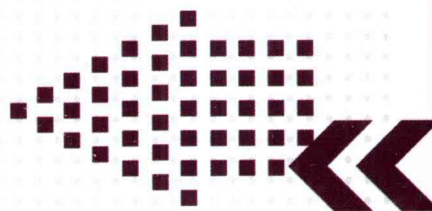




普通高等教育“十二五”计算机类规划教材

Java程序设计

◎ 陈 锐 主编



免费电子课件



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”计算机类规划教材

Java 程序设计

主 编 陈 锐

副主编 陈亚民 肖 敏

参 编 张恒喜 王 悦 李 红

刘 河 李学国 刘益洪



机械工业出版社

Java 语言是目前最为流行的程序设计语言之一,也是计算机专业的一门必修课。本书是作者多年来从事教学和研究的心得之作,内容包括 Java 语言概述、Java 开发环境、基本数据类型、流程控制、对象、类、包、继承和多态、接口和抽象、异常与处理、Java Applet 编程、数组和字符串、数据库编程、图形用户界面设计、多线程机制、网络编程。本书内容全面,结构清晰,语言通俗流畅,重点难点突出,例题丰富,所有程序都能够直接运行。

本书可作为大中专院校计算机及相关专业的 Java 程序设计课程教材,也可作为计算机软件开发、等级考试和软件资格(水平)考试人员的参考书。

本书配有免费电子课件,欢迎选用本书作教材的老师登录 www.cmpedu.com 注册下载或发邮件到 xufan666@163.com 索取。

图书在版编目(CIP)数据

Java 程序设计/陈锐主编. —北京:机械工业出版社,2011.8

普通高等教育“十二五”计算机类规划教材

ISBN 978-7-111-34595-4

I. ①J… II. ①陈… III. ①Java 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 117473 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:徐 凡 责任编辑:徐 凡

版式设计:霍永明 责任校对:赵 蕊

封面设计:张 静 责任印制:杨 曦

保定市中国画美凯印刷有限公司印刷

2011 年 8 月第 1 版第 1 次印刷

184mm×260mm·21 印张·516 千字

标准书号:ISBN 978-7-111-34595-4

定价:39.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

前 言

在多年来的教学与研究过程中，作者读过不少 Java 教材，有些教材写得较好，但有些教材内容不是很全面，语言表述方面也偏于晦涩，看书时间长了会比较辛苦。基于此，作者使用了活泼的语言编著了本书，把 Java 语言的一些心得分享给大家。

Java 语言诞生于 20 世纪 90 年代，由于 Java 语言的跨平台、面向对象、安全性和健壮性等特点而受到 IT 界的欢迎。现在，Java 语言已经在各个领域获得广泛的应用，Java 语言已经成为广大程序员的理想开发工具，Java 语言越来越显示出它强大的生命力。

目前，大多数高等院校的计算机专业不仅开设了 Java 语言课程，而且非计算机专业也开设了 Java 语言课程。全国计算机等级考试、软件资格（水平）考试都将 Java 语言列入考试范围。在数据结构与算法课程中，也将 Java 语言作为描述语言。

本书作者多年来一直从事 Java 程序设计的教学与研究工作，并从事数据结构与算法、自然语言理解的研究，具有较为丰富的科研实践经验与程序开发能力，对程序设计有深刻的研究，书中很多地方都是作者教学经验的总结与积累，抽取出需要掌握的重点与难点，方便学生掌握与学习。

本书特点

1. 内容全面，讲解详细

本书从 Java 语言的发展历史与特点、开发环境的安装与使用开始介绍，然后介绍了数据类型、流程控制、继承与多态等基本内容，最后还介绍了数据库编程与网络编程。本书内容全面，覆盖了 Java 语言的几乎全部基础知识，对于每个知识点，都使用具体实例进行讲解，以便学生理解。

2. 结构清晰，内容合理

本书按照章、节和小节划分知识点，将每个知识点细化，结构框架清晰，内容讲解合理，易于理解与学习。在知识点的讲解过程中，循序渐进，由浅入深，最后通过例子强化知识点。这样的讲解方式使学生更容易理解和消化。

