

一群中国优秀程序员编程的亲身体会
一盏进入程序设计大门的指路灯
一套全新的多媒体教材
祝贺您的选择

洪恩教育
Human Education

编程之道

新编Java

程序设计入门

主编 赵 杰 杨 柳 高 巍

专家提示:

先学光盘后学教材
多媒体教学, 全程语音讲解, 案例演示

教材重点:

按初学者特点安排知识点的顺序
语言浅显, 实例穿插于讲解之中
通过章后练习, 提高实践能力



天津科学技术出版社

新编 Java 程序设计入门

赵杰 杨柳 高巍 主编



天津科学技术出版社

内 容 提 要

本教材是专门针对没有任何编程基础,但又想学习 Java 程序设计的人所编写的,从配置编程环境开始一步一步引领读者走入 Java 的世界。教材中没有用太多生涩的语言去讲述概念,而是尽量通过通俗易懂的语言、浅显的实例讲解,从实例出发引出相关的概念,使读者真正理解 Java 语言的实质,并建立面向对象的编程思想。最后通过图形编程的学习,既可以复习 Java 前面学习过的知识,又可以欣赏到自己编写的窗口应用程序。

本教材从开始到结束,力图通过一个完整的实例去阐述 Java 不同的应用技术。在结束时,再次完整地展现 Java 的魅力,使得读者在阅读本教材时有种浑然一体的感觉。

通俗、形象、生动的多媒体教学光盘更是为用户铺平了编程之道,可以辅助读者更快捷地掌握 Java 编程。该光盘具有以下特点:

- 全程语音讲解
- 真实操作演示
- 重要代码分析
- 详细示例剖析

 **提示** 强烈建议读者先学习教材中的多媒体光盘,通过光盘能在最短的时间内得到作者倾力传授的多年编程经验和技巧,可以有效地降低学习的难度,对读者来说是不可多得的快速入门的学习软件。教材中所涉及的全部例子源代码和相关软件的安装程序都可以在该光盘中找到。

图书在版编目(CIP)数据

新编 Java 程序设计入门 / 赵杰, 杨柳, 高巍主编. — 天津: 天津科学技术出版社, 2008.1
ISBN 978-7-5308-4451-9

I. 新… II. ①赵…②杨…③高… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 191218 号

责任编辑: 杨庆华 石崑

责任印制: 白彦生

天津科学技术出版社出版

出版人: 胡振泰

天津市西康路 35 号 邮编 300051

电话 (022) 23332393 (发行部) 23332392 (市场部) 27217980 (邮购部)

网址: www.tjkjcs.com.cn

新华书店经销

北京密云胶印厂印刷

开本 787 × 1092 1/16 印张 38 字数 790 000

2008 年 1 月第 1 版第 1 次印刷

定价: 59.00 元

为普及计算机技术做贡献

张孝文 书

原清华大学校长

前言

自 1995 年以来, Java 作为一种新型的面向对象的程序设计语言, 已闯入了 Internet 领域, 并迅速占据了显著地位。尤其是在网络编程、数据库操作等方面的出色表现, 使得 Java 在网络上的应用正呈现出高涨的强势, Java 美好的前景也正吸引着越来越多的人加入到 Java 编程的行列。

本教材从开始到结束, 力图通过一个完整的实例去阐述 Java 不同的应用技术。在结束时, 再次完整地展现 Java 的魅力, 使得读者在阅读本教材时有种浑然一体的感觉。

