普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术教材

程序设计导论

——Java编程

□ 吴萍　主编

□ 蒲鹏　朱丽娟　编著

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精炼，实例丰富
- 可操作性强，实用性突出

清华大学出版社　●　北京交通大学出版社

普通高等教育“十一五”国家级规划教材
高等学校计算机科学与技术教材

程序设计导论——Java 编程

吴　萍　主编
蒲　鹏　朱丽娟　编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书以 Java 编程语言为平台，系统地介绍了程序设计的基本概念、Java 语言和面向对象程序设计技术。全书共分为七章，内容包括程序设计概述，Java 语言基础，对象、字符串与数组，面向对象编程，异常处理和输入输出，Applet 程序及图形用户界面 Swing 编程。

本书为“十一五”国家级规划教材，适用于程序设计的初学者及面向对象程序设计语言 Java 的初学者。本书可作为大学各科有关课程的教材或教学参考书，亦适用于相关的培训和自学。

图书在版编目(CIP)数据

程序设计导论：Java 编程/吴萍主编. —北京：清华大学出版社；北京交通大学出版社，2008. 12
（高等学校计算机科学与技术教材）
ISBN 978 - 7 - 81123 - 403 - 9

Ⅰ. 程… Ⅱ. 吴… 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(2008)第 144955 号

责任编辑：谭文芳
出版发行：清华大学出版社　邮编：100084　电话：010 - 62776969　http://www. tup. com. cn
　　北京交通大学出版社　邮编：100044　电话：010 - 51686414　http://press. bjtu. edu. cn
印 刷 者：北京东光印刷厂
经　　销：全国新华书店
开　　本：185 × 260　印张：19　字数：483 千字
版　　次：2008 年 12 月第 1 版　2008 年 12 月第 1 次印刷
书　　号：ISBN 978 - 7 - 81123 - 403 - 9/TP · 438
印　　数：1 ~ 5000 册　定价：29. 00 元

前　言

面向对象技术是程序设计方法学的一场革命，目前已成为计算机开发领域的主流技术。Java 作为一种面向对象的程序设计语言，无论在网络程序设计中，还是在程序设计的教学中都呈现出显著的优势。本书定位于高等学校非计算机专业学生，目标是成为学生的第一门编程课程。针对非计算机学生的特点，偏重于实用性。在介绍 Java 编程的同时讲解面向对象程序设计的主要原则和方法，培养学生利用面向对象的技术分析和解决问题的能力，并指导学生在较短的时间内学会利用最先进的 Java 工具软件开发 Java 应用程序，以顺应信息时代对人才的新需求。

Java 程序设计的内容涵盖面非常广，学习起来有一定的难度。对于非计算机专业的初学者来说，书中内容的编排、组织，实例的选取都需要通过精心设计，遵循由浅入深、循序渐进的原则，同时也应保证有一定的深度和广度。使学生通过对本教材的学习，能够快速掌握 Java 编程语言的核心内容并能灵活运用所学的 Java 语言知识及面向对象的编程思想。

通过第 1 章的学习，程序设计语言的初学者能够迅速掌握程序设计的基本思想和方法，为后续章节的进一步学习打下基础。本书难点和重点安排合理：读者可按书中的章节顺序学习，以提高学习效率。对一些重难点知识，书中通过剖析其本质，让读者能够从根本上理解、掌握并灵活运用这些知识。本书实用性强：提供了大量针对性的实例，编程中的注意事项及出现问题的解决方法等书中都会逐一说明，带领读者迅速掌握编程的全过程。本书涵盖了 Java 编程语言的核心内容：比较完整地介绍了 Java 的语法、面向对象的特性、核心类库的使用和图形用户界面的编程等。通过这一层次的学习，读者能够较全面地掌握 Java 面向对象的程序设计思想和技术。

