



Java 经典教材译丛

Java 实例导学



THOMSON



TM

Gary B.Shelly

著者: [美] Thomas J.Cashman

Joy L.Starks

译者: 董庆霞 李雪非

北京大学出版社

<http://cbs.pku.edu.cn>

Java 经典教材译丛

Java 实例导学

Java Programming: Complete Concepts and Techniques

	Gary B.Shelly	
[美]	Thomas J.Cashman	著
	Joy L.Starks	
	董庆霞 李雪非	译

北京大学出版社

内 容 简 介

本书是 Java 程序设计的入门书籍, 重点介绍了面向对象程序设计的概念。通过编写现实中的案例, 详细地讲解 Java 程序设计语言的基本概念和技术, 例如, 循环、数组、字符串、排序、AWT、数据文件等。

本书适合初学 Java 语言的读者, 可作为大学计算机及相关专业的教材, 也可作为程序员的参考用书。

Original English language title: Java Programming: Complete Concepts and Techniques, by Gary B. Shelly

et al.

EISBN: 0-7895-6099-2

Copyright © 2001 by Course Technology, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All rights reserved. 本书原版由汤姆森学习出版集团出版。版权所有, 盗印必究。

Peking University Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SARs and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由汤姆森学习出版集团授权北京大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可, 不得以任何方式复制或发行本书的任何部分。

981-254-148-9

北京市版权局著作权合同登记号 图字: 01-2003-8802 号

图书在版编目(CIP)数据

Java 实例导学/(美)谢尔(Shelly,G.B.)等著;董庆霞等译.—北京:北京大学出版社,2004.7

(Java 经典教材译丛)

ISBN 7-301-07391-7

I.J... II.①谢... ②董... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 03 号

书 名: Java 实例导学

著作责任者: [美] Gary B. Shelly, Thomas J. Cashman, Joy L. Starks 著

译 者: 董庆霞 李雪非

责任编辑: 温丹丹

标准书号: ISBN 7-301-07391-7/TP·0760

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750581

电子信箱: xxjs@pup.pku.edu.cn

排 版 者: 北京东方人华北大彩印中心 电话: 62754190

印 刷 者: 河北涿县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 17 印张 408 千字

2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

定 价: 24.00 元

编 者 序

欢迎学习 Java 程序设计语言！

自计算机发明以来，无处不在显示着它对整个人类文明和社会进步产生着如此巨大、如此深刻的影响，推动着人类社会一日千里地向前发展

历经半个多世纪，计算机软件——计算机的重要组成部分，也发生了日新月异的变化。从机器语言到汇编语言再到高级语言，从结构化语言到面向对象语言，人们不断地探索和发明新的、功能强大又简单易学的计算机语言。

用计算机语言编程，有如我们用世界上的任何一种语言写作一样，两者都是创造性的工作，而且都要用到相应的技术。如果您掌握了一种语言，熟知它的词汇和语法规则，也许您能熟练地运用这种语言进行表达，但这并不能使您成为一位作家。写作是一种创造性的工作，它需要有创造性的人来做。用计算机语言编程也是同样的道理。也许您掌握了当今最流行的程序设计语言，而且也能迅速地写出程序语句，但是，程序的内容只能靠您的创造性来产生。

1. 为什么要学习 Java 程序设计语言？

工欲善其事，必先利其器。没有得心应手的工具又怎么能随心所欲地发挥创意？计算机高级语言众多，而 Java 语言的得天独厚的优点，使它从众多语言中脱颖而出，越来越受到人们的青睐。Java 是一种跨平台、适合于分布式计算环境的面向对象编程语言。具体来说，它具有如下特性：简单、面向对象、分布式、易解释、可靠、安全、与平台无关、体系结构中立、可移植、高性能、多线程、动态等。

鉴于此，我们从国外遴选了 8 本流行的 Java 书，组成了《Java 经典教材译丛》，以期读者能从中受益。

2. 您渴望从《Java 经典教材译丛》学到什么？

博大精深是我们选此套丛书的宗旨，从 Java 基本概念和技术到其高级主题，丛书的内容涵盖了 Java 的全部知识。原书多由国外教学经验丰富的教师编写，书中提供大量的实例和案例。为了给读者提供良好的服务，我们将大部分书中的实例中的源代码放在 <http://cbs.pku.edu.cn> 中的【下载专区】上，读者可从中下载。

3. 什么人使用《Java 经典教材译丛》？

无论您是一位计算机语言初学者，还是一位熟知计算机语言的编程人员，《Java 经典教材译丛》中必有一本适合您。当然，本套丛书主要针对在校大专院校的学生而编写，所以大多数读者都可以轻松上手，教师更可以从中找到教学用书和参考读物，部分书中带有星号(*)的章节则是为更高层次的读者而设计，为选学内容。

4. 对计算机的硬件和软件有什么需求?