本教材分 20 章。第 1 章是对 Java 语言的整体描述, 初步建立面向对象的思想, 并针对初学者学习 Java, 提出了一些建议; 第 2 章主要是引导读者一步一步去搭建编程环境, 安装必要的软件, 配置环境变量, 使读者能自己动手配置适合于 Java 编程的环境; 第 3 章通过编写一个简单的示例程序, 测试编程环境, 并对 Java 源文件的命名与规范作了详细的讲解; 第 4 章开始讨论 Java 语言, 主要涉及数据类型、变量、常量等, 如果读者有一定 C 或 C++ 的编程基础, 这部分可能会相对简单一些, 因为 Java 语言的语法从本质上讲是和 C 相同的, 如果没有 C 的语言基础, 读者应当仔细阅读本章, 这是 Java 语言的基础; 第 5 章主要讲述了 Java 语言的循环与控制, 循环与控制是每一个程序所必须具有的, 是 Java 的核心内容之一, 作者对所举示例逐一进行了细致的分析; 第 6 章真正进入到面向对象的设计过程, 具体介绍了 Java 的类、包、方法等, 作者在实例分析过程中, 从生活中提取示例, 引导读者构建一个对象, 将面向对象程序设计的思想应用到实践中去; 第 7 章我们就会接触到进入 Java 语言的第一个对象——字符串, 从这个最基本的对象出发, 开始学习面向对象程序设计方法, 构建面向对象程序设计的思想; 第 8 章到第 11 章, 主要讲述了 Java 的核心技术, 包括抽象、接口、封装、继承等。通过全过程的示例使读者能够融会贯通地应用这些核心技术, 并且学会如何以面向对象的方式在实际的项目中应用这些技术; 第 12 章和第 13 章分别介绍了回收机制和异常处理; 第 14 章和第 15 章对数据库和数据结构基础作了详细的描述; 第 16 章到第 19 章讲的是封装的概念、图形编程和事件处理等内容; 第 20 章, 通过对一个实例的分析, 使读者不仅能运用所学知识, 而且能从 Java 面向对象编程的角度去分析并解决问题; 最后, 本教材通过一个附录, 简单探讨了 HTML 与 APPLET 的初步知识, 为下一步 JSP 与 Servlet 的学习, 设计与编写 Web 应用程序打下一个基础。

本书由赵杰、杨柳、高巍主编, 柴宝杰和邢军参加编写。赵杰拟定了编写内容和大纲, 执笔编写了第一章至第五章、第八章和第九章及附录, 并统阅了全书; 杨柳执笔编写了第十章至第十四章、第二十章; 高巍执笔编写了第十八章和第十九章; 柴宝杰执笔编写了第六章、七章; 邢军执笔编写了第十五章至第十七章。参加编写和审教等工作的还有王伟东、赵志芳和吴艳华。限于时间与水平, 书中难免有不妥处, 敬请批评指正。

编者

2007 年 12 月

光盘使用说明

一、光盘特点及内容

本光盘分为16部分,分别讲解了Java语言综述、Java编程环境、第一个程序分析、Java编程结构、流程控制、对象、类、包、数组、字符串、继承、多态、接口与内部类、抽象、异常与处理、JDBC 及其应用和封装等知识。

二、使用方法

为了获得最佳的学习效果,推荐在16位色800×600分辨率或以上显示模式下运行。将其中的一张教学光盘放入光驱后,它会自动播放。片头播放结束后,将会出现程序的主界面,如图1所示。如果光盘不能自动播放,则请双击光驱所对应的盘符来打开光盘内容,然后双击“Start.exe”(或“Start”)文件来播放光盘。