本书第 1 章、第 2 章、第 3 章由吴萍编写，第 6 章、第 7 章由蒲鹏编写，第 4 章、第 5 章由朱丽娟编写。吴萍负责全书的内容结构设计和统稿工作。

对于书中的疏漏和不妥之处，恳望读者批评指正。

编　者

2008 年 10 月于华东师范大学

目　录

第1章 概 述

本章学习指引：

- 计算机软硬件基本知识
- 程序设计的基本概念
- 面向对象的程序设计方法
- Java 语言的基本特性和运行平台
- Java 程序的开发环境
- 最简单的 Java 程序实例

1.1 计算机和程序

计算机是一种能按照预先存储的程序，自动、高速地进行大量数值计算和各种信息处理的现代化智能电子设备。计算机能实现自动运算，是由于它采用了“程序存储”的工作原理。这一原理是1946年由美籍匈牙利数学家冯·诺依曼等在一篇题为《初步探讨电子计算机装置的逻辑结构》论文中首先提出并论证的，它确定了计算机的基本组成和工作方式。

现代计算机系统从性能指标、运算速度、工作方式、应用领域和价格等方面都有了很大的发展，但基本结构仍一直沿袭冯·诺依曼的传统框架。

1.1.1 计算机的基本组成

计算机系统由硬件和软件两大部分组成，硬件由 CPU（Central Processing Unit，中央处理器）、存储器和各种输入输出（Input/Output，I/O）设备等主要功能部件组成，软件则包括计算机运行所需的各种程序、数据及相关文档资料。

1. CPU

运算器和控制器合在一起称为中央处理器，即 CPU，它是计算机的核心。

运算器是计算机对各种数据或信息进行算术运算和逻辑运算的主要部件，由寄存器、加法器和移位器等逻辑电路组成。

控制器的作用是指挥整个计算机的各个部件按照指令的功能要求有条不紊地协调工作。它由程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）和微操作控制电路等组成。指令寄存器暂时保存正在执行的指令，当计算机工作时，控制器依次从内存储器中读取程序的一条指令，存入指令寄存器；程序计数器用来对程序中的指令进行计数，存放的是将要执行的指令在内存储器中的存储地址，使得控制器能依次读取指令；指令译码器则用于对指令的操作码进行译码，产生的译码信号能识别该指令要进行的操作，并传送给微控制部件，以产生相应的控制信号。

2. 存储器

程序和数据存放在存储器中。存储器是计算机的记忆和存储部件，按功能一般可分

为内存储器(内存)和外存储器(外存)两大类。内存由半导体存储器所组成,直接与运算器和控制器相连接。外存容量大、价格低,但存取速度较慢,不能与 CPU 直接交换信息。一般情况下,程序以文件的形式存储在外存中,当需要执行时,再从外存调入到内存中。

如图 1 -1 所示,内存由若干存储单元组成,每个存储单元都有地址,可以字节(Byte)为单位存放二进制数据或一条由二进制编码表示的指令。

3. 输入输出设备

输入输出设备简称 I/O 设备,它是外部与计算机交换信息的渠道,用户通过输入设备将程序、数据、操作命令等输入计算机,输出设备则将计算机处理的结果显示或打印出来。

常用的输入设备有:键盘、鼠标器、扫描仪、光笔、数字化仪和语音输入装置等。

常用的输出设备有:显示器、打印机、绘图仪和声音播放装置等。

4. 程序的概念

计算机能够接收的、指示计算机完成特定功能的一组有序的指令序列称为程序。

利用计算机进行数据处理时,首先把要解决的实际问题,用计算机语言编写成程序,然后将待处理的数据和程序输入到计算机中,如图 1 -2 所示。计算机会按程序的要求,一步一步地进行各种运算,直到存入的整个程序执行完毕为止。