- 软件 可以通过访问 Sun 的网站(<http://java.sun.com>)下载 JDK 编译程序, 下载的软件大约为 20MB。
- 硬件 必须具备可以运行 Windows 操作系统的计算机或者 Windows NT 工作站, 100MB 的空闲磁盘空间, 最少 32MB 的内存(如果使用 Windows NT, 建议准备 64MB 的内存)。

丛书编委会
2003 年 11 月

目 录

第 1 章 Java 程序设计简介.....	1
1.1 简介.....	1
1.2 什么是 Java.....	1
1.3 对计算机进行编程.....	3
1.4 程序开发生命周期.....	3
1.5 结构化的程序设计.....	5
1.6 控制结构.....	7
1.6.1 顺序控制结构.....	8
1.6.2 选择控制结构.....	8
1.6.3 重复控制结构.....	9
1.6.4 嵌套.....	10
1.7 面向对象的方法.....	11
1.7.1 对象术语.....	11
1.7.2 名词.....	11
1.7.3 动词.....	12
1.7.4 形容词.....	14
1.7.5 面向对象的同义词.....	15
1.8 OOP 的原理.....	15
1.9 封装、继承和多态.....	16
1.9.1 封装.....	16
1.9.2 继承.....	16
1.9.3 多态.....	17
1.10 快速应用程序开发和 OOP 的优点.....	17
1.11 小结.....	18
1.12 知识测试.....	18
1.13 知识应用.....	22
第 2 章 创建 Java 程序.....	24
2.1 简介.....	25
2.2 项目 1——Anita 古董店.....	25
2.3 桌面的设置.....	26
2.3.1 启动记事本.....	26
2.3.2 打开命令提示符窗口.....	27
2.4 编写程序代码.....	27

2.4.1 文档.....	27
2.4.2 访问修饰符和类名.....	28
2.4.3 方法和方法头.....	29
2.4.4 代码.....	29
2.5 测试程序.....	30
2.6 错误.....	31
2.6.1 语法错误.....	31
2.6.2 语义错误.....	32
2.6.3 运行时错误.....	32
2.6.4 运行程序.....	33
2.7 编辑源程序和再次编译.....	33
2.7.1 使用转义字符格式化输出结果.....	33
2.7.2 重新编译和重新运行程序.....	34
2.8 迁移到网上.....	34
2.8.1 引进需要的文件包.....	34
2.8.2 编辑小程序.....	36
2.8.3 paint 方法和 drawString 方法.....	37
2.8.4 返回到命令提示符窗口并编译小程序.....	38
2.9 小程序驻留的 HTML 文件.....	38
2.9.1 HTML 标记.....	38
2.9.2 使用小程序查看器运行小程序.....	39
2.10 小结.....	40
2.11 知识测试.....	41
2.12 知识应用.....	43
2.13 实验室.....	44
2.14 案例.....	46
第 3 章 使用方法操作数据.....	48
3.1 简介.....	48
3.2 项目 2——Bert 车行购车贷款指导程序.....	48
3.3 开始一个新的 Java 程序.....	49
3.4 存储数据.....	50
3.4.1 Java 数据类型.....	50
3.4.2 变量和标识符.....	51
3.4.3 变量的声明.....	52
3.5 使用样本数据.....	52
3.6 运算符.....	53
3.6.1 算术运算符.....	53
3.6.2 比较运算符.....	54