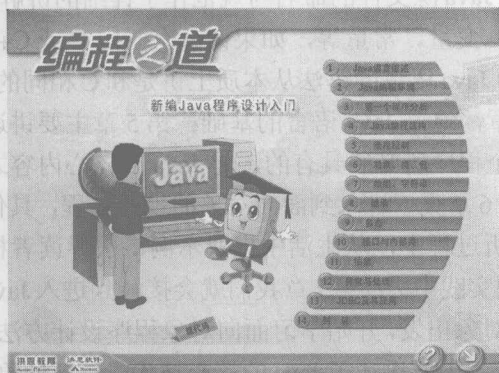


图1 程序主界面

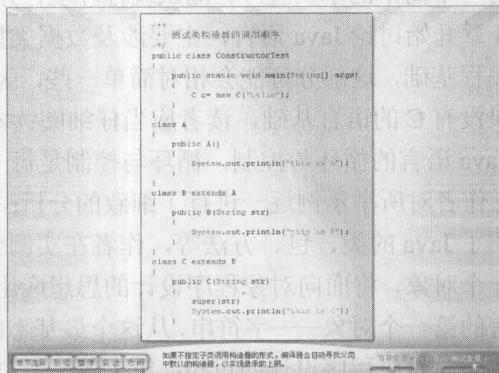


图2 学习界面

在程序主界面上,鼠标单击菜单条上的标题即可进入相应的学习内容(如图2所示),单击“帮助”按钮,可以查看使用说明,单击“退出”按钮,可以退出学习程序。

进入学习界面后,可以通过播放控制面板(图3)来控制学习状态,也可以通过键盘的快捷键(表1)来控制。需要返回到主界面时,按一下键盘上的Esc键即可。

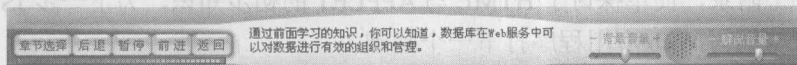


图3 播放控制面板

表1 键盘操作快捷键

作用范围	功能	按键
学习界面	后退	←(左方向键)
	暂停/播放	空格键
	前进	→(右方向键)
	返回	Esc
	背景音乐音量调节	“+”增大音量;“-”减小音量
	背景音乐开/关	M或m,按一下关,再按一下开
	背景音乐手动选择	按数字键1~5可以选择不同的背景音乐

在控制面板上单击“章节选择”按钮后,程序将弹出一个菜单,把鼠标移到该菜单上,然后单击就可以快速进入相应的学习内容;控制面板的“后退”、“暂停/播放”、“前进”可以灵活地控制学习进度,控制面板的中间部分显示对应的解说内容,而单击最后一个按钮(返回)时,将返回到正常的学习界面。

目 录

第1章 Java 语言综述	1
1.1 Java 是一门程序设计语言	1
1.2 面向对象初步	2
1.2.1 什么是对象	3
1.2.2 什么是对象的属性	4
1.3 Java 的特点	4
1.4 Java 与 C、C++ 的区别	6
1.5 Java 的不同版本	7
1.6 Java 的前景	8
1.7 学好 Java 的关键	8
第2章 Java 编程环境	11
2.1 J2SE 的下载与安装	11
2.1.1 J2SE 的下载	11
2.1.2 J2SE 的安装	14
2.2 环境变量配置与测试	15
2.2.1 设置 classpath	15
2.2.2 设置 path	17
2.2.3 环境变量的测试	18
2.3 API 文档的下载与安装	20
2.3.1 文档的下载	20
2.3.2 文档的安装	21
2.4 开发工具的安装	22
2.4.1 EditPlus 文本编辑器	22
2.4.2 UltraEdit 文本编辑器	22
第3章 第一个程序分析	23
3.1 第一个程序	23
3.1.1 编写你的第一个程序	23
3.1.2 编译与运行	24
3.1.3 第一个程序分析	24
3.1.4 注释	26
3.2 命名与保存	26
3.2.1 标识符与关键字	27
3.2.2 保存	27
3.3 Java 程序的开发流程	30

第4章	Java 编程基本结构	32
4.1	数制	32
4.1.1	基本概念	32
4.1.2	在 Java 中的表示形式	33
4.2	数据类型	33
4.2.1	整型	33
4.2.2	char 型	35
4.2.3	浮点型	36
4.2.4	布尔型	37
4.3	变量	38
4.3.1	变量的声明	38
4.3.2	变量的含义	39
4.3.3	变量的分类	39
4.3.4	变量的引用	43
4.4	基本类型的初始化	46
4.5	常量	49
4.6	Java 运算符	50
4.6.1	算术运算符	50
4.6.2	关系运算符	53
4.6.3	逻辑运算符	54
4.6.4	位运算符	58
4.6.5	移位运算符	60
4.6.6	赋值运算符	61
4.6.7	三元运算符	62
4.6.8	逗号运算符	63
4.6.9	字符串运算符	63
4.6.10	转型运算符	63
4.6.11	运算符的优先级	64
第5章	流程控制	66
5.1	编程风格	66
5.2	作用域	70
5.2.1	块作用域	70
5.2.2	对象的存活时间	74
5.3	条件语句	75
5.3.1	if (···)	76
5.3.2	if (···){···} else {···}	77
5.3.3	if ··· else if ···	79
5.4	循环语句	82
5.4.1	不确定循环	82
5.4.2	确定循环	86