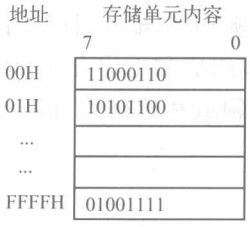

图 1 -1　内存的结构

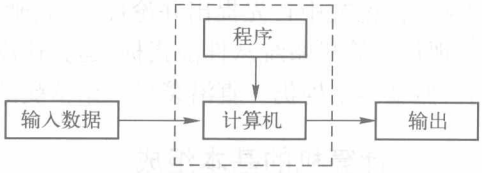

图 1 -2　程序的执行

1.1.2　计算机基本工作过程

计算机采用的是“程序存储和程序控制”的基本工作原理,它的工作过程是由其所存储并执行的程序控制的。简单地说,计算机的任务就是执行指令。

程序和指令执行的具体过程如下:

① 首先将待执行的程序装入计算机内存储器中,执行时,将程序在内存中的起始地址送入程序计数器;

② 控制器根据程序计数器中的地址,从存储器中取出第一条指令,送入指令寄存器;

③ 指令寄存器中的操作码部分经指令译码器译码,指令译码器将识别出的操作性质送入操作命令产生部件;

④ 操作命令产生部件按一定顺序发出一系列控制命令信号,送到各有关部件,使各部件完成指令所规定的操作。

因此,一条指令执行的全过程,大致可以分为 3 个阶段:取指令、分析指令和执行指令,如图 1 -3 所示。

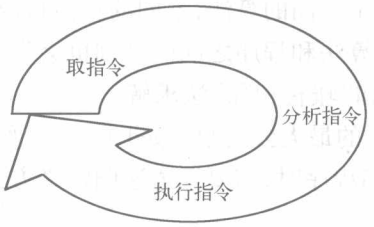

图1-3 指令执行周期

【例1-1】 指令“MOV AL, X”的工作过程。

(1) 取指令并分析指令

在取指令机器周期内,取出该条指令“MOV AL,X”的机器码送入指令寄存器中,该指令的操作码部分经指令译码器分析产生传送操作的信号,“告诉”微操作控制部件本指令将要执行传送操作。与此同时,指令寄存器中的寻址方式和形式地址部分经地址形成器,计算出源操作数的物理地址1FC0,如图1-4所示,目标操作数是内部寄存器AL。在取指令机器周期内还会更新程序计数器的内容,为执行下一条指令作好准备。

变量名	地址	内容
X	1FC0	6

图1-4 变量存储情况

(2) 执行指令

微操作控制部件接收到来自指令译码器的译码信号“取数和传递”,则转入执行“存储器读机器周期”。在该周期内将完成从相应的地址单元1FC0中取出X的值6,并送入寄存器AL中,该条指令执行完毕,转入执行下一条指令。

1.2 程序设计

1.2.1 程序设计的概念

简单地说,程序设计就是编写程序的过程。但程序设计本身并不简单,它是一种高智力的活动,不同的人对同一实际问题的处理可以设计出完全不同的程序。

【例1-2】 求两个正整数a和b的最大公约数。

【解法一】 采用欧几里德“辗转相除”计算方法,其核心思想是:以较大的数除以较小的数,余数与较小的数构成新的两个数,重复同样的操作直到大数被小数除尽,则较小数就是所求的最大公约数。

假设a>b,a除以b的余数表示为:a mod b。

求解步骤如下:

① 输入a、b;c是中间变量;

② 重复执行以下两步:若b=0,则输出a,停止;若b≠0,则c←a mod b,a←b,b←c。

有了上述步骤(算法),用程序设计语言很容易写出在计算机上执行的程序。若求解方法不是“辗转相除”法,而是我国古代数学家提出的“更相减损”法,那么,算法就会发生改变。不

难想象,不同的求解方法会产生出不同的算法,不同的算法将设计出不同的程序。但有时同一算法采用不同的程序设计技术,算法和程序之间的区别也会很大。

【解法二】 采用递归方法和“辗转相除”法求解。

假设使用过程 gcd 求 a 和 b 的最大公约数。gcd 是一个递归过程,所谓递归,就是用自身的结构来描述自身。在编写递归程序时,需要知道递归定义方式及递归终止的条件。

```
Procedure gcd(b,a:int,var c:int)
begin if b =0 then c←a
          a≥b & b≠0 then gcd(b,a mod b,c)
          a<b & b≠0 then gcd(a,b,c)
   endif
end
```