3.6.3 运算符的优先级.....	54
3.6.4 计算公式.....	55
3.7 结果输出.....	55
3.7.1 在输出结果的过程中使用变量.....	55
3.7.2 在输出结果的过程中使用的方法.....	56
3.8 保存、编译并运行文件.....	57
3.9 用户输入.....	58
3.9.1 从键盘输入.....	58
3.9.2 流和构造函数.....	58
3.9.3 实例化和构造函数.....	59
3.9.4 修改 Bert 程序.....	60
3.9.5 数据处理.....	60
3.10 运行交互的程序.....	64
3.11 迁移到网上.....	64
3.11.1 将程序转换为小程序.....	64
3.11.2 标签组件.....	65
3.11.3 文本框组件.....	65
3.11.4 按钮组件.....	65
3.11.5 创建小程序中的对象组件.....	66
3.11.6 init 方法.....	67
3.11.7 ActionListener.....	68
3.12 HTML 驻留文档和交互小程序.....	72
3.12.1 创建驻留文档.....	72
3.12.2 使用小程序查看器运行一个交互的小程序.....	72
3.13 文件管理.....	74
3.14 小结.....	74
3.15 知识测试.....	75
3.16 知识应用.....	77
3.17 实验室.....	79
3.18 案例.....	83
第 4 章 Java 中的判断、循环和组件.....	84
4.1 简介.....	85
4.2 项目 3——CandleLine 送货费用.....	86
4.3 开始设计新的 Java 程序.....	88
4.4 用户输入.....	88
4.5 组件的模块化.....	89
4.6 异常处理.....	90
4.7 测试部分程序.....	94

4.8 if 判断结构.....	97
4.9 创建方法.....	102
4.9.1 编写用户定义的方法.....	102
4.9.2 对输出结果进行编码.....	104
4.9.3 测试 getCost 方法.....	104
4.9.4 case 结构.....	105
4.9.5 switch 语句.....	105
4.9.6 测试 switch 结构.....	107
4.10 循环结构.....	108
4.10.1 while 语句.....	108
4.10.2 测试 while 结构.....	109
4.11 迁移到网上.....	112
4.11.1 创建驻留文档.....	112
4.11.2 创建小程序存根程序.....	112
4.12 在小程序中做出判断.....	114
4.12.1 创建复选框.....	115
4.12.2 向小程序添加组件、颜色和焦点.....	117
4.12.3 小程序中的判断.....	119
4.12.4 执行小程序.....	120
4.13 小结.....	123
4.14 知识测试.....	124
4.15 知识应用.....	126
4.16 实验室.....	129
4.17 案例.....	133
第 5 章 外部类、数组和 Choice 组件.....	135
5.1 简介.....	136
5.2 项目 4——Donna 的辩论赛得分.....	136
5.3 创建 Person 类.....	139
5.3.1 输入 Person 类的开始代码.....	139
5.3.2 创建实例.....	140
5.3.3 使用实例方法.....	142
5.4 数组.....	145
5.4.1 数组表示法.....	146
5.4.2 对象数组.....	147
5.4.3 创建具有数组的应用程序.....	148
5.5 可以测量的循环.....	149
5.5.1 for 循环.....	149
5.5.2 赋值和一元运算符.....	150

5.5.3 输入 for 循环代码	152
5.6 对数组进行排序	153
5.6.1 完成 Donna 应用程序	155
5.6.2 编译并运行应用程序	158
5.7 迁移到网上	159
5.7.1 创建驻留文档	160
5.7.2 编写小程序的开始部分代码	160
5.7.3 创建 Choice 组件	161
5.7.4 addItem 方法	163
5.7.5 actionPerformed 事件	165
5.7.6 编译 DonnaApplet	167
5.8 小结	169
5.9 知识测试	169
5.10 知识应用	172
5.11 实验室	175
5.11.1 编写外部方法	175
5.11.2 创建具有下拉列表框的小程序	175
5.11.3 对数组进行排序	177
5.12 案例	178
第 6 章 AWT 应用	180
6.1 简介	181
6.2 项目 5——Calculator 程序	181
6.3 AWT	182
6.3.1 容器	183
6.3.2 面板	184
6.4 布局管理器	186
6.4.1 FlowLayout	186
6.4.2 BorderLayout	187
6.4.3 GridLayout	188
6.4.4 CardLayout	189
6.4.5 GridBagLayout	189
6.4.6 创建 Calculator 的框架和布局	190
6.5 制作键盘	191
6.5.1 构建按钮并设定布局	191
6.5.2 将按钮放到面板中	192
6.5.3 在框架中加入组件	193
6.6 窗口方法	194
6.7 actionPerformed 方法	195