5.5	中断与继续	90
5.5.1	中断	90
5.5.2	继续	91
5.5.3	标签语句	92
5.6	分支语句	94
5.7	返回语句	99
第 6 章	对象、包、类	101
6.1	对象	101
6.1.1	什么是对象	101
6.1.2	对象句柄	102
6.1.3	对象变量的初始化	102
6.2	类	102
6.2.1	类的含义	103
6.2.2	设计自己的类	103
6.2.3	分析你的类	106
6.3	使用已有的类	125
6.3.1	Math 类	125
6.3.2	Date 类	131
6.3.3	GregorianCalendar 类	135
6.3.4	对象重构	141
6.4	包	150
6.4.1	命名空间	151
6.4.2	包的概念及作用	151
6.4.3	包的实现	151
6.4.4	使用包	151
6.4.5	类路径与默认包	153
6.4.6	包作用域	154
6.5	注释及嵌入文档	154
6.5.1	如何添加注释	154
6.5.2	类注释	155
6.5.3	方法注释	156
6.5.4	字段注释	157
6.5.5	其他注释	157
6.6	类设计技巧	158
第 7 章	数组和字符串	160
7.1	数组	160
7.1.1	数组的声明	160
7.1.2	数组的初始化	161
7.1.3	匿名数组	164
7.1.4	数组的拷贝	167

7.1.5	命令行参数	170
7.1.6	数组排序	171
7.2	文本处理类	173
7.2.1	String 类	173
7.2.2	加号运算符	182
7.2.3	StringBuffer 类	183
7.3	格式化输入输出	185
7.3.1	读取输入	185
7.3.2	格式化输出	187
第 8 章	继承	192
8.1	继承的含义	192
8.2	继承的实现	192
8.3	构造器的调用	200
8.4	类之间的关系	202
8.4.1	依赖	203
8.4.2	聚合	203
8.4.3	继承	204
8.4.4	类记号与类图	204
8.5	继承层次图	204
8.6	关于设计好继承的几点建议	205
8.7	防止类的继承	205
第 9 章	多态	207
9.1	多态	207
9.1.1	多态的产生	207
9.1.2	多态的概念	208
9.1.3	多态的实质	208
9.1.4	绑定	211
9.2	超类	212
9.2.1	equals 方法	213
9.2.2	通用编程	213
9.3	数组列表	214
9.4	对象包装器	221
第 10 章	接口与内部类	224
10.1	接口	224
10.1.1	接口的产生	224
10.1.2	接口的概念	225
10.1.3	接口的声明	225
10.1.4	接口的实现	226
10.1.5	接口的多重实现	230

10.1.6	接口的属性	235
10.1.7	接口的继承	235
10.1.8	接口的意义	236
10.2	克隆	236
10.2.1	浅拷贝	237
10.2.2	深拷贝	244
10.3	内部类	247
10.3.1	使用内部类访问对象状态	247
10.3.2	局部内部类	257
10.3.3	静态内部类	259
10.3.4	匿名内部类	265
第 11 章	抽象	266
11.1	抽象行为	266
11.2	抽象类	267
11.3	抽象与接口的区别	271
第 12 章	内存管理与回收	273
12.1	不要自己清除对象	273
12.1.1	基本类型	273
12.1.2	对象类型	274
12.2	特殊情况要做的工作	274
第 13 章	异常与处理	276
13.1	处理错误	276
13.1.1	异常的分类	277
13.1.2	异常的声明	277
13.1.3	异常的抛出	278
13.2	异常的捕获	281
13.2.1	捕获多个异常	282
13.2.2	重新抛出异常	282
13.3	finally 子句	283
第 14 章	JDBC 及其应用	286
14.1	JDBC 综述	286
14.1.1	JDBC 驱动程序分类	286
14.1.2	JDBC 的目的	287
14.2	JDBC 基本编程概念	288
14.2.1	设定 ODBC 数据源	288
14.2.2	数据库 URL	291
14.2.3	建立连接	292
14.2.4	建立会话	294
14.2.5	操作数据库	295