3. 语言通俗，叙述简单

本书从头到尾全部采用通俗流畅的语言讲解每个知识点，避免使用晦涩难懂的语句。这样的表述方式使学生和读者轻松学会 Java 语言。

4. 例子典型，深入剖析

在讲解每一个 Java 语言知识点时，结合具体例子进行剖析。所选取的例子都是最为常见且涵盖知识点丰富的典型程序。

本书内容

第 1 章介绍 Java 语言的发展历史与特点、面向对象概念，并介绍 Java 语言的前景。

第 2 章介绍 J2SE 的安装与配置，并分析第一个 Java 语言程序。

第3章主要介绍基本数据类型（整型、浮点型、字符型、布尔型）、变量与常量、Java运算符（算术运算符、关系运算符、逻辑运算符、位运算符、移位运算符、赋值运算符等）。

第4章主要介绍条件语句和循环语句、break语句和continue语句。

第5章主要介绍面向对象程序设计基础。首先介绍对象和类的概念，然后介绍包的概念。

第6章主要介绍继承和多态。首先介绍继承的概念，然后介绍类与类之间的关系，最后介绍多态。

第7章主要介绍接口与抽象。首先介绍接口的产生、接口的声明与实现、接口的继承，然后介绍浅拷贝和深拷贝，最后介绍内部类与抽象。

第8章主要介绍异常与处理。

第9章主要介绍Java Applet小程序的开发过程。

第10章主要介绍Java的数组与字符串。

第11章主要介绍数据库编程。首先介绍JDBC相关技术，然后介绍如何查询、建表、创建数据库。

第12章主要介绍图形用户界面设计。首先介绍AWT基本组件，然后介绍布局管理器，接着介绍AWT事件处理模型，最后介绍Swing。

第13章主要介绍Java的多线程机制。首先介绍Java的线程概念、多线程的实现方法，最后介绍线程的控制。

第14章主要介绍网络编程。首先介绍网络编程的概念，然后介绍基于URL的网络编程和基于套接字的网络编程，最后利用数据报发送与接收数据。

本书由陈锐（国家高级程序员）任主编，武汉理工大学陈亚民、肖敏任副主编，中国人民解放军徐州空军学院张恒喜、西安文理学院王悦、华中师范大学汉口分校刘河、北京信息职业学院李红、重庆科创职业学院李学国、娄底职业技术学院刘益洪参编。全书由陈锐负责统稿。

在本书的出版过程中，中原工学院夏敏捷、渤海船舶职业学院邢容、聊城职业技术学院冯志祥等老师给本书提了许多宝贵意见，在此表示衷心的感谢。

本书配有免费电子课件，欢迎选用本书作为教材的老师登录 www.cmpedu.com 注册下载或发邮件到 xufan666@163.com 或 nwuchenrui@126.com 索取。

作者

目 录

前言

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 对象的概念	3
1.2.2 对象的属性	4
1.3 Java 的特点	4
1.4 Java 的不同版本	5
1.5 Java 的前景	6
1.6 学好 Java 的关键	6
1.7 小结	7
1.8 练习	7
第 2 章 Java 编程环境	8
2.1 J2SE 的安装与配置	8
2.2 环境变量的配置与测试	9
2.2.1 设置 classpath	9
2.2.2 设置 path	12
2.2.3 环境变量的测试	12
2.3 第一个 Java 程序	14
2.3.1 编写第一个 Java 程序	14
2.3.2 编译与运行	15
2.3.3 第一个程序分析	15
2.3.4 注释	17
2.4 命名与保存	18
2.4.1 标识符与关键字	18
2.4.2 保存	18
2.5 Java 程序的开发流程	21
2.6 小结	22
2.7 练习	22
第 3 章 数据类型	23
3.1 数据类型	23
3.1.1 整型	23
3.1.2 char 型	25
3.1.3 浮点型	26

3.1.4 布尔型	27
3.2 变量	28
3.2.1 变量的声明	28
3.2.2 变量的含义	29
3.2.3 变量的分类	29
3.2.4 变量的引用	32
3.3 基本类型的初始化	35
3.4 常量	38
3.5 Java 运算符	39
3.5.1 算术运算符	40
3.5.2 关系运算符	43
3.5.3 逻辑运算符	44
3.5.4 位运算符	47
3.5.5 移位运算符	50
3.5.6 赋值运算符	51
3.5.7 三元运算符	52
3.5.8 逗号运算符	52
3.5.9 字符串运算符	53
3.5.10 强制类型转换	53
3.5.11 运算符的优先级	53
3.6 小结	54
3.7 练习	54