6.7.1 确定哪一个按钮	196
6.7.2 switch 结构	197
6.7.3 单击运算符按钮	197
6.7.4 对运算符按钮的操作	198
6.7.5 完成 actionPerformed 事件剩下部分的代码	199
6.8 为 main 方法编程	200
6.9 测试程序	201
6.10 迁移到网上	204
6.10.1 创建驻留文件	204
6.10.2 将应用程序转换为小程序	204
6.10.3 继承 Applet 包	204
6.10.4 对 Calculator 程序中的类做适当的改动	205
6.10.5 为小程序增加新的功能	207
6.10.6 测试【清屏】按钮	209
6.11 小结	209
6.12 知识测试	210
6.13 知识应用	212
6.14 实验室	213
6.15 案例	219
第 7 章 使用外部数据	221
7.1 简介	222
7.2 项目 6——Flora 市车船使用税	222
7.3 创建界面	224
7.4 创建界面组件	226
7.4.1 创建组件	226
7.4.2 设置布局管理器	228
7.4.3 将组件加入到框架中	229
7.5 外部数据	231
7.5.1 数据结构层次	231
7.5.2 顺序存取文件和随机存取文件	232
7.5.3 增添 ActionListener 并打开数据文件	234
7.5.4 向文件里写数据	235
7.5.5 增加一个清空输入内容的方法并返回焦点	236
7.5.6 测试按钮	237
7.6 错误提示	238
7.6.1 消息提示框	239
7.6.2 创建 MessageBox 类	239
7.6.3 调用 MessageBox 类	241

7.6.4 运行带有消息提示框的 Flora 应用程序	242
7.7 迁移到网上	244
7.8 小结	245
7.9 知识测试	246
7.10 知识应用	249
7.11 实验室	252
7.12 案例	256
附录 A 保留关键字	257
附录 B 选项和异常	258

第 1 章 Java 程序设计简介

案例背景

许多年来，Central 学院的计算机科学系在编程概念课程方面已经尝试了各种不同的程序设计语言。这门课程吸引了大批学生，包括主修和辅修计算机的学生，以及其他系中知道计算机技术已经变得非常重要的那些学生。当许多学生在这门课程中获得成功时，指导教师报告说，有些学生开始学习屏幕、图像和按钮等用户接口方面的高级课程，但是对于良好程序的底层结构却知之甚少。

计算机科学系已经要求学生在这个第一次开设的程序设计课程的试点部分进行选课，教授 Java 程序设计语言。计算机科学系希望 Java 结构、面向对象的特征和网络应用程序能够有助于程序设计概念的学习，并帮助学生掌握良好的初级阶段程序设计技巧。如果成功的话，下一个学期所有的程序设计概念介绍部分都将使用 Java 语言。大家自己决定是否选修此课程。

1.1 简介

为什么首先介绍 Java 程序设计语言？

- 初级程序员需要了解计算机程序设计的基本概念和创建良好程序的基本结构。Java 是结构化语言。
- 当发现自己的工作可以应用到现实世界时，初级程序员最有可能得到改善提高。Java 是网络应用程序所选择的语言。
- 初级程序员在直接关系到访问硬件和软件方面大都坚持始终如一，而 Java 是与平台无关的语言。
- 初级程序员希望学习适合在市场上销售的知识，而 Java 是面向对象的语言。

Java 是优秀的、多用途的程序设计语言。学校、公司，以及软件企业正逐渐意识到 Java 是非常适合市场需要的语言，而且 Java 为编写良好的计算机程序提供了所需的结构化基础。编写计算机程序的大多数公司也逐渐意识到面向对象的方法可以创建易于开发、调试和维护的程序。有时被程序设计语言的复杂性所吓倒的初学者，或者那些陶醉在图形用户接口的铃声和笛声中的编程人员，需要诸如 Java 等具有交互性结构的语言来培养良好的编程习惯。

