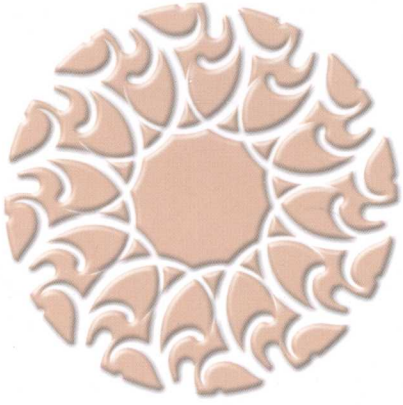
信息通信专业教材系列

Java高级语言程序设计

Java GAOJI YUYAN CHENGXU SHEJI

李 青 编著



北京邮电大学出版社
www.buptpress.com

信息通信专业教材系列

Java 高级语言程序设计

李 青 编著

北京邮电大学出版社
·北京·

内 容 简 介

本书以 Java 语言为载体，循序渐进地讲述了高级语言程序设计的基础知识和 Java 语言程序设计。全书分为 10 个主题，主题 1 到主题 2 介绍了程序语言的基本概念和 Java 程序语言的特征；主题 3 到主题 7 讲解了高级程序语言的基本语法和语句构成；主题 8 到主题 10 则深入学习 Java 面向对象编程技术。

本书在结构体例和写作方法上充分考虑了初学者的特点，在诸多环节上设计了各种学习指导，并且通过生动形象的比喻和接近口语化的叙述方式帮助初学者理解晦涩的术语和概念。

本书可作为远程教育和成人教育计算机专业各类高等院校非计算机专业高级语言程序设计课程和 Java 语言程序设计课程教材，也可供其他计算机程序语言的初学者使用。

图书在版编目(CIP)数据

Java 高级语言程序设计/李青编著. —北京：北京邮电大学出版社，2009.12
ISBN 978-7-5635-2118-0

Ⅰ.J… Ⅱ.李… Ⅲ.JAVA 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2009)第 184397 号

书　　名：Java 高级语言程序设计
作　　者：李　青
责任编辑：李欣一
出版发行：北京邮电大学出版社
社　　址：北京市海淀区西土城路 10 号(邮编：100876)
发 行 部：电话：010-62282185　传真：010-62283578
E-mail：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：北京忠信诚胶印厂
开　　本：787 mm×960 mm　1/16
印　　张：15
字　　数：317 千字
印　　数：1—3 000 册
版　　次：2009 年 12 月第 1 版　2009 年 12 月第 1 次印刷

ISBN 978-7-5635-2118-0　　定　价：28.00 元

· 如有印装质量问题，请与北京邮电大学出版社发行部联系 ·

前　言

高级语言程序设计一直是计算机科学与技术专业的基础课程之一，也是该专业的核心课程。该课程经过长期的演进和发展，已经成为一门较为成熟的课程。本课程的早期使用C语言讲解，但最近几年来Java语言逐渐成为主要的编程语言，很多高校开设了Java编程语言相关的课程。这几年的教学实践表明，Java程序语言课程和以C语言为基础的高级语言程序设计课程在内容上有所重复。考虑到在以往的“高级语言程序设计”课程中主要讲关键字、标识符、数据类型、变量与常量、数组、操作符与表达式以及程序控制语句等内容，而这些高级语言程序设计的概念和技能完全可以用Java语言讲解，我们决定把Java语言的基础部分放到“高级程序设计语言”课程中讲解，这样既让学生学习了高级语言程序设计的基础知识，同时也熟悉了Java程序语言的基本语法，为后续学习更深层次的Java课程奠定基础。本课程从完整性角度考虑，在讲解高级语言共有的编程基础知识的基础上也介绍了Java面向对象编程的基本技术，可以作为完整的Java程序语言课程学习。

本书分为10个主题，每个主题中按照内容的多少分为2～6个模块，每个模块包含一个或数个完整的知识点。主题1到主题2介绍了程序设计语言的基本概念和Java程序语言的基础知识；主题3到主题7以Java语言为例介绍了高级程序语言的语法和语句构成；主题8到主题10则是关于Java面向对象编程的内容，深入浅出地介绍了对象、类、继承等概念。