第 4 章 流程控制	55
4.1 编程风格	55
4.2 作用域	58
4.2.1 块作用域	59
4.2.2 对象的存活时间	62
4.3 条件语句	63
4.3.1 if (···)	64
4.3.2 if (···) {···} else {···}	65
4.3.3 if ··· else if ···	66
4.4 循环语句	69
4.4.1 不确定循环	69
4.4.2 确定循环	73
4.5 中断与继续	77
4.5.1 中断	77

4.5.2 继续	78	6.4.2 聚合	154
4.5.3 标签语句	79	6.4.3 继承	154
4.6 分支语句	81	6.5 继承层次图	155
4.7 返回语句	85	6.6 关于设计好继承的几点建议	155
4.8 小结	87	6.7 防止类的继承	155
4.9 练习	87	6.8 多态	156
第5章 面向对象编程基础	88	6.8.1 多态的产生	156
5.1 对象	88	6.8.2 多态的概念	157
5.1.1 对象的特征	88	6.8.3 多态的实质	157
5.1.2 对象实例名	88	6.8.4 绑定	159
5.1.3 对象变量的初始化	89	6.9 超类	161
5.2 类	89	6.9.1 equals 方法	162
5.2.1 类的含义	89	6.9.2 通用编程	162
5.2.2 设计自己的类	90	6.10 数组列表	163
5.2.3 分析你的类	93	6.11 对象包装器	169
5.3 使用已有的类	111	6.12 小结	171
5.3.1 Math 类	111	6.13 练习	171
5.3.2 Date 类	116	第7章 接口和抽象	172
5.3.3 GregorianCalendar 类	119	7.1 接口	172
5.3.4 对象重构	125	7.1.1 接口的产生	172
5.4 包	134	7.1.2 接口的概念	172
5.4.1 命名空间	134	7.1.3 接口的声明	172
5.4.2 包的概念及作用	134	7.1.4 接口的实现	173
5.4.3 包的实现	135	7.1.5 接口的多重实现	177
5.4.4 使用包	135	7.1.6 接口的属性	182
5.4.5 类路径与默认包	137	7.1.7 接口的继承	183
5.4.6 包作用域	137	7.1.8 接口的意义	183
5.5 注释及嵌入文档	137	7.2 克隆	184
5.5.1 如何添加注释	138	7.2.1 浅拷贝	184
5.5.2 类注释	138	7.2.2 深拷贝	191
5.5.3 方法注释	139	7.3 内部类	194
5.5.4 字段注释	140	7.4 抽象行为	197
5.5.5 其他注释	140	7.5 抽象类	198
5.6 小结	141	7.6 抽象与接口的区别	203
5.7 练习	142	7.7 小结	203
第6章 继承和多态	143	7.8 练习	204
6.1 继承的含义	143	第8章 异常与处理	205
6.2 继承的实现	143	8.1 处理错误	205
6.3 构造器的调用	151	8.1.1 异常的分类	205
6.4 类之间的关系	153	8.1.2 异常的声明	206
6.4.1 依赖	154	8.1.3 异常的抛出	207