上述程序也是采用“辗转相除”算法,但是程序设计时采用了一定的变换,所以形式上发生了很大变化。

从这两个例子可以了解到:程序设计不仅与算法密切相关,它本身也是一种方法学。

1.2.2 算法

算法可理解为是对处理方法和步骤的描述。更精确地说,一个算法就是一个有穷规则的集合,其中的规则确定了求解某一特定类型问题的一个运算序列。这一运算序列应该具有以下 5 个重要特性。

- 有穷性:一个算法必须在执行有穷个计算步骤后终止。
- 确定性:一个算法给出的每个步骤必须都是精确定义的、无二义性的。
- 可行性:算法所要执行的每一个计算步骤都是可以在有限时间内完成的。
- 输入:一个算法一般要求有一个或多个输入信息,这些输入信息是算法所需的初始数据。
- 输出:一个算法一般要求有一个或多个输出信息,这些信息一般就是对输入信息计算的结果。

其中有穷性和可行性是算法最重要的两个特征。算法必须能够在有限步骤内终止,它的每一个步骤的执行顺序也是确定的,且能够在有限时间内完成。

算法的表示方法多种多样,既可用日常生活语言来表达,也可用流程图或用表述编程方法的伪代码来表示。

1. 自然语言

使用自然语言的算法描述,和日常生活的文字表达比较接近。例 1-2 中的“解法一”就是使用自然语言对算法进行描述的。下面是求 n! 的算法描述。

【例 1-3】 用自然语言表示求 n! 的算法。

① 输入正整数 n。

② 结果存放在变量 fac 中,fac 初始值为 1。

③ 重复执行以下两步:若 n = 1,则输出 fac,停止;若 n > 1,则 fac = n * fac,n 减 1。

④ 打印结果。

2. 流程图

流程图又称为程序框图，它是以图形的方式描述算法。传统的流程图由几种基本图形构成，如图 1 - 5 所示，通过流程线可以把各种框图连接起来，流程线的箭头指示程序执行的方向。使用流程图表示算法，形象直观，并便于理解。

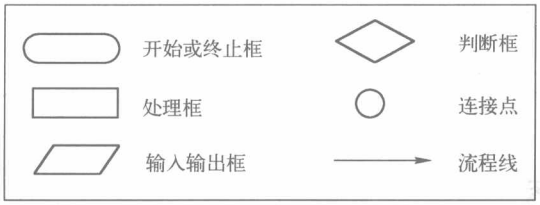

图 1 - 5　流程图的基本组成

【例 1-4】　用流程图表示求 n! 的算法。

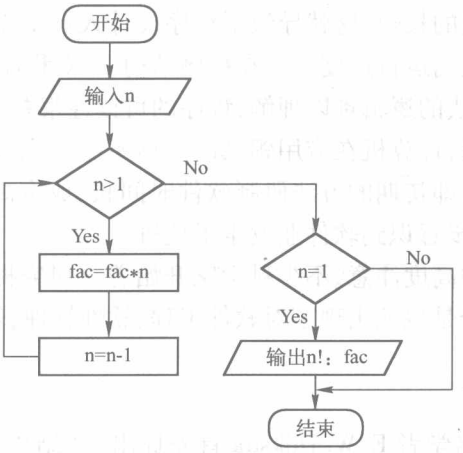

图 1 - 6　求 n! 的算法

3. 伪代码

伪代码(Pseudo Code)介于自然语言和某种编程语言之间，采用一种类似于程序设计语言的代码来描述算法，目的是使被描述的算法易于通过编程语言实现。因此，伪代码必须结构清晰、代码简单、可读性好。用于伪代码表示的编程语言通常有 Pascal，C 和 Java 等，称为类 Pascal 语言，类 C 和类 Java 语言。例 1 - 2 的“解法二”是使用类 Pascal 语言对算法进行描述的。