1.2 什么是 Java

Java 是一种计算机编程语言。在计算机开始产生所需的结果之前，它必须一步步或者系统地描述将要完成的任务。计算机程序是告诉计算机需要做什么的一套指令集。对于多

达 2 000 多种的用来编写计算机程序的程序设计语言和工具来说,Java 只不过是一个新生儿。程序设计语言是一组字、符号和代码,用来创建计算机能够理解或识别的指令。高级语言(如 Java),使用命令和字代替隐晦的数字代码或内存区域地址,从而把数据处理成信息。高级语言中的每一条指令对应于计算机机器语言的多条指令。用来指定如何编写指令的特殊文法或规则集称为语言的语法。

Java 的历史

20 世纪 90 年代初期,James Gosling 领导的 Sun 公司的一个工作组设计了 Java 语言。Java 设计者以 C、C++ 和 Smalltalk 等语言的基本语法为起点开始设计 Java 语言。Java 工作组想开发一种简洁的、面向对象的语言。Java 语言最初应用在蜂窝电话等信息设备中,但是几年后,Sun 公司已使用 Java 为 WWW 提供动画和交互性。IBM 已经把 Java 采纳为主要的应用程序开发语言。

网络浏览器已经为运行 Java 小程序提供了时机,Java 小程序(applet)是迷你型程序,可以下载并作为所显示网页的一部分执行。这就使得 Java 成为网络应用程序可以选择的语言。JavaScript 与 Java 不同,它是 Netscape 和 Sun 公司创建的脚本工具,用来把代码语句直接插到网页的超文本标记语言(Hyper Text Markup Language, HTML)中,从而增加网页的功能并改善其外观。与 JavaScript 语句不同,Java 是嵌入到 HTML 文档中的,而小程序则是作为单独的文件发送到浏览器中,它与 HTML 文档是并列的。小程序的例子包括为网页增加交互性动画或游戏。

Java 是世界上发展最迅速的程序设计语言,部分原因是因为设计团队成功地使这种语言具有简洁性、健壮性、安全性和可移植性。计算机专业人员使用简洁性这个单词,意味着一种语言具有简洁的指令集,同一条命令没有太多的版本或修订版。健壮性意味着 Java 支持程序开发,不会意外地覆盖内存和毁坏数据,这使得 Java 非常适合于网络和分布式应用程序。例如,Java 在运行时会检查数组数据的访问情况,从而可以确保这样的访问不会超过数组的界限。Java 是一种强制类型检查语言,这意味着 Java 编译器会提供大量的编译时间,用来检查与数据类型有关的潜在问题。对于初级程序员来说,这非常有利。Java 是安全的,因为 Java 程序很容易防止病毒攻击和篡改。

如果计算机程序可以应用在不同于创建它的操作系统中,而且不需要经过较多的改写,那么这样的程序被认为具有可移植性。Java 是与平台无关的语言,这意味着 Java 在设计构造上是中立的。我们可以使用 Java 在具有任何操作系统的任何平台上编写程序,不管平台是 PC、苹果机、Unix 机器还是大型机。Java 编译器在 Sun 公司的 Java 开发工具包(Java Development Kit, JDK)中提供,可以把 Java 源代码转换为计算机可以阅读的对象代码(称为字节码)。相同的对象字节码可以在任何计算机上运行,只要计算机具有解释器来执行 Java 字节代码。用于 Java 程序设计语言的解释器称为 Java 虚拟机(Java Virtual Machine, JVM)。JVM 也在 JDK 中提供,其中包含解释器和运行时系统。运行时系统包含所有运行 Java 程序所需的文件和软件包。其他编程语言则需要使用系统特有的解释器和编译器。

本书中 Java 开发工具包将用来开发独立的程序,即程序可以独立于其他任何软件运

行。在操作系统的命令提示符下输入命令就可以执行 Java 程序，不需要外部用户接口。在操作系统的命令行中运行的程序称为在控制台模式下运行。到目前为止，已经开发了许多用户接口和集成开发环境(Integrated Development Environments, IDE)，用来帮助编写 Java 程序。但是，因为学校和企业可能会使用各种各样的用户接口和 IDE，因此在控制台模式下运行程序将使本书的内容具有可移植性。该模式也将利用真正的 Java 命令和类，而不是那些由接口创建的命令和类。在每个项目中，大家可以在控制台模式下运行程序，然后修改代码，从而使用 Java 小程序查看器执行程序。也可以用网络浏览器运行修改的版本。当通过项目不断进步时，读者将学到更多有关 Java 的网络能力和 IDE 的知识。