8.2 异常的捕获	209	第 12 章 图形用户界面设计	266
8.2.1 捕获多个异常	210	12.1 AWT 工具集简介	266
8.2.2 重新抛出异常	211	12.2 AWT 基本组件	268
8.3 finally 子句	212	12.2.1 标识	268
8.4 小结	214	12.2.2 按钮	268
8.5 练习	214	12.2.3 画布	268
第 9 章 Java Applet 编程	215	12.2.4 复选框	270
9.1 编写 Applet 小程序	215	12.2.5 单选按钮	270
9.2 Applet 的执行过程	216	12.2.6 菜单	271
9.3 小结	218	12.2.7 菜单项	271
9.4 练习	219	12.2.8 列表框	272
第 10 章 数组和字符串	220	12.2.9 单行文本输入框	272
10.1 数组	220	12.2.10 文本输入区	273
10.1.1 数组的声明	220	12.2.11 滚动条	273
10.1.2 数组的初始化	220	12.2.12 文件对话框	273
10.1.3 匿名数组	223	12.2.13 一个 AWT 的综合例子	274
10.1.4 数组的拷贝	224	12.3 AWT 容器	275
10.1.5 命令行参数	228	12.3.1 Frame	275
10.1.6 数组排序	229	12.3.2 Panel	276
10.2 文本处理类	230	12.4 布局管理器	277
10.2.1 String 类	230	12.4.1 FlowLayout	277
10.2.2 StringBuffer 类	238	12.4.2 BorderLayout	278
10.3 使用 StringTokenizer 类分析字符串	240	12.4.3 GridLayout	280
10.4 Character 类	241	12.4.4 CardLayout	280
10.5 字符串与字符、字节数组	242	12.4.5 GridBagLayout	281
10.5.1 字符串与字符数组	242	12.4.6 容器的嵌套	283
10.5.2 字符串与字节数组	244	12.5 AWT 事件处理模型	284
10.6 小结	244	12.5.1 事件类	286
10.7 练习	245	12.5.2 事件监听器	287
第 11 章 数据库连接及其应用	246	12.5.3 AWT 事件及其相应 的监听器接口	288
11.1 JDBC 基本编程概念	246	12.5.4 事件适配器	292
11.1.1 设定 ODBC 数据源	246	12.6 Swing 简介	295
11.1.2 数据库 URL	250	12.6.1 Swing 与 AWT	295
11.1.3 建立连接	251	12.6.2 Swing 程序结构简介	295
11.1.4 建立会话	252	12.7 小结	297
11.1.5 操作数据库	252	12.8 练习	297
11.2 预查询	264	第 13 章 Java 多线程机制	299
11.3 小结	265	13.1 Java 中的线程	299
11.4 练习	265	13.1.1 进程与线程	299
		13.1.2 线程的状态	299
		13.1.3 多线程的实现方法	300



13.2 通过继承 Thread 类实现多线程	301	14.2 基于 URL 的 Java 网络编程	312
13.3 通过 Runnable 接口实现多线程	302	14.2.1 URL 类	312
13.4 线程的调度	303	14.2.2 URLConnetction 类	314
13.5 线程的同步	305	14.3 基于套接字的 Java 网络编程	315
13.5.1 基本的线程控制	305	14.3.1 Socket 通信简介	315
13.5.2 多线程的同步实现	306	14.3.2 创建 Socket 和 ServerSocket	315
13.6 小结	309	14.3.3 Socket 简单应用	317
13.7 练习	309	14.4 数据报	319
第 14 章 Java 网络编程	310	14.5 小结	324
14.1 网络编程的基本概念	310	14.6 练习	324
14.1.1 网络基础知识	310	参考文献	325
14.1.2 网络基本概念	311		

第 1 章 Java 语言综述

本章先介绍 Java 语言的发展，分析它的特点，初步建立面向对象的概念，使读者对面向对象的核心思想有个比较清晰的认识。通过学习本章，读者可以对 Java 语言本身有一个比较全面的认识，并了解学习好 Java 语言应该注意的问题。

就现阶段而言，完全掌握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 语言的学习，可以略过本章，等对 Java 语言本身有了一定程度的掌握后，再回来阅读本章，也许能有更加深刻的理解。

提示

Java 的由来

印度尼西亚有一个重要的盛产咖啡的岛屿，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。因此在 Sun 公司的有关 Java 的产品上都会有一杯冒着热气的咖啡的卡通图标。

1.1 Java 是一门程序设计语言

Java 语言是 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 为革命性的编程语言，是因为传统编程语言的实现与具体的操作环境有关，而用 Java 语言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，用 Java 语言编写的程序就能在该计算机上运行，实现了 Java 语言程序编写者“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。

用 Java 语言编写的程序称为源码 (Source Code)，又称为源文件，源码是不能被机器执行的。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 语言来讲，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 语言来讲，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。



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、互联网信息网站以及车载计算机等应用的标准。Java 语言自从正式发布至今已经变得更加稳定、强健，现在 Java 语言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 语言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 编程，首先必须明确的就是什么是面向对象以及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；对于程序员来说，更容易理解并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, UML），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、设计和编写的规范。UML 就像一张施工设计图，只要是按照统一模式设计的图样，每一个人都可以看懂。

