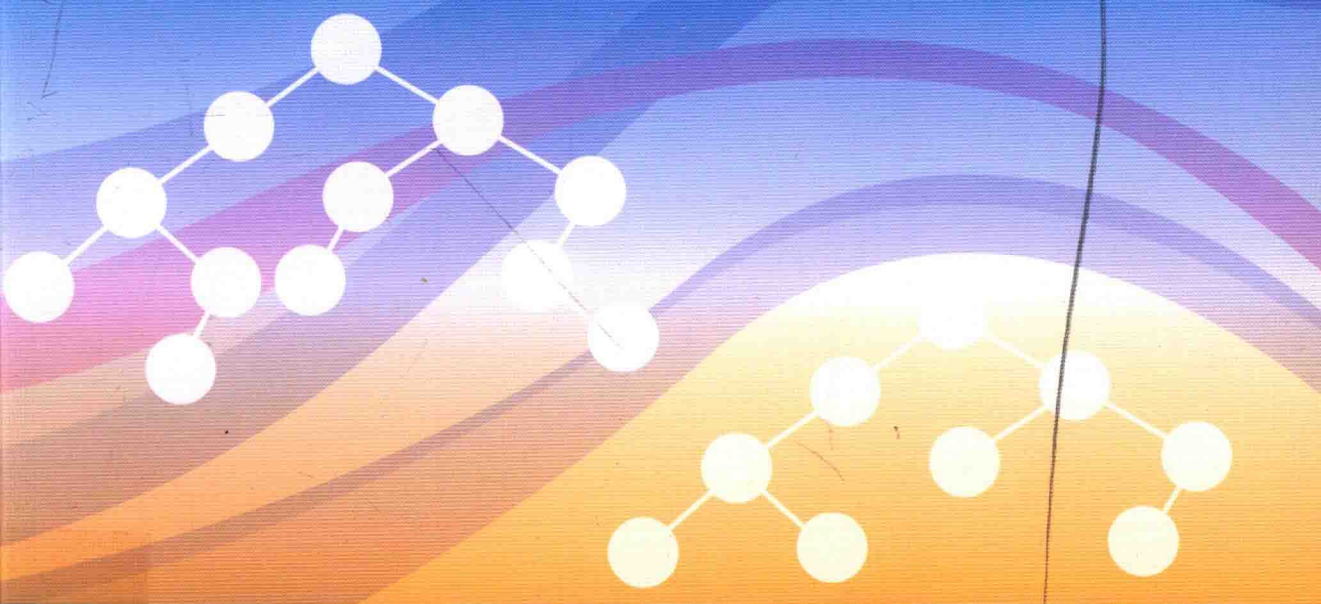
高等学校计算机专业规划教材

Java语言程序设计



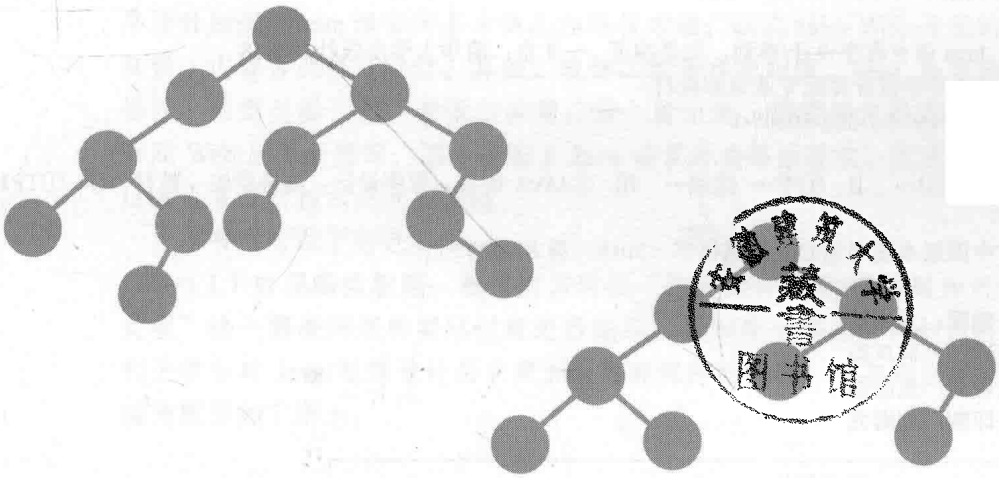
李莉 宋晏 编著

清华大学出版社

高等学校计算机专业规划教材

Java语言程序设计

李莉 宋晏 编著



清华大学出版社
北京

内 容 简 介

本书以 Java SE 7 为基础，从程序设计基础知识入手，由浅入深、循序渐进地介绍 Java 语言的基本概念、理论知识、程序设计方法及部分企业级应用技术。

全书共 11 章，第 1 章为预备知识，简要介绍程序设计、算法、软件工程的基础知识；第 2 章介绍 Java 语言的概况、程序结构和程序开发过程；第 3、4 章介绍 Java 的数据表示、运算和处理，包括数据类型、数据表示形式（常量、变量和字面量）、运算符及表达式、流程控制等内容；第 5、6 章为面向对象的编程知识，介绍面向对象的基本思想、Java 的类、包、常用类的使用、继承、多态和接口等重要概念；第 7～10 章为 Java 编程的常用知识，包括异常处理、输入输出、GUI 程序设计和集合框架等；第 11 章简要介绍 Java Web 应用程序开发，是对以上各部分知识的综合应用。

本书内容详尽、条理清晰，书中内容由浅入深、前后呼应，注重培养问题分析和求解的实际能力。书中示例丰富，所有示例均在 JDK1.7.0_79＋Eclipse Mars Release (4.5.0)环境下测试通过。

本书可作为高等院校 Java 程序设计类课程的教材，也可供广大工程技术人员和程序设计爱好者自学。

图书在版编目（CIP）数据

Java 语言程序设计/李莉，宋晏编著. —北京：清华大学出版社，2018
（高等学校计算机专业规划教材）
ISBN 978-7-302-50307-1