类 Java 语言则是采用一种类似于 Java 程序设计语言的代码来描述算法，其基本指令如表 1 - 1 所示。

表 1 - 1　Java 伪代码基本指令

基本指令	描　　述
赋值指令	助记符 ← 表达式
输出(输入)指令	输出(输入)(表达式)或 output(input) 表达式
循环指令	while(条件表达式)
	｛循环体｝
条件指令	if (条件表达式)｛指令序列 1｝
	else ｛指令序列 2｝

【例 1-5】 用伪代码表示求 n! 的算法。

```
input n
fac←1
while (n>1)
   { fac←fac * n
     n←n-1 }
output fac
```

1.2.3 程序设计方法

1. 初期的程序设计

计算机发展的初期,由于 CPU 运行速度慢、内存容量小,因此衡量程序质量优劣的标准是占用内存的大小和运行时间的长短,这就导致了程序设计人员不得不把很大的精力耗费在尽量减少程序的代码数量和提高运行速度上。在程序结构上,对程序的流程没有严格的限制,程序中可以随意跳转,导致算法的逻辑难以理解,程序的可读性很差。

随着计算机技术的发展,计算机在应用领域的发展不断扩大,软件的规模也越来越大,软件产品供不应求。传统方法即初期的方法研制软件时间长、成本高、可靠性低、难以修改和维护等问题日益突出,人们逐步意识到软件业发生了危机。

软件危机引起了人们的高度注意,不少科学家开始着手研究探讨产生软件危机的原因以及解决软件危机的途径。于是也就出现了对软件工程、软件管理、软件可靠性及程序设计方法等问题的研究。

2. 结构化程序设计

20 世纪 60 年代末,著名学者 E. W. Dijkstra 首先提出了“结构化程序设计”的思想。结构化程序设计要求程序设计者按照一定的结构形式来设计和编写程序,使程序易阅读、易理解、易修改和易维护。结构化程序设计的主要思想包括以下两方面的内容。

(1) 采用自顶向下、逐步细化的原则

按照这个原则,整个程序设计过程应分成若干层次,逐步加以解决;每一步都是在前一步的基础上对前一步的细化。这样,一个较复杂的大问题,就被层层分解成多个相对独立的、易于解决的小模块,有利于程序设计工作的分工和组织,也使调试工作比较容易进行。

(2) 程序由三种基本的控制结构组成

三种程序的基本结构包括:顺序结构、选择结构和循环结构。

顺序结构是指程序以语句出现的先后顺序控制执行。顺序结构是一种最简单、最基本的过程控制结构,是构成程序框架的基础部分。整个顺序结构只有一个入口和一个出口。

选择结构又称为分支结构,指程序能够根据条件判断是否执行某些语句,可以改变程序的执行流向。选择结构使程序变得智能化,它也是只有一个入口和一个出口。

循环结构可使程序根据需要重复执行某些语句,这些重复执行的语句称为循环体,用来判定是否继续重复的条件称为重复的终止条件。循环结构也是只有一个入口和一个出口。

上述三种基本结构都具有以下特点:

- 只有一个入口和一个出口;

- 结构内没有“死循环”,即无终止的循环;
- 结构中的每一个块都有被执行的可能,即每一个子结构都有一条从结构的入口到出口的路径通过。

按照结构化程序设计方法,程序由上述三种基本控制结构组成,这样的程序就是结构化程序。如果一个算法是用上述三种基本控制结构表示的,那么该算法就是结构化的算法。

结构化的算法具有如下的优点:

- 结构清晰,易读易理解;
- 易于修改、调试和维护,可按模块修改,而且只要模块间接口不变,模块内部细节可以任意修改,不影响整个结构化程序;
- 便于很多人协作编制大型软件;
- 便于保证和验证程序的正确性;
- 便于移植。