任何事情都具有两面性，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，这种思想的形成需要长时间的努力和实践。因此说，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 语言诞生之前，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。

可以用一个比较形象的比喻来说明。例如，对于五子棋游戏，面向过程的设计思路就是按照具体的执行步骤分析问题：

- 1) 开始游戏。
- 2) 黑子先走。
- 3) 绘制画面。
- 4) 判断输赢。
- 5) 轮到白子。
- 6) 绘制画面。

- 7) 判断输赢。
- 8) 返回步骤 2。
- 9) 输出最后对局结果。

把上面每个步骤分别用函数来实现，问题就解决了。

而面向对象的设计则是从功能角度来解决问题。整个五子棋游戏可以分为 3 个子系统：一是黑白双方，这两方的行为是一模一样的；二是棋盘系统，负责绘制画面；三是规则系统，负责判定诸如犯规、输赢等。玩家对象负责接受用户输入，并告知棋盘对象的棋子布局变化，棋盘对象接收到了棋子的变化就要负责在屏幕上显示这种变化，同时利用规则系统来对棋局进行判定。

同样是绘制棋局，这种行为在面向过程的设计中分散在许多步骤中，很可能出现不同的绘制版本，因为设计人员通常会根据实际情况进行各种各样的简化。而面向对象的设计中，绘图只可能在棋盘对象中出现，从而保证了绘图的统一。

面向对象设计具有可维护性。比如，要加入悔棋的功能，如果要改动的是面向过程的设计，那么从输入、判断到显示这一连串的步骤都要改动，甚至步骤之间的循序都要进行大规模调整。如果是面向对象的设计，只用改动棋盘对象就行了，棋盘系统保存了黑白双方的棋谱，简单回溯就可以了，而其他功能则不用修改，改动只是局部的。

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其有一个初步的认识，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，我们会将面向对象的思想贯穿其中，使大家在学习的过程中，不仅掌握 Java 语言并使编程能力得到提高，更重要的是逐渐形成面向对象的思维方式。

1.2.1 对象的概念

任何事物都可以被看成是一个对象，对象随处可见。我们可以把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。每个物品、每件独立的事物都可以作为一个对象；还可以把一类东西作为一个对象，如砖和积木，把不同形状的砖或积木作为一个对象；还能把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 3 顶不同颜色、形状和材质的帽子，每一顶帽子可以作为一个对象（因为它们确实有不同之处），而所有的帽子也可以作为一个对象（因为它们确实有相同之处）。

通过这幅图我们可以看出，3 顶不同颜色、形状、材质的帽子可以被看成一个统一的对象，那就是帽子。可以说，不同的对象可以构建成新的对象。结合我们上面的描述，可以得出下面的结论：

1) 所有的事物都可以是对象。

2) 对象可以是一个容器，有自己的存储空间，可以容纳不同的对象。如图 1-1 所示，对象帽子可以容纳 3 顶或多顶不同形状、颜色和材质的帽子。也就是说，我们可以根据已经存在的对象构建新的对象。

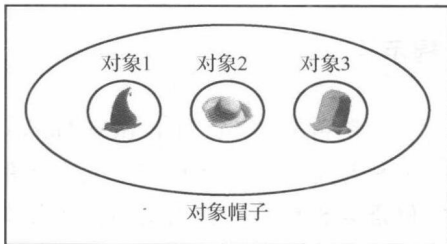


图 1-1 对象



3) 对象是程序构建的基本元素, 程序是一大堆对象的组合。通过各个对象间的消息传递, 每个对象都知道自己要做些什么。

4) 同一类的所有对象都能接收相同的信息。根据图 1-1 所示, 帽子都有材质这个属性, 可以向对象 1 传递材质的请求, 也可以向对象 2 及对象 3 传递材质的请求。

1.2.2 对象的属性

对象的属性就是与对象密切相关的特征。还以图 1-1 为例, 对象 1 是一顶帽子, 它的颜色是红、蓝相间的, 它的材质是布的。对象 2 的材质却是草的, 两个对象的颜色、形状也不同。所以我们可以说, 颜色、形状和材质就是帽子的特征。每一种固定的特征属于一个固定的对象, 正如图 1-1 中, 材质为草的帽子只属于对象 2, 而颜色为绿色的帽子只属于对象 3。

