



高等院校计算机类课程“十二五”规划教材

Java程序设计

Java CHENGXU SHEJI

主 编◎陈 锐 李绍华

副主编◎曾贤灏 魏 东 冯晶莹



可下载教学资源

<http://www.hfutpress.com.cn>
<http://blog.csdn.net/crcr>
nwuchenrui@126.com



清华大学出版社
Tsinghua University Press

Java 程序设计

孙亚非 郎薇薇 安强强

内容简介

本书内容包括 Java 语言概述、Java 语言开发环境、数据类型与运算符、流程控制、数组、字符串、面向对象编程基础、继承和多态、接口和抽象、异常与处理、Java Applet 编程、图形用户界面设计、JDBC 及其应用、多线程机制、网络编程和 Java 中的数据结构等。

本书内容全面,结构清晰,语言通俗流畅,重难点突出,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 程序设计课程的教材,也可作为计算机软件开发、全国计算机等级考试和软件资格(水平)考试人员的参考用书。

图书在版编目(CIP)数据

Java 程序设计/陈锐,李绍华主编. —合肥:合肥工业大学出版社,2012. 10

ISBN 978 - 7 - 5650 - 0951 - 8

I. ①J… II. ①陈…②李… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 244550 号

Java 程序设计

陈 锐 李绍华 主编

责任编辑 汤礼广 魏亮瑜

出 版 合肥工业大学出版社
地 址 合肥市屯溪路 193 号
邮 编 230009
电 话 理工编辑部:0551-2903087
市场营销部:0551-2903163
网 址 www.hfutpress.com.cn
E-mail hfutpress@163.com

版 次 2012 年 10 月第 1 版
印 次 2012 年 10 月第 1 次印刷
开 本 787 毫米×1092 毫米 1/16
印 张 26.5
字 数 627 千字
印 刷 合肥现代印务有限公司
发 行 全国新华书店

ISBN 978 - 7 - 5650 - 0951 - 8

定价: 49.00 元

如果有影响阅读的印装质量问题,请与出版社市场营销部联系调换。

前言

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

目前, 大多数高等院校不仅计算机专业开设了 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 程序设计课程的教学工作, 并长期致力于数据结构与算法、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, 对程序设计有独特体会, 书中很多内容均打上作者教学经验和科研成果的烙印。本书特色较为明显。

本书的特点

1. 内容全面, 讲解详细

本书首先介绍了 Java 语言的发展历史与特点、开发环境的安装与使用, 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, 几乎覆盖了 Java 语言的全部基础知识, 且对于每个知识点都结合具体示例进行讲解, 以方便学生理解。

2. 结构清晰, 内容合理

本书按照章、节和小节划分知识点, 再将每个知识点进行细化, 做到了结构框架清晰、内容讲解合理, 易于学生理解与学习。在讲解知识点时, 循序渐进, 由浅入深, 最后通过示例强化知识点, 这样的讲解方式的好处是使学生更容易理解和消化。

3. 语言通俗, 叙述简单

本书采用通俗流畅的表述形式讲解每个知识点, 力求避免使用晦涩难懂的语句, 目的是使学生避开障碍, 轻松学习。

4. 示例典型, 深入剖析

在讲解每一个知识点时, 不仅结合具体示例进行剖析, 而且在示例的选取方面, 尽力选取一些最为常见且能涵盖知识点的典