本书是专门为网络教育的成人学习者和计算机语言的初学者

编写的教材，在编写过程中，作者针对远程教育学生基础不同，选取适合的教学内容，在充分了解从业人员实际需求的基础上，设置必修和选修内容，区分重点知识点和非重点知识点。在进行知识点划分时，引入“模块化”和“微型学习”的理念，精心设计每一个知识点模块，将知识点的阅读学习时间尽量控制在30分钟左右。此外，本书对已有教学资源进行整理和综合，提供了丰富多彩的多媒体课件*，并按相关资源建设标准建立了题量丰富的试题库*，可满足学生自测、单元测验、作业练习及模拟考试等要求。课程在每个主题的后面都设有练习题*。练习题涵盖本知识点的最基本内容，能够让学生及时掌握刚刚学习过的课程内容；综合练习题则用于学生在学完本书后的自我评价。

衷心感谢北京邮电大学网络教育学院教育技术学专业的研究生马楠、王瑞娥、胡延芳等同学在资料收集、素材创作和文字校对等工作中提供的帮助和支持。衷心感谢北京邮电大学高大永、张志青、王晓军等老师提出的宝贵建议和支持。

作　者

注：标*的内容需要本课程教学网站的配合，请访问北京邮电大学网络教育学院网站（http://jx2.buptnu.com.cn）。此外正文中的动画和演示视频可在该课程网站观看。

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

注：带 * 内容为选学内容。

此为试读，需要完整PDF请访问：www.ertongbook.com

主题 1

Java 程序语言

内容提要

本主题的第一部分介绍计算机程序和程序语言的基础知识；第二部分介绍 Java 程序语言的基本特征和 Java 程序的工作原理。作为本书的引入部分，本主题的作用是帮助读者了解计算机程序的基本知识和 Java 语言的基本特征及其工作原理，这些内容对于后面 Java 语言的学习很重要，因此必须掌握。

学习目标

学完本主题以后，你应该能够：

- 解释程序、语句、面向对象、面向过程等基本概念；
- 复述程序的工作原理、Java 程序的工作原理；
- 列举计算机程序语言的发展阶段，列举 Java 程序语言的主要特征。

重点难点

- 解释语言和编译语言的差异；
- 面向过程编程和面向对象编程的差异；
- Java 程序的工作原理。

学习内容

模块 1.1 程序和程序语言（建议学习时间：30 分钟）

计算机程序的基本知识、编程语言的发展历史和程序的工作原理。

模块 1.2 Java 程序语言（建议学习时间：30 分钟）

Java 语言的历史、Java 语言的特点和工作原理。

模块 1.1 程序和程序语言

1.1.1 什么是计算机程序

计算机程序也叫软件，它告诉计算机该做什么。计算机执行的任何操作(从启动到关机)都是由程序控制的。常见的计算机程序有：

- Windows XP、Ubuntu Linux、Mac OS X 之类的操作系统；
- IE、Firefox、Opera 等用于浏览网页的浏览器；
- MS Office 等办公软件和同花顺等金融软件；
- QQ、MSN、网易泡泡等聊天软件；
- 星际争霸、帝国时代等游戏；
- 各种计算机病毒；
- 其他各类软件。

在日常生活中，我们常遇到这样的情况：父母、老师或者同学需要你帮他们做事情。例如在某个周末，你回到家，看到家人留下的便条：

我有急事出差两天，这两天你要做的事情有：

- 收拾屋子
- 去超市买两大包卫生纸
- 把筐里的脏衣服洗掉
- 交电费

上述对要做的事情的安排和描述相当于给你的指令，告诉你该做什么以及如何做。在完成以上任务的过程中，我们不但可以灵活地安排任务的先后顺序，例如，可以先买菜，也可以先收拾屋子，而且在做某件事情的时候也可以灵活处理，例如，去买菜也一定是根据自己的需要和市场里面供应的品种购买。

然而，计算机却没有这么灵活，它们严格地按命令执行。**程序**由一系列命令组成，程序运行时，计算机按特定顺序执行这些命令，每一行就是语句。计算机按顺序处理和执行程序的做法就像厨师按照菜谱的顺序炒菜一样。每个程序都必须精确地执行，每次只能执行一条语句。