Ⅰ. ①J…　Ⅱ. ①李… ②宋…　Ⅲ. ①JAVA 语言－程序设计－高等学校－教材　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 101439 号

责任编辑： 龙启铭
封面设计： 何凤霞
责任校对： 焦丽丽
责任印制： 刘海龙

出版发行： 清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62795954
印 装 者： 三河市铭诚印务有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 30.75　　**字　　数：** 714 千字
版　　次： 2018 年 8 月第 1 版　　**印　　次：** 2018 年 8 月第 1 次印刷
定　　价： 59.00 元

产品编号：079474-01

前言

Java 语言是一种典型的面向对象的、跨平台的、支持分布式和多线程的优秀编程语言，具有极强的扩展性。自其诞生以来，迅速被业界认可并广泛应用于 Web 应用程序的开发中。在此形势下，国内高校在计算机及相关专业广泛开设了 Java 程序设计相关课程，旨在培养学生的编程能力，提高学生使用 Java 语言解决实际问题的能力，使学生建立良好的程序设计思想和编程习惯。本书正是基于此目的，结合 Java 语言学习的实际需要和作者多年的实践教学经验而编写的。

本书的内容编排遵循由浅入深、循序渐进的基本原则，以“数据如何表示/存储到如何计算/处理”为主线，从程序设计基础入手，详细介绍了程序设计知识、Java 语言的基本概念和编程方法，以及 Java Web 开发的基础知识，内容涉及程序设计、算法、软件工程等相关知识、Java 语言的基本语法、数据类型、类、继承、异常、输入输出流、图形用户界面设计、集合及 Web 应用开发等，基本覆盖了 Java 语言的大部分技术，是进一步使用 Java 语言进行技术开发的基础。