Sun 公司在 1999 年发布了 Java 2 SDK 平台。SDK 中包含编译器等开发工具和 Java 2 运行时环境的标准版本(J2SE 1.2 版本)。

1.3 对计算机进行编程

大多数计算机用户并不自己编写程序。字处理器或电子制表软件等通用商业和个人应用程序可以从软件商那里购买，或者从销售计算机产品的商店购买。这些购买的程序称为应用程序软件包。应用程序是告诉计算机如何接受终端用户指令，以及如何产生响应这些指令的信息程序。

即使好的应用程序不是很昂贵，人们也需要学习编程。对于和计算机相关的职业来说，学习程序设计语言可以改善逻辑和批判性的思维技巧，同时还可以学习为什么应用程序会如此执行命令。大公司需要行业特有的软件，这些软件由于有限的使用范围而不能在零售市场上获得。程序需要经常维护和监控。小公司希望能够调整和改编程序，以便适合自己的需要。生产应用程序软件的软件企业总是在寻找更多的程序员。随着硬件、互联网技术和因特网的进步和变革，需要人们能够迎接创建新型应用程序的挑战。程序设计是工程学和艺术的结合，是在市场上有销路的一种技能。

1.4 程序开发生命周期

程序员并不是立刻坐下来编写代码，相反他们遵循有组织的方案或方法论，即把过程分割成一系列的任务。就像存在许多程序设计语言一样，也存在许多应用程序开发方法论。但是不同的方法论往往是称为程序开发生命周期(Program Development Life Cycle, PDLC)的变体。PDLC 遵循以下 6 个步骤：(1)分析问题，(2)程序设计，(3)编写程序代码，(4)测试程序，(5)程序定型，(6)程序维护。表 1.1 描述了程序员实现计算机应用程序所经历的每个步骤。图 1.1 把 PDLC 描述为连续性的过程或循环。当维护阶段确认需改动之处时，循环重新开始。

表 1.1 PDLC 步骤

步 骤	过 程	描 述
1	分析问题	精确定义将要解决的问题并编写程序规范，即对程序输入、处理、输出和用户接口的描述
2	程序设计	通过伪代码、流程图、对象结构图或事件图等工具，使用算法思维开发详尽的逻辑方案，从而把程序行为组织成模块，并为每个模块设计解决方案或算法，同时测试方案算法
3	编写程序代码	通过创建用户接口并编写代码，使用编程语言或应用程序开发工具或 IDE，把设计翻译成应用程序，同时包括代码内部解释代码语句意图的内部文档(注释)
4	测试程序	测试程序，寻找并纠正错误(调试)，直到没有错误出现，并且包含足够的安全措施，确保获得预期的结果
5	程序定型	复查并且如果需要，修订内部文档，定型并完成终端用户(外部)文档，实现用户级别的解决方案
6	程序维护	向终端用户提供培训和支持，纠正任何非预期的、用户请求修改(增强)时出现的错误