如果我们把 3 顶帽子看成一个对象, 那么对象 1、对象 2、对象 3 就是对象帽子的一个实例, 每一个实例保持着属于本身所特有的特征。

1.3 Java 的特点

Java 语言自产生以来, Sun 公司致力于使 Java 成为一个简单 (Simple)、面向对象的 (Object Oriented)、分布式的 (Distributed)、解释的 (Interpreted)、健壮的 (Robust)、安全的 (Secure)、结构中立的 (Architecture Neutral)、可移植的 (Portable)、高效能的 (High Performance)、多线程的 (Multithreaded)、动态的 (Dynamic) 程序语言 (摘译自 The Java Language: A White Paper, 1995)。正是由于它的这些特点的存在, 使得 Java 语言成为目前使用最为广泛的网络编程语言。

另外, Hank Shiffman (《Making Sense of Java》一书的作者) 也针对部分人对 Java 语言的错误看法及观念提出了他的说明, 特在此摘译如下。

(1) Java 是编写 Web 页面的一种语言, 就如同 HTML 和 VRML 一样

事实上, Java 并不是 HTML 一类的描述语言 (Description Language), 而是一种编程语言 (Programming Language)。描述语言是标明内容和位置, 而编程语言则是描述一种产生结果的过程。

提示

1. 什么是 HTML

所谓 HTML (Hypertext Markup Language) 是目前 WWW 上应用的文本标记语言。又称为超文本标记语言。它主要用于创建静态页面的。由于网络发展的需要已经与其他语言相结合, 以满足动态页面的需要, 如 JSP, ASP 等。

2. 什么是 VRML

这是类似于 HTML 语言的模型语言, 它的原意是虚拟现实模型语言。也就是在现实的世界中构建一个环境, 模拟出一个近乎完美的虚拟世界。有关这方面的知识读者可以参阅相关的资料。

(2) Java 语言容易学习和使用, 不像 C、C++ 和其他程序语言

Java 是一种编程语言。Java 语言容易学吗? Java 或许是比 C 或 C++ 容易学, 但仍是一种编程语言, 要想真正掌握它, 也必须付出相当的努力。

(3) Java 代码是可移植的, 但 C 及 C++ 不是

Java 源代码 (Source Code) 的可移植性比 C 语言好一些, 差别在于 Java 的字节码。Java 源码编译后, 就能在所有的机器上执行, 条件是机器上要有 Java 解释器。

(4) Java 语言能被拓展, 从而能在机器上执行任何事情

理论上说, Java Applet (Java 小应用程序) 能做任何事情, 如模拟 3D VRML 模型、播放音频、视频等。但事实上, 一个小应用程序 (Applet) 仅能在那一个页面上被执行, 而没有办法在除此以外的其他页面执行。同时, Java 的拓展也受限于 Java 类库的功能。

(5) Java 适合于建立大型的应用程序

如果说 Java 适合做大型程序, 则 Java 就不适合应用于 Web 浏览器了。第一个商业用的 Java Applets 并不是全部用 Java 开发的, Java 只是用来处理界面部分。所以说 Java 对于开发大型的应用程序有一定的难度。

(6) Java 是解释执行的, Basic 也是解释执行的, 因此 Java = Basic

虽然 Java 的确是使用解释器, 但事实上, Java 则与 C 或 C++ 等完全编译语言较为相近, 只不过编译所产生的文件不一致而已, 这个区别在上面已经有所叙述。

1.4 Java 的不同版本

每一个接触到 Java 的人, 刚开始时也许会为 Java 不同的版本所迷惑, 搞不清楚它们是在做什么的, 也不清楚自己应该选用哪个版本。下面就 Java 目前不同的版本做一个简单介绍。

(1) Java Developers Kits (JDK)

JDK 是 Java 开发工具箱, 是 Sun 公司发布的 Java 最初版本。自 Java 1.2 发布后, Java 改名为 Java2, 相应的 JDK 也就改名为 J2SE。

(2) Java SDK Micro Edition (J2ME)