下面让我们以典型的 Java 程序为例，看看程序是如何表述任务步骤的。表 1.1 中的左栏是 Java 程序片段，右栏是对程序的解读。根据这个表格，我

们可以大概了解每行语句的意思和整个程序片段要完成的任务。

表 1.1　Java 程序片段-1

if(isGameOver == true){
　　System.out.println("游戏结束!");
}else{
　　System.out.println("请重新再玩一次");
}
	如果游戏已经结束
	　显示:游戏结束
	否则
	　显示:请重新再玩一次

又如下例所示(见表 1.2)。

表 1.2　Java 程序片段-2

public class test{
　　public static void main(String[ ] args){
　　int i = 10, j = 3;
　　System.out.println(i/j);
　　System.out.println(i + j);
　　System.out.println(i - j);
　　}
}
	
	将 10 赋值给 i,3 赋值给 j
	程序执行 10÷3 后,输出:3
	程序执行 10+3 后,输出:13
	程序执行 10−3 后,输出:7

计算机完全是按照预先编写的命令执行任务的,因此除非是计算机硬件出现故障或者是由于计算机病毒攻击了系统而造成错误,在程序运行过程中发生任何错误,都是程序员的问题,因为计算机只是按照其所写的程序执行操作。所以,我们每次写程序时都应该好好地检查程序的设计是否有问题。不过值得庆幸的是,现在的计算机系统足够强壮,我们可以大胆尝试,而不用担心编程错误会搞坏计算机硬件。

1.1.2　编程语言的发展历程

人类语言的发展是一个渐变的过程,直至今天仍在不断改进。同样,计算机程序语言也不是一步到位,而是经历了一个从面向机器的语言,到面向过程的语言,再到今天的面向对象的语言的发展过程。

编程语言的诞生年代。

电子计算机所使用的是由"0"和"1"组成的二进制数,二进制是计算机的语言的基础。计算机发明之初,人们只能降贵屈尊,用一串串由"0"和"1"组成的各种指令操纵计算机,这种语言就是**机器语言**。使用机器语言是十分痛苦的,在程序有错需要修改时更是如此。而且,由于每台计算机的指令

系统往往各不相同，在一台计算机上执行的程序，如果要移植到另一台计算机上必须另编程序，这就造成了重复浪费。但由于使用的机器语言是针对特定型号计算机开发的，故而运算效率是所有语言中最高的。

为了减轻直接使用机器语言编程的痛苦，科学家们发明了一种新语言，用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这类程序设计语言就称为**汇编语言**。相比机器语言，汇编语言的可读性有所增强，但是仍旧难以读懂和编写。

在汇编语言的基础上，科学家们意识到，应该设计一种接近于数学语言或人类自然语言的计算机语言，这种语言又必须不依赖于计算机硬件，编出的程序能在不同的计算机上通用，这类语言后来被称为**高级语言**。经过多年的努力，他们发明了 FORTRAN 语言，随后又根据应用的领域和设计思想的不同，陆续发明了 C 语言、PASCAL 语言和 BASIC 语言等。这类语言关注做事情的先后顺序，因而被称为面向过程的程序语言。它们用接近数学语言的方式进行程序设计，加快了编程速度，也使得人们能够把注意力从烦琐的硬件细节转向算法本身。

【提示】
编写程序的过程实际上就是针对现实世界中的问题寻找解决方法的过程，这个过程需要用程序语言抽象化地描述客观世界问题，并且给出解法。

现实世界是复杂多变的，往往不能简单归结于先后顺序和数学模型。以数学逻辑为基础的**面向过程**的程序语言也有其局限性。因此，科学家们又陆续发明了 SmallTalk、LISP、Java、C＋＋等**面向对象**的程序语言。面向对象的程序语言解决了传统结构化方法中现实世界（需要解决的问题）和程序世界（问题的解决方案）在结构上不一致的问题，避免了从分析和设计到软件模块结构间的多次转换过程，使软件开发变得简单、高效、合理，是真正最接近人类思维方式的计算机程序设计语言。两者之间的差异见图 1.1。