本书具有以下特色。

（1）内容编排新颖。教材内容围绕“数据如何表示/存储到如何运算/处理”这一解决问题的实际过程进行编排，更加符合学生的认知过程，有利于学生对 Java 程序设计形成更加全面和深刻的认识。全书的主要结构和编排顺序如下所示。

数据的表示/存储 →

<table>
<tr><td rowspan="3">基本数据类型</td><td colspan="4">引用类型</td></tr>
<tr><td colspan="2">类</td><td rowspan="2">接口</td><td rowspan="2">数组</td></tr>
<tr><td>自定义类</td><td>系统定义类</td></tr>
<tr><td colspan="3">OOP(类)、继承和多态</td><td>Java 集合</td></tr>
</table>

数据的运算/处理 →

<table>
<tr><td>用运算符/表达式处理</td><td>调用类/对象的方法处理</td><td>用代码段处理</td></tr>
<tr><td colspan="3">流程控制结构</td></tr>
<tr><td>数据的输入输出</td><td>数据可视化显示及控制</td><td>异常的处理</td></tr>
<tr><td>输入输出处理</td><td>GUI 程序设计</td><td>Java 异常处理</td></tr>
</table>

（2）重思路、轻语法。本书注重培养学生的程序设计思路。书中添加了程序设计基础知识，包括算法、软件工程等内容，并将常用算法通过案例融合到教材内容中，使学生掌握问题求解策略和算法设计的基本思路，能够独立完成常用算法设计/系统设计、程序编写与调试，提高编程能力。书中的例题均配有流程图或解题思路。

（3）内容有机整合。本书专门设置了“Java Web 应用开发”一章，通过 Web 应用案例的设计和开发过程，将类与继承、流程控制、异常处理、输入输出、集合类等各部分知识有机地整合起来，使本书知识更成体系，更容易使学生建立起整体知识架构，也为学生后期从事 Java 相关的开发奠定基础。

（4）注重代码规范。代码规范性是学生在初始学习编程技术时非常容易忽略的部分。良好的编码规范性是提高代码可读性、可维护性的重要基础。本书在示例中严格遵循代码规范，在 2.2.6 节专门介绍了官方发布的编程开发规范，并将该规范渗透至各相关部分的介绍中，希望读者在初始编程时就养成良好的编码习惯。

本书第 1～7 和第 9 章由李莉编写，第 8、第 10 和第 11 章由宋晏编写，全书由李莉负责审核和统稿。

感谢各位审稿专家对本书的编排提出的宝贵意见。本书的编写得到了北京科技大学教材建设经费的资助，在此一并谢过。

由于编者水平有限，书中难免有疏漏之处，敬请广大读者批评指正。

编　者

2018.4

目录

第 1 章 程序设计概述

本章主要介绍程序设计的相关知识，包括程序设计技术、算法和软件工程的基础知识。通过本章的学习，掌握程序设计的基本概念，理解程序设计方法，理解算法的描述和评价指标，了解软件工程的基本概念和软件开发过程，为进一步学习后续知识打好基础。

1.1 程序设计基础

1.1.1 程序的相关概念

1. 程序的概念

随着计算机的出现和普及，“程序”几乎成了计算机领域中耳熟能详的名词。人们要利用计算机完成各种预定的工作，必须先把完成该项工作所需要的步骤编写成计算机可以执行的指令序列。计算机程序就是为解决特定问题而利用计算机语言编写的指令序列的集合。

【例 1-1】 用 Java 语言编写程序，输入圆半径，计算圆的面积和周长。

例 1-1 Circle.java

```
import java.util.Scanner;
public class Circle {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        double r, perimeter, area;
        r = scan.nextDouble();
        perimeter = 2 * Math.PI * r;
        area = Math.PI * r * r;
        System.out.println("perimeter = " + perimeter + ", area = " + area);
    }
}
```

