用所覆盖的面极广，各行各业的计算机应用各不相同，因此非数值运算的种类繁多，很难规范。非数值运算的算法，往往需要程序设计人员在具体的问题事务的解决过程中去不断地研究和总结。

学习程序设计过程中，除了自己研究设计算法以外，还应该多查找资料，收集和学习已

经成熟的各类算法和算法程序。

1.2.2 算法的表示

程序设计中，有几种不同的算法表示方法：自然语言、传统流程图、N-S结构流程图、伪代码、PAD图等。下面简单介绍前三种。

1. 自然语言

使用人们日常生活中比较习惯的语言表示，这种表示比较通俗易懂，容易理解。但也存在表述烦琐、容易出现歧义、表达不清晰等问题。这种算法表示一般不能在算法流程中使用，但可以作为算法流程中的文字表述的补充。

【例1-1】 瓶子A中装酒，瓶子B中装醋，将酒和醋交换瓶子。

分析：瓶子A中的酒和瓶子B中的醋不能直接交换，需要第三个瓶子C。交换前应该考虑C瓶是不是空的。

使用自然语言的算法表示如下。

步骤1：查看C瓶是不是空的。

步骤2：如果C瓶是空的，转步骤4。

步骤3：C瓶不是空的，把C瓶清空。

步骤4：将瓶子A中的酒倒入瓶子C： $A \rightarrow C$ 。

步骤5：将瓶子B中的醋倒入瓶子A： $B \rightarrow A$ 。

步骤6：将瓶子C中的酒倒回瓶子B： $C \rightarrow B$ 。

完成交换。

2. 传统流程图

传统流程图使用框图来表示算法流程。用框图表示算法，比较直观形象，易于理解，表示也比较简单清晰。图1-2所示是美国国家标准化协会（ANSI）规定的常用的流程图符号。

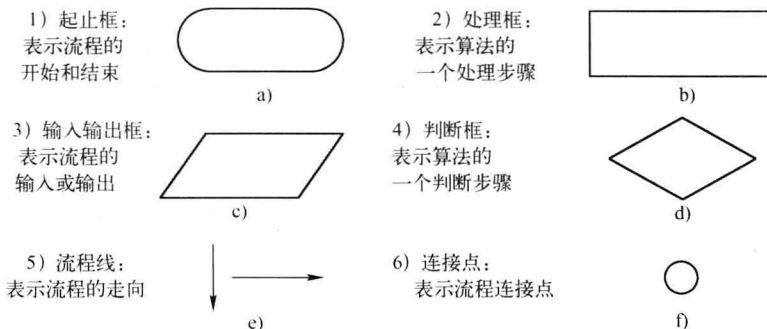


图1-2 几种常用的传统流程图符号

a) 起止框 b) 处理框 c) 输入输出框 d) 判断框 e) 流程线 f) 连接点

程序基本的流程结构有三种，用以下传统流程图来表示。

(1) 顺序结构

顺序结构按算法的步骤先后顺序逐行执行，其间没有条件判定和循环重复过程。顺序结构是程序中最简单的结构。顺序结构见图1-3a。

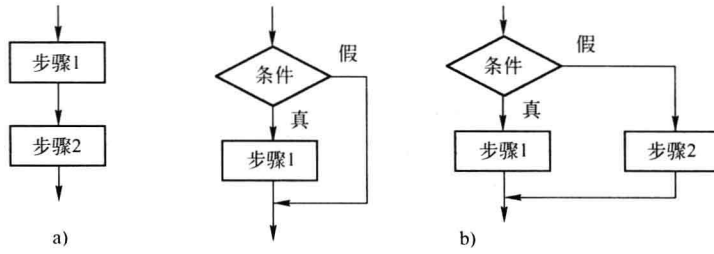


图 1-3 顺序结构与条件结构图

a) 顺序结构 b) 条件结构

(2) 条件结构

条件结构又称为分支结构或选择结构。这种结构中至少有一个条件判断，根据条件的是否成立（通常表示为结果为“真 (true)”或者“假 (false)”），来确定程序执行不同的操作。图 1-3b 是两种不同的条件结构。图 1-4a 是一个简单的条件结构的实例，例子中根据条件“成绩 ≥ 60 ”是否成立，确定程序的分支走向。

(3) 循环结构

程序流程中，往往有大量重复性的操作，如计算学生的平均成绩，其计算方法对于每个学生都是相同的，每个学生都要重复一样的计算。这种重复计算就需要使用循环结构来进行。循环结构中的循环次数应该是有限的，否则就称为无限循环或死循环。因此循环都有一个对循环次数进行限定的条件判断。循环结构如图 1-5 所示，图中是两种不同类型的循环。