14.2.6 高级 SQL 类型	307
14.3 事务处理	307
14.4 预查询	312
第 15 章 数据结构	314
15.1 数据结构的接口	314
15.2 Collection 与 Iterator 接口	316
15.2.1 Collection 接口	316
15.2.2 Iterator 接口	318
15.3 几个重要的类	319
15.3.1 链表	319
15.3.2 数组列表类	325
15.3.3 散列表	326
15.3.4 散列集	330
15.3.5 树集	331
15.3.6 映像	333
15.3.7 Vector 类	346
15.3.8 枚举	350
15.3.9 属性集	351
第 16 章 封装	354
16.1 数据封装的背景	354
16.2 数据封装在应用程序中的体现	355
16.2.1 封装类的设计	355
16.2.2 数据库查询类的设计	357
16.2.3 数据库连接类的设计	359
16.2.4 测试类的设计	361
16.2.5 复杂情况	364
第 17 章 图形编程初步	371
17.1 概述	371
17.2 创建一个空白框架	371
17.3 框架美化	373
17.3.1 设定标题	374
17.3.2 设定最小化图标	374
17.3.3 定位框架	375
17.3.4 框架的缩放	376
17.3.5 实例	376
17.4 内容显示	378
17.4.1 框架的结构	378
17.4.2 内容窗格	378
17.4.3 面板	379

17.4.4	实例	379
17.5	2D 图形	381
17.5.1	点	382
17.5.2	直线	382
17.5.3	矩形	383
17.5.4	椭圆	383
17.5.5	实例	384
17.6	色彩	387
17.6.1	面板配色	387
17.6.2	2D 图形	391
17.6.3	填充色	394
17.7	字体	399
第 18 章	事件模型与事件处理	402
18.1	事件处理基础	402
18.1.1	事件源与事件	405
18.1.2	事件监听器接口与事件监听器	405
18.1.3	注册事件监听器	407
18.1.4	添加组件	407
18.1.5	选择事件监听器	408
18.1.6	匿名内部类	411
18.2	适配器类	417
18.3	Java 事件模型	418
18.3.1	事件层次图	419
18.3.2	语义事件与低级事件	420
18.3.3	事件模型小结	421
18.4	实例	422
第 19 章	SWING 组件与布局管理器	439
19.1	MVC 设计模式基础	439
19.2	简单按钮	440
19.2.1	动作	440
19.2.2	生成按钮	442
19.2.3	按钮模型	443
19.2.4	添加按钮	444
19.2.5	按钮的美化	447
19.3	布局管理器初步	452
19.3.1	流布局管理器	452
19.3.2	边界布局管理器	456
19.4	其他的按钮	461
19.4.1	单选按钮	461
19.4.2	复选按钮	467

19.4.3 单击按钮	471
19.5 文本输入组件	474
19.5.1 文本域	474
19.5.2 密码域	480
19.5.3 文本区	481
19.6 标签组件	484
19.7 选择组件	485
19.7.1 组合列表框	485
19.7.2 滑块	488
19.8 菜单组件	495
19.8.1 创建菜单栏	495
19.8.2 菜单	496
19.8.3 菜单项	497
19.8.4 菜单项的美化	499
19.8.5 复选框及单选按钮菜单项	500
19.8.6 弹出菜单	503
19.8.7 快捷键	506
19.8.8 菜单项的启用与禁用	508
19.8.9 对话框	511
19.9 复杂布局管理器	517
19.9.1 网格布局	517
19.9.2 箱式布局	521
19.9.3 网格组布局	526
第 20 章 综合实例与分析	533
20.1 需求任务	533
20.2 分析实现	534
20.2.1 数据库的实现	534
20.2.2 基础类的实现	540
20.2.3 界面类的实现	546
附录 HTML 与 APPLET 初步	588
1. APPLET 与 HTML 基础	588
2. 一个简单的 applet	588
3. applet 与 AWT 组件	590
4. applet 与 Swing 组件	591
5. 浏览器打开 applet	592
6. applet 的生命周期	592