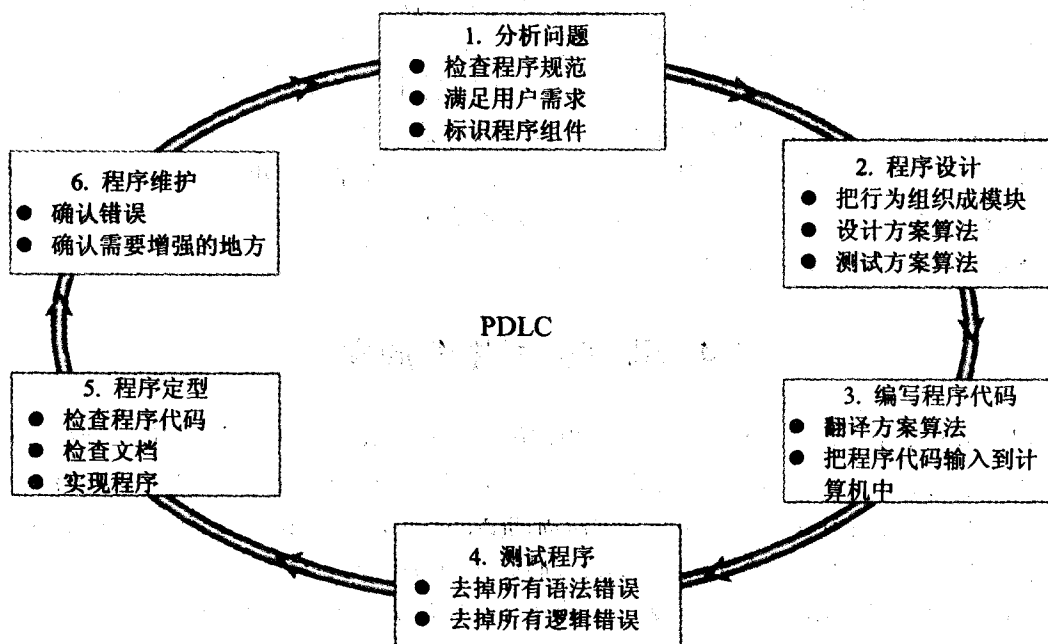


图 1.1

1.5 结构化的程序设计

Java 是一种结构化语言。结构化程序设计语言的概念是基于：这种语言可以把大的编程任务分解成比较小的、模块化的行为。通过遵循 PDLC 的各个步骤设计程序，这种方式在本质上是模块化的，将会产生结构化的程序。PDLC 的每个阶段也都包含模块。例如，在 PDLC 的设计阶段，从顶层结构图到最低一级的指令，这种放大的模块化过程称为自上而下的设计。图 1.2 举例说明了层次图的概念，层次图也称为自上而下的图表或层次化输入处理输出(Hierarchical Input Process Output, HIPO)图。Java 支持模块化操作，把最初的程序规范集分解成比较小的、更好管理的组件，每个组件与原来的任务相比都更加容易解决。

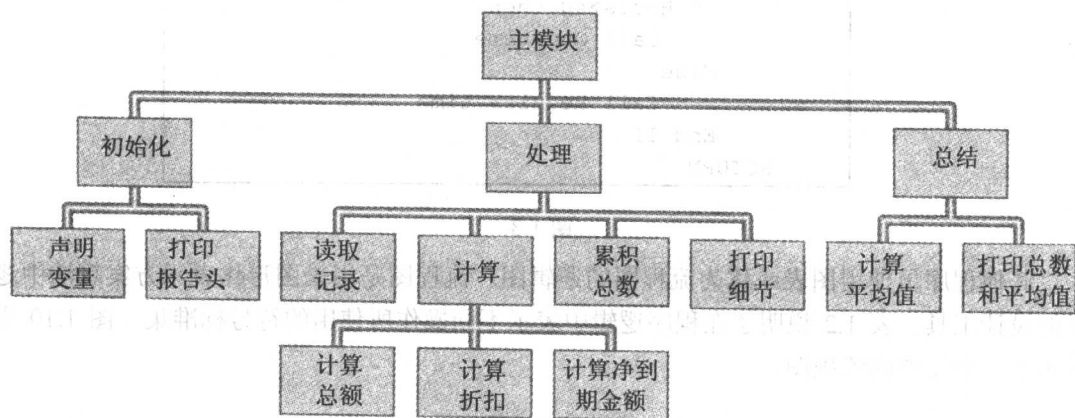


图 1.2

算法思维

计算机程序设计的目标是创建算法，即解决问题所需的明确、清晰的步骤规范。算法同样也需要准确和高效。准确指的是以有组织的方式使用逻辑构架和有效数据，以便可以正确执行各个步骤，而且程序可以对无效数据做出适当反应，例如，对于超出给定范围的数字、程序将给出告警信息，或者对与某些标准不匹配的数据进行反馈。高效指的是程序在非常短的保证有效性的时间内，或者在足够小的能够适应环境的空间内传递结果的能力。例如，如果程序查询某个产品的价格花费了很长时间，那么会失去顾客；或者如果计算机游戏占据非常大的计算机内存或附属存储器，那么它将失去市场占有率。计算机程序应该在需要进行修改和修订的某些事件中尽可能地简洁。

程序员使用许多工具进行算法上的考虑，并准确高效地设计他们的程序。把逻辑算法用图表示出来将会使程序编写得非常完美，而且也会使应用程序成功运行。有些程序员使用伪代码列出计算机应该执行的动作。伪代码(见图 1.3)使用关键字表达计算机动作，并使用缩排方式描述逻辑分组或结构。