循环分为“当”型结构和“直到”型结构。图 1-5a 为“当”型结构，也就是当条件满足时，执行循环操作，如果条件不满足则结束循环操作。当型循环是先判断条件，然后执行循环。图 1-5b 为“直到”型结构，直到型循环先进入循环，然后判断条件，直到条件不满足时结束循环操作。直到型循环是先执行循环，然后判断条件。“直到”型结构至少完成一次循环。

图 1-4b 是一个循环结构的例子。该循环程序对整数 1~100 进行累加。

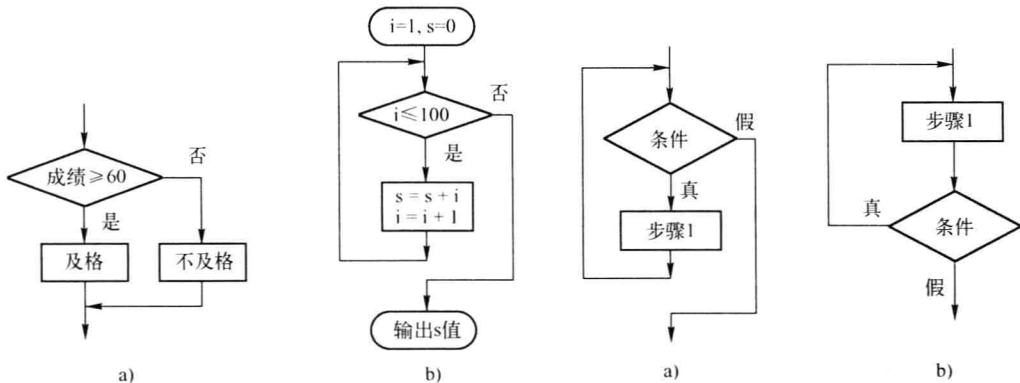


图 1-4 条件结构与循环结构示例

a) 条件结构例子 b) 循环结构例子

图 1-5 程序的循环结构图

a) “当”型循环结构 b) “直到”型循环结构

传统的流程图绘制简单，阅读起来比较直观，但由于使用了流程线，并且对流程线没有严格的要求，因此如果流程线使用得不合理，流程图会变得很杂乱，流程图的可读性会降

低。流程图越大，可读性就越差。

3. N-S 结构流程图

N-S 结构流程图本质上和传统的结构流程图是相同的。但这种结构流程图取消了流程线，全部算法集中在矩形框内。N-S 结构流程图功能明确，流程的入口和出口清晰，并且可以嵌套，特别适用于面向结构的程序流程描述。与传统的流程图相比，N-S 结构流程图可读性要好一些。与前面传统的结构图（图 1-3 ~ 图 1-5）相对应，以下结构图（图 1-6 ~ 图 1-8）分别是三种不同的 N-S 结构流程图和示例结构图。

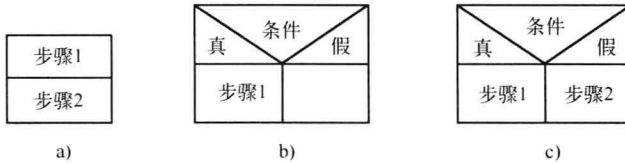


图 1-6 N-S 结构流程图—顺序结构与条件结构

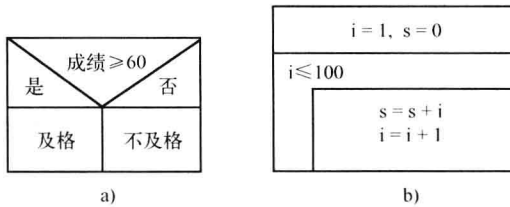


图 1-7 N-S 结构流程图—条件结构与循环结构示例

a) 条件结构例子 b) 循环结构例子

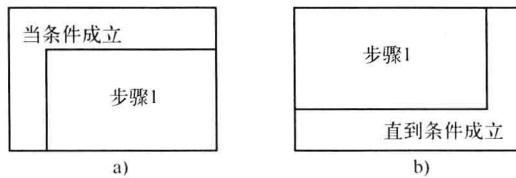


图 1-8 N-S 结构流程图—循环结构

a) “当”型循环结构 b) “直到”型循环结构

顺序、条件、循环三种算法结构，是程序设计最基本的结构，也是学习程序设计必须掌握的最基本的内容。

有关流程图的具体应用，将以后面相关的程序中体现。

1.2.3 算法的特点

确定程序的算法，应该注意以下几个特点。

- 1) 有穷性：算法的操作步骤应该是有限的，必须要有算法的终结点。
- 2) 确定性：算法的每个步骤必须是明确的，不能模棱两可。
- 3) 有效性：算法的每个步骤在不同的约束条件下，都应该能够被有效地执行，得到正