第1章 Java语言综述

本章要点：

- 面向对象设计初步
- Java语言的特点
- 学好Java的关键

这一章，我们将通过Java的发展，分析它的特点，初步建立面向对象的概念，并对面向对象的核心思想有个比较清晰的认识。通过本章的学习，读者可以对Java本身有一个比较全面的认识，并了解学习好Java应该注意的事项。

就现阶段而言，弄清本章中介绍的所有细节并不重要，读者会在本书稍后的章节中深入地学习这些内容。如果想尽快地进入Java的学习过程中，本章完全可以略过，等对Java语言本身有了一定程度的掌握，再回来浏览一下，也许能有更深的理解，更好地品尝这杯美味的咖啡。

 **提示** Java的由来印度尼西亚有一个重要的盛产咖啡的岛屿Java（中文译名为爪哇）。这门语言的开发者就为它起名为Java，其寓意是为世人端上一杯热气腾腾的咖啡。所以在Sun公司有关Java的产品上都会有一杯冒着热气咖啡的卡通图标。

1.1 Java是一门程序设计语言

Java是1995年6月由Sun公司发布的。它被美国著名杂志《PC Magazine》评为1995年十大优秀科技产品。之所以称Java为革命性编程语言，是因为传统编程语言的实现与具体的环境有关，而用Java编写的程序却不必考虑这些问题，它在所有平台上的字节码是兼容的，只要提供Java解释器，Java编写的程序就能在计算机上运行，以实现Java编写者“一次编写，随处可用”的口号。

 **提示** 提到字节码，也许大家比较陌生，下面对字节码进行简单的解释。我们用Java编写的程序称为源码（source code），又称为源文件，源码是不可以被机器执行的。要想使程序得以运行，必须利用编译器（不同的语言有不同的编译器）对源码进行编译，编译器将源码编译成计算机能读懂的语言。对于C来讲，它是将源码编译为二进制码（exe文件，也被称为机器码），这样计算机就可以读懂，它就可以按照人类的意愿去实现一定的功能。对于Java来讲，它并不是被编译为二进制码，而是将源码编译为一种独立于机器平台的中间代码，这种中间代码就称为字节码。字节码可以被Java解释器所执行，由解释器将字节码再翻译成二进制码，使得程序得以运行。

Java 是一种新型的面向对象的程序设计语言，可以说它是顺应网络的迅猛发展而出现的。它是第一个能编写可嵌入 Web 网页中的所谓小应用程序（applet）的程序设计语言。

作为计算机语言，Java 曾有过火的宣传。但从现在看来，Java 真正的用武之地应当是嵌入式开发，它也正在逐渐变成开发手持设备、互联网信息网站、车载计算机等应用的标准。Java 自从正式发布至今已经变得更加稳定、强健，现在 Java 的类库仍然在不断的壮大、扩展中，相信它的发展与壮大，必将使得 Java 在网络世界的应用越来越广泛。

1.2 面向对象初步

Java 是一种面向对象的编程语言，要想真正掌握 Java，首先必须明确的就是面向对象的含意，面向对象的核心思想。最近几年，面向对象编程在软件开发领域掀起了一阵狂热的风潮，得到了迅猛发展，受到越来越多的关注，也有越来越多的人加入到 Java 的开发行列，那么究竟是什么原因使如此多的人热衷于面向对象编程呢？

面向对象编程（OOP）具有多方面的吸引力，对于生产管理人员来说，它实现了一次性投入多次使用，使开发成本更加低廉。对于设计分析人员来说，利用 UML 建模更加直观、方便，完成的程序更加易于维护。对于程序员来讲，更易于领会设计人员的意图，使开发过程不再变得枯燥无味。



提示 什么是 UML？

统一建模语言（Unified Modeling Language），是在 1997 年 11 月由国际对象管理组织批准认定的面向对象建模语言，它的伟大之处就是统一了软件的分析、设计、编写的规范。就像一张施工设计图，只要是按照统一模式设计的图纸，每一个人都可以看懂。