从以上程序中可以看到，一个计算机程序主要描述两部分内容：

（1）描述待解决问题中涉及的对象或数据，即数据结构的内容。数据结构是加工处理的对象，要设计一个好的程序，就需要将这些数据对象组织成合适的数据结构。

（2）描述处理这些数据的方法、过程或步骤，即求解的算法。算法是程序的灵魂，在程序编制、软件开发乃至整个计算机科学中都占有重要地位。

一个计算机程序必须对问题的每个对象和处理过程给出正确详尽的描述，即数据结构与算法是计算机程序的两个重要内容，针对问题要处理的对象设计合理的数据结构，可以有效地简化算法。可以说，程序就是在数据的特定表示方式以及结构的基础上，使用某种程序设计语言对算法的具体实现。

2. 程序设计的概念

编制程序的工作称为程序设计，包括分析需要解决的问题，设计解决问题的算法，应用某种程序设计语言编写算法代码等过程。

程序设计过程一般包含以下几个步骤，如图 1-1 所示。

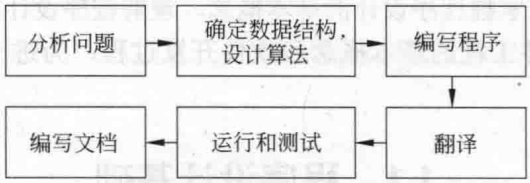


图 1-1　程序设计的过程

（1）分析问题，将实际问题抽象成一个计算机可以处理的模型。

- 要解决问题的目标是什么？
- 问题的输入是什么？已知什么？未知什么？使用什么格式？
- 问题的输出是什么？需要什么类型的信息、报告或图表？
- 数据的具体处理过程和要求是什么？

（2）确定数据结构，设计算法。

在分析问题的基础上，针对问题涉及的数据设计合适的数据结构，将数据的具体处理过程用算法进行描述，但算法不是计算机可以直接执行的程序，只是编制程序代码前对处理思想的一种描述。

当要处理的问题比较复杂时，可以将复杂问题分解成一些容易解决的子问题，每个子问题作为一个功能模块。这些模块组织在一起来解决整个问题。

（3）编写程序。

选择某种程序设计语言来对算法进行实现。

（4）翻译。

通常用高级程序设计语言编写的源程序并不能直接被计算机执行，必须经过翻译转换为目标代码方可在机器上直接执行。高级语言的翻译分为编译和解释两种。编译之后的目标代码经过链接过程后生成可执行程序，可以直接运行。解释则是直接边翻译边运行，对源程序逐行解释成特定平台的机器码并且立即执行，不会生成目标代码。

（5）运行和测试。

为保证程序的正确性，必须对已编制好的程序进行运行和测试。测试的目的是找出程序中的错误，使得程序尽可能完善。

（6）编写文档。

文档相当于一个产品说明书，对程序的使用、维护、更新都很重要。

3. 程序设计语言

程序设计语言是为了书写计算机程序而人为设计的符号语言，用于对计算过程进行描述、组织和推导。程序设计语言的广泛使用始于 1957 年出现的 FORTRAN 语言，其发展是一个不断演化的过程，根本的推动力是更高的抽象机制以及对程序设计思想的更好支持。

计算机的硬件只能识别由 0、1 组成的机器指令序列，即机器指令程序，因此机器指令是最基本的计算机语言。但机器指令是特定计算机系统所固有的、面向机器的语言，所以用机器语言进行程序设计时，效率低，程序可读性很差，难以理解，难以修改和维护。

之后，人们使用容易记忆的助记符代替 0、1 序列来表示机器指令，例如，用 ADD 表示加法、SUB 表示减法等。用助记符表示的指令称为汇编指令，汇编指令的集合称为汇编语言。汇编语言与机器语言十分接近，其书写格式在很大程度上取决于特定计算机的机器指令，它仍然是面向机器的语言。机器语言和汇编语言统称为低级语言。

在此基础上，人们开发了功能更强、抽象级别更高的语言以支持程序设计，产生了面向各类应用的程序设计语言，称为高级语言。高级语言的表达方式更接近人们对求解过程或问题的描述方式，而且与具体的计算机指令系统无关，大大提高了程序设计的效率。常见的高级语言有 Java、C、C++、C#、Python 等。