J2ME 用于开发掌上电脑、手机等移动通信设备上使用的应用程序。现阶段并不是所有的移动设备都支持 Java, 只有具备 J2ME 运行环境 (JVM + J2ME API) 的设备才能运行 Java 程序。J2ME 的集成开发工具 (通常都带有一些仿真器) 有 Sun 的 J2ME Wireless Toolkit、IBM 的 Visual Age Micro Edition 等。

(3) Java SDK Standard Edition (J2SE)

J2SE 主要用于开发一般台式计算机应用程序。我们平时所说的 JDK 就是指 J2SE, 而我们学 Java 就是从学习 J2SE 开始的, 这也就是本书主要讲述的内容。

(4) Java SDK Enterprise Edition (J2EE)

J2EE 用于开发分布式的企业级大型应用程序。其中的核心是 Enterprise Java Beans (EJB, 分布式 Java 组件, 又称为企业 Java Bean) 的开发。

如果想了解更多关于 Java 的信息, 可以登录 Sun 公司的网站 (<http://www.sun.com>)。

1.5 Java 的前景

通过 Java 的简单发展史可以感受到, Java 的发展是随着网络及 Web 应用而发展壮大的。Java 使平淡、枯燥的网页变得有声有色, 使得声音、图形、图像和动画等被认为是难以在网络上动态交互的内容变得简单、稳定和丰富。可以说, 网络的应用促进了 Java 的诞生, Java 的诞生又同时推动了网络的发展, 两者是相辅相成的。

Java 语言从诞生到发展壮大, 可以说是得益于它在网络世界的出色表现, 没有网络, Java 也就失去了它的魅力。随着现代社会 Web 应用的普及、推广, Java 也越来越体现出它的能力及优点。

Sun 公司也在致力于加强 Java 的功能, 尤其在网络设备、网络通信, 以及 Web 上的应用能力。Java 的优势并不是开发已经存在的应用程序或对它们进行升级, 而是在它应用于新型设备或新应用领域中。

1.6 学好 Java 的关键

每一个走入 Java 世界的人都想学好 Java, 但是如果没有正确的指引, 虽然付出很多努力, 但收效甚微。所以, 掌握正确的学习方法, 不仅能提高学习效果, 更重要的是有利于为以后程序设计能力的提升打下良好的基础。

我们根据从事 Java 程序设计的实践经验及学习经历, 提出以下几点建议。

(1) 多动手

学习编程语言不仅仅是理论上的学习, 更重要的是要利用这门语言为读者的思想服务。理解并掌握这门语言是首要的, 但如果要达到心领神会、融会贯通, 就必须亲自动手, 多实践, 编一些具有特定功能的程序, 用实践去论证自己的思想。实践是检验真理的唯一标准, 所以建议大家多在实践中锻炼。在本书中, 我们会列举大量的示例, 包括演示、实例分析等, 希望读者在学习的过程中, 能将这些示例手动输入、编译和测试, 这样才能真正明白示例的含义。

(2) 多动脑

人类是最高级的动物, 之所以高级就是人有智慧、有思维, 尤其对于编程语言的学习, 更需要周密的逻辑思维。对于 Java 语言的学习, 不仅仅是对语言本身的学习, 更重要的是面向对象思想的建立过程, 如果想把 Java 学习提升到一个更高的层次, 建议大家从一开始就尝试用这种面向对象的思维方式去对待所接触的每一件事情。

(3) 多查阅 Java API 文档

Java 提供了大量的类, 以满足网络化、多线程和面向对象编程的需要。这就是 J2SE API, 它是以 HTML 的形式发布的, 读者可以把它下载到本地机上 (在第 2 章会详细介绍 J2SE API 的下载与安装), 也可以在线查找。它是 Java 编程的基本方法, 也是在编程过程中可以利用的重要资源。Java 的学习过程不仅仅是基本语法的学习, 其实更多的是去学习和掌握它所提供的 API 类库。我们建议读者在学习的过程中, 对于所接触到的每一个类和方法, 首先仔细去阅读文档说明, 然后再用编写的实例去调试、运行。

(4) 约束自己, 规范编写代码习惯

养成良好的编写代码习惯对于一个程序员来讲具有相当大的意义。一方面良好的编程习惯可减少编码过程中一些人为的错误；另一方面一段程序写得好坏，不仅是功能上的实现，更主要的是可读性、可维护性，没有任何人愿意去阅读一段没有顺序、杂乱无章的代码。当然，习惯的养成不是一日之功，所以建议大家在编写代码的时候要力争做到别人看得懂，条理清。