总之,采用结构化程序设计方法有利于提高软件生产率和质量。

3. 面向对象程序设计

面向对象的程序设计(Object Oriented Programming,OOP)起源于20世60年代中期的仿真程序设计语言Simula 67。20世纪80年代初,Smalltalk语言及其程序设计环境的出现成为面向对象技术发展的一个重要里程碑。目前面向对象方法已成为软件开发的主流方法。

通过面向过程的结构化程序设计技术使得程序更加规范,功能更加强化,但软件质量的问题仍未得到很好的解决。面向过程的问题求解能够精确地描述具体的解题过程,但却不足以把一个包含了多个相互关联过程的复杂系统表达清楚;在方法的实现中只突出了实现功能的操作方法(模块),而被操作的数据(变量)处于实现功能的从属地位,即程序模块和数据结构是松散地耦合在一起的。因此,当应用程序比较复杂时,容易出错,难以维护。

面向对象方法学的出发点和基本原则是尽可能模拟人类习惯的思维方式,使开发软件的方法与过程尽可能接近人类认识世界、解决问题的方法与过程。用面向对象的方法解决问题,不再将问题分解为过程,而是将问题分解为对象。将现实世界的任何事务均视为对象,认为客观世界是由许多不同类的对象构成的,每个对象都有自己内部状态、运行规律,不同对象之间的相互关系和相互作用构成了完整的客观世界。

面向对象方法具有以下4个要点:

① 客观世界任何事物都是对象,复杂的对象可由较简单的对象以某种方式组合而成;

② 每个对象都定义了一组数据和一组方法,数据和方法都被封装于对象之中,数据用于表示对象的静态属性,是对象的状态信息,而方法用于定义改变对象状态的各种操作;

③ 对象按其属性进行归类,类具有一定的结构,类可以有子类与父类;

④ 对象彼此之间仅能通过传递消息互相联系。

面向对象的程序设计具有如下的优点。

- 符合人们习惯的思维方法:由于对象对应于现实世界中的实体,因而可以很自然地按照现实世界中处理实体的方法来处理对象,软件开发者可以很方便地与问题提出者进行沟通和交流。
- 易于软件维护和功能增减:对象的封装性及对象之间的松散耦合,都给软件的修改和维护带来了方便。

- 可重用性好：重复使用一个类，可以比较方便地构造出软件系统，加上继承的方式，极大地提高了软件开发的效率。
- 与可视化技术相结合，改善了工作界面。

提示： 面向对象的程序设计并不是要抛弃结构化程序设计方法，而是通过不同的角度、不同的思维方法去解决程序设计方面的问题。当所要解决的问题被分解为低级代码模块时，仍需要结构化编程的方法和技巧，只是它分解一个大问题为小问题时采取的思路与结构化方法是不同的。结构化的分解突出过程，强调的是如何做、代码的功能如何完成；而面向对象的分解突出现实世界中具体的对象和思维中抽象的对象，强调的是做什么。面向对象的程序设计方法将大量的工作交给相应的对象来完成，程序员在应用程序中只需说明要求对象完成的任务。因此，学习面向对象的程序设计时，一方面要学习它分解问题的方法，另一方面还要重视对结构化程序的基本设计方法的掌握。

1.2.4 程序设计语言

用计算机能够接收的、指示计算机完成特定功能的命令序列称为程序；编写程序的过程称为程序设计；用于描述程序中操作过程的命令、规则的符号集合便称为程序设计语言。

1. 程序设计语言的四个发展阶段

计算机语言是一类面向计算机的人工语言，它是进行程序设计的工具。程序设计语言按其发展过程可划分为 4 个阶段。

第一阶段：机器语言（Machine Language）

机器语言是最初级的且依赖于硬件的计算机语言。用机器语言编写程序，程序人员必须熟悉机器指令的二进制符号代码，记忆指令代码能完成的操作，还应指出这一操作对象的位置，即记忆指令的操作码和地址码。

机器语言程序用二进制代码编写，计算机可以直接执行，但是由于它直接依赖于机器，所以不同型号的计算机，其机器语言是不同的，缺乏通用性。同时，机器语言不易记忆和理解，用机器语言编制程序的难度很大，编程效率比较低。