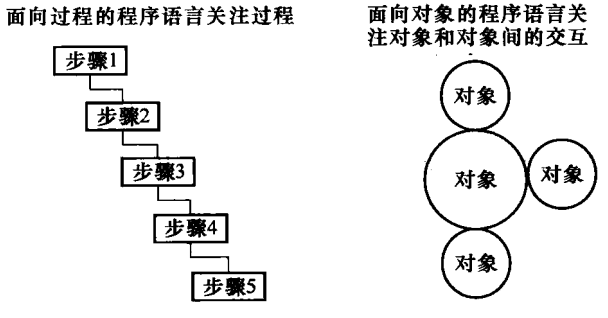


图 1.1　面向过程和面向对象对比

学习编程的关键是选择正确的编程语言。新语言的出现不代表旧语言的没落，在有上百种高级语言的今天，汇编语言在单片机编程等方面还是被广泛使用。不同的语言适合于不同的领域和任务要求，任何编程语言都有其最适合完成的工作，也有最不适合用来完成的工作。我们需要依据应用领域和要求计算机完成的任务来选择编程语言。

想一想

1. 最早的程序语言是_____。
2. Java 是一种面向_____的程序语言。
3. 选择编程语言的原则是_____。

参考答案：
1. 机器语言
2. 对象
3. 根据应用领域和任务要求选择

1.1.3　程序的工作原理

前面我们提到，计算机能够从程序中读取命令，并执行，因此操作计算机的第一步是写程序。写程序就要用到编程工具，编程工具和程序的关系就像 Word 和用 Word 写的文章的关系：我们用 Word 写文章，那么 Word 软件就像是编程工具，而文章就是编写出的程序。编写程序时，程序员用高级语言编写的程序文件叫做**源程序**。然而，电子计算机系统只能够直接识别和执行机器代码，用高级语言编写的源程序必须转换成二进制的机器语言，计算机才能执行。这个转换通过特定的转译程序自动或手工完成。此时，源程序、转译程序和计算机的关系就像是英国人、翻译人员和中国人的关系：英国人和中国人的语言不通，这时需要一位翻译人员在中间进行翻译解释。源程序和计算机之间也一样，计算机不能直接明白源程序的意思，此时的转译程序就像是这位翻译人员，读取程序中的每条语句，然后告诉计算机做什么。

【提示】
绝大多数的程序语言的源程序都是纯文本格式的，我们可以用文本编辑器，如记事本、写字板甚至是 Word 打开编辑，但是在再次保存的时候一定要将其保存为纯文本格式。

源程序转译成机器语言程序的实现方式有两种：编译方式和解释方式。**编译方式**是将用高级语言编写的源程序中的全部语句转换成机器语言，并把结果保存起来以备使用。以后当需要执行程序时，就直接执行转换后的结果。**解释方式**是按照源程序语句的顺序逐条进行转换。转换一条后，立

即执行，然后再转换下一条，再执行。如此重复，直到程序结束。解释方式不存储转换后的机器语言。由此可见，对于编译方式，只需编译一次，就可以多次执行机器码，因此执行速度快。对于解释方式，每次执行时都要把源程序的语句逐条转换成机器码，然后再执行，所以执行速度慢。但是解释方式也有其优点：对于修改后的源程序，不需要重新编译。两者的工作原理对比见表 1.3。

表 1.3　解释型和编译型编程语言工作原理对比

解释型程序语言	编译型程序语言
使用编辑器→编写源程序→解释执行	使用编辑器→编写源程序→编译→获得二进制代码→执行二进制代码

有些语言采用编译方式，如 PASCAL、C/C++等；而另外一些语言采用解释方式，如早期的 BASIC 语言；还有一些编程语言兼有两者的特点，如 Java。解释方式和编译方式各有优缺点，如表 1.4 所示。

表 1.4　解释方式和编译方式优缺点比较

	优　点	缺　点
解释方式	使用解释型语言编写程序时，可以立即进行测试，找出错误并修正，然后再试	运行速度比编译方式慢
编译方式	尽可能地对程序进行优化，使其高效运行。编译后的程序不需要解释器就可直接运行，且运行速度比解释型快	测试起来需要更多的时间。在测试前，需要编写并编译程序。发现错误并修改后，必须重新编译，以确定错误是否已消除

想一想

1. 程序的执行方式分为_____和_____。
2. 解释型语言的源程序必须通过_____来运行。
3. 编译型语言的源程序可以直接运行。（正确　错误）

参考答案：
1. 解释执行　编译执行
2. 解释器
3. 错误

此为试读，需要完整PDF请访问：www.ertongbook.com