任何事情都是双方面的，在利用面向对象的好处时，我们必须为掌握它而付出努力，也就是思考对象的过程，需要采用抽象思维，而不是程序化的思维方式，这种思想的形成是需要长时间努力和实践的。所以说面向对象不仅仅是一种编程语言的实现，更重要的是抽象思想的形成。这种思想就是把一切事物都抽象化为对象，并给对象赋予一定的特征，简单地说面向对象就是抽象与具体的过程，抽象的过程是得到对象，具体的过程是对象的实例化。也许有人会问那以前我们软件开发的方式是怎么样的呢？

这个问题可以这样回答：在 Java 诞生以前开发软件的方式基本上都是面向过程的。接下来也许大家更想知道面向过程与面向对象到底有什么样的区别呢？我们就用一个比较形象的比喻来简单说明一下：如果我们把程序实现的过程形象地比喻为盖房子，那么面向过程就是将水、土等原材料按照房子的建筑顺序一直盖上去，一旦房子过时需要重新盖的话，所有的原材料都将没有用处了。而面向对象则是在动手盖房子之前，先烧砖，将原材料烧制成一块块用于不同目的的砖（这些砖并不像我们日常生活中所见的砖都是一样的形状，我所强调的是用于不同目的的砖，你可以理解为不同的形状。比如说我们有的地方需要圆形的砖，有的地方需要方形的砖，那么我们就制成圆形或方形的砖），再根据房子的需要将这些砖按照一定的顺序组织起来，很明显，一旦房子过时了，你的砖还可以搬到别

处用（如果用专业一点的话说就是代码的复用性）。这就是面向对象的精髓。当然，砖烧制的好坏，更确切地说应该是形状烧制的好坏，这才是面向对象编程最具挑战性的一面，也就是说能烧制出可以不断重复使用的砖，对于程序开发人员来说，正是它的魅力所在。

由于砖的制成，盖新房子时，可以利用部分旧砖，而不需要重新去烧制同样的砖。这对于房子的主人来说，在一定程度上降低了成本，对于施工者来说，使盖房子这种工作也变得轻松了。

我们可以再说一个更加形象的例子，积木相信大家都见过，也都玩过，利用一块块不同形状の木块，你可以搭建不同形状的房子，这些不同形状の木块就是你所要烧制的砖。当然积木是有人已经给做好了的不同形状の木块，现在编程中需要的是你自己去做这些不同形状の木块。这是不是很有挑战性呀？

当然，通过这几段文字，不可能完全涵盖面向对象的思想。这里只是想让大家对面向对象有一个初步的认识，并在以后的学习过程中尝试着用面向对象的思维方式去思考问题、分析问题。在以后的章节中，笔者会将面向对象的思想贯穿其中，使大家在学习的过程中，不仅仅学习语言，更重要的是面向对象思维方式的逐渐形成。

1.2.1 什么是对象

也许大家一看到对象，就觉得担心，其实没有必要，Java的对象是一个相当简单的东西，你可以把对象理解为一个有形的东西，就像物品一样实实在在存在的東西。

任何事物都可以看成是一个对象，对象随处可见。正如前面所提到的，你可以暂时将对象理解为现实生活中存在的实实在在的物品，像灯、桌子等。也可以理解为抽象中的每一件事情、逻辑等。那么这每一个物品、每一件独立的事物你都可以作为一个对象；也可以把一类东西作为一个对象，正如我们前面所讲述的砖和积木，你可以把不同形状的砖或积木作为一个对象，也可以把两个或多个相同形状的砖或积木看做一个对象，也可以把所有的砖或积木作为一个对象。

如图1-1所示，有三顶不同颜色、不同形状、不同材质的帽子，每一项帽子可以作为一个对象（因为他们确实有不同之处），而所有的帽子也可以作为一个对象（因为他们确实有相同之处）。每一个对象都可以对应于Java中的一个类（class）。现在大家先不要急于去理解什么是类，在这里你可以只把它作为一个概念，在以后的学习中我们会慢慢深入地学习、理解它。

通过这幅图我们可以看出，三顶不同颜色、形状、材质的帽子可以看成是一个统一的对象，所以我们可以说不同的对象可以构建新的对象，结合我们上面的讲述，可以得出下面的结论。

1) 所有的事物都可以是对象。