第二阶段：汇编语言（Assembly Language）

汇编语言用助记符号代替二进制编码，易于理解和记忆。用汇编语言编写的程序称为汇编语言程序，它有较直观、易理解等优点。但计算机却不能识别和直接运行汇编语言程序，必须由一种翻译程序将汇编语言程序翻译成机器语言程序后才能识别并运行，这种翻译程序即称为汇编程序。翻译产生的机器语言程序也称为目标程序。一个程序往往由多个汇编程序文件组成，需要用连接程序把这些块连接在一起，形成可执行程序，然后才能真正地调入内存运行。

用汇编语言编写程序与机器语言相比，运行效率基本相同，编程效率有所提高，但是，除较直观和易记忆外，仍然存在工作量大、面向机器、无通用性等缺点，所以汇编语言为“低级语言”，它仍然依赖于具体的机器。

第三阶段：高级语言（High-Level Language）

高级语言是一类面向问题的程序设计语言，且独立于计算机的硬件，其表达方式接近于被描述的问题，易于人们的理解和掌握。用高级语言编写程序，可简化程序编制和测试，其通用性和可移植性好。目前，计算机高级语言虽然很多，据统计已经有好几百种，但广泛应用的却

仅有十几种，它们有各自的特点和适用范围。常用的各种语言及其功能特点如表1-2所示。

表1-2 常用的各种语言及其功能特点

常用语言	功能特点
BASIC	适合初学者使用的普及型编程语言
FORTRAN	多用于科学及工程计算
COBOL	多用于商业事务处理和金融业
Pascal	有利于结构化程序设计，常用于教学
C	常用于系统软件的开发
PROLOG	多用于人工智能领域
C++	面向对象的程序设计语言，是以C语言为基础
Java	面向对象的常用于网络环境的程序设计语言

从机器语言、汇编语言到高级语言，程序设计越来越贴近人类的语言。

高级语言编写的源程序必须由系统翻译成计算机可以识别的指令代码，方可执行，如图1-7所示这个过程称为编译(Compiling)。根据编译方式的不同，高级语言可分为编译型和解释型两种。编译型高级语言源程序由编译器生成目标程序，再经过连接定位到内存，即可在操作系统下独立运行，但它具有平台依赖性，Pascal、C语言等高级语言均属于此类范畴；解释型高级语言源程序是由该语言的解释器逐条翻译逐条执行的，基本BASIC语言就是典型的解释性高级语言，Java语言则是一种新型的解释性高级语言。

图1-7 程序的翻译

第四阶段：面向对象语言(Object-Oriented Language)

面向对象语言是高级语言中的一种。面向对象作为一种思想及编程语言，为软件开发的整个过程提供了一个完整的解决方案。面向对象堪称是软件发展取得的里程碑式的伟大成就。

2. 程序的调试

调试程序是程序开发中必不可少的步骤。调试的目的是发现程序中的错误(Bug)。

发现错误的方法有两种，即理论法和实验法。理论法是属于程序正确性证明问题，它是利用数学方法证明程序的正确性，但目前仍处在研究之中。实验法是现在普遍使用的程序调试方法，而且卓有成效。下面所提到的调试都是指实验法。

程序调试的过程有两步：程序的语法(Syntax)调试和程序的逻辑(Logic)检查。通常程序的语法调试比较简单，只要熟悉编程语言，逐行调试，最终运行时没有出错警告，就算语法调试的完成。但这还远远不够，难的就在程序的逻辑检查。

在程序逻辑检查之前，需编造测试数据。测试数据除采用正常数据外，还应编造一些异常数据和错误数据，用来考验程序的正确性。错误数据可以用来试验程序对错误的处理能力，包括显示出错信息及容许修改错误的可能等。

对于大的程序来说，调试更加复杂。一般来说，大的程序是由多个功能模块组成的。调试