(5) 用有意义的名字

名字是一个标识，是一种有内涵的简单表述。我们强烈建议读者在编写程序的过程中，为每个类和方法起一个有意义的名字。在程序的运行过程中，看到这个名字就可以知道它所具备的功能。比如，编写一个学生类，我们可以将类命名为 `Student`，这样在阅读代码时就会知道它是关于学生的类。程序员在编写程序中一定要避免用 `AA`、`BB` 等类似于这种毫无内涵的名字命名类、方法、变量及常量。

(6) 添加适量的注释

注释不仅仅是对程序逻辑处理的一种注解，更多的是提高了程序的可读性或可维护性。作为一个软件产品，哪怕是一个小功能的实现，其中不同的变量及方法也可能很多，虽然我们强调在命名的过程中要使用具有内涵的名字，但也并不能完全涵盖变量及方法的功能及内涵，所以为了提高程序的可读性，必须添加一定量的注释。合理的注释不仅能美化程序，提高程序的可读性及可维护性，而且使程序看起来更加具有专业的味道。添加注释的格式将在后面的章节中学习。

(7) 多调试

一个好的程序员不仅能编写出好的代码，而且能解决代码编写过程中所遇到的任何问题。很少有程序员一次就能写出成功的代码，只有在不断地调试、修正中才能编写出真正好的代码。调试、解决问题的过程就是自己学习提高的过程。

1.7 小结

这一章中我们简单介绍了 Java 的诞生及特点，初步分析了面向对象的内涵，旨在使读者对于对象能有一个基本的概念，并对 Java 的前景进行了展望，最后根据我们的学习历程提出了几点供读者参考的学习建议。

1.8 练习

问答题

- (1) 什么是面向对象？
- (2) Java 语言有些什么特点？
- (3) Java 语言与 C、C++ 语言的区别？
- (4) 如何学好 Java 语言？

第 2 章 Java 编程环境

提到环境，大家都很熟悉，每个人都需要生活环境，我们的住房需要空间环境等。可以说，环境是保证我们日常生活的一个重要条件。

同样，程序也需要各种各样的环境，比如说测试环境、编译环境和运行环境等。这一章主要讲解 Java 编译环境的配置与建立。

Sun 公司为 Java 开发了适于不同平台的版本，各个平台所需要配置安装的文件也不一样。本章以 Windows 2000/XP 操作系统为例，介绍 Java 编译环境的配置与建立，其他平台的安装文件可以参阅 Sun 公司网站的相关信息。

2.1 J2SE 的安装与配置

安装 Sun 公司的 Java 标准开发工具箱是任何一个人走入 Java 大门的第一步。由于这个开发工具箱现仍在不断的升级中，请读者在阅读本书时，注意 J2SE 版本的变动，如果有最新版本，请下载最新版本并安装。

目前 J2SE 的版本是 5.0，也就是 1.5.0，比较稳定好用的是 J2SE 1.4.2，所以在本书中将采用 J2SE 1.4.2 的版本。

J2SE 1.4.2 在 Windows 系统下的安装过程非常简单，用鼠标双击，即可安装。随书 CD 中包含了一个适合于 Windows 平台的 J2SE 1.4.2 版本，如果读者需要其他的版本，请到 Sun 公司的网站去下载相应的平台版本。

注意

安装过程提供了一个默认的安装目录，其默认目录中包含了 Java SDK 版本号 D: \ j2sdk1.4.2_11，为了方便以后配置环境变量，笔者建议将它改为 D: \ j2sdk 1.4.2_11。也可以按照默认的目录进行安装。只要以后在设置有关环境变量时注意不要拼写错误就可以了。本书中的安装目录是 D: \ j2sdk1.4.2_11。

安装完成的目录如图 2-1 所示。

在表 2-1 中对每一个目录进行了说明。

表 2-1 Java 目录树

目 录	说 明
bin	编译器及一些工具
demo	演示程序
include	用于本地方法的文件，现在基本不用
jre	Java 运行环境文件
lib	类库文件
src	公开库源文件