4. 高级程序的执行过程

计算机只能理解由 0、1 序列构成的机器语言，因此高级程序语言在执行时需要翻译为机器指令。翻译方式有多种，基本方式为汇编、解释和编译。高级语言的执行过程如图 1-2 所示。

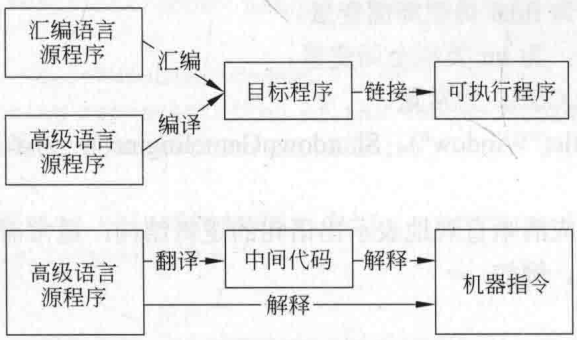


图 1-2　高级语言的执行过程

用某种高级语言或汇编语言编写的程序称为源程序。

如果源程序是用汇编语言编写的，则需要使用汇编程序将其翻译成目标程序后才能执行。

如果源程序是用某种高级语言编写的，则需要对应的解释程序或编译程序对其进行翻译，然后在机器上运行。

其中，编译程序（编译器）是将源程序一次性翻译成目标语言程序，然后在计算机上运行目标程序。解释程序（解释器）则对源程序逐行解释成机器指令并执行，或者将

源程序翻译成某种中间代码后再逐行地解释执行。

编译方式和解释的根本区别在于：在编译方式下，编译器将源程序翻译成单独保存的目标程序，机器上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，不会生成单独的目标程序，解释程序和源程序（或其中间代码表示形式）要参与到程序的运行过程中。

1.1.2 程序设计风格

程序设计风格（Programming Style）是指程序员在编制程序时所表现出来的特点、习惯、逻辑思路等。在程序设计中要使程序结构合理、清晰，形成良好的编程习惯，对程序的要求不仅是可以在机器上执行、给出正确的结果，而且要便于程序的调试和维护。

程序设计风格也是编程过程中使用的规则集合，主要体现在命名、语句、注释、输入输出等各个方面。

1. 命名

命名是程序设计风格中最重要的部分，也是初学者最容易忽略的部分。程序中的变量、常量、函数、方法、数组、类等各类数据都需要进行命名。命名应该满足以下要求：

（1）见词知义：命名应当直观，易于理解。例如：

- nextDouble()。
- positivePrefix、lineSeparator。
- arraycopy(Object src, int srcPos, Object dest, int destPos, int length)。

（2）表明身份：通过名字即可获知该数据的身份，即是变量、常量、函数，还是类。例如：

- fDeltaTime：为 float 类型局部变量。
- g_iGameState：为 int 类型全局变量。
- EXIT_ON_CLOSE：为常量。
- SetWindowTitle("Window")、ShutdownGameEngine()：为函数。

2. 语句

语句的书写形式应清晰直观地表示出语句的逻辑结构，通常需要合理地缩进、增加空格、空行或大括号。例如：

```
if ( a > b ) {
    t = a;          // 通过缩进表示这三条语句是 if 语句的内嵌语句
    a = b;
    b = t;
}
```

在语句构造方面，尽量简单直接，不能为了追求效率而使代码复杂化；为了便于阅读和理解，不要在一行内写出多条语句；不同层次的语句采用缩进形式，使程序的逻辑结构和功能特征更加清晰；要避免复杂的判定条件，避免多重的循环嵌套；表达式中使用括号以提高运算次序的清晰度等；避免使用具有二义性或很难理解的语句，少使用多用途的长表达式，以保证程序的良好可读性及代码的正确性。例如，以下表达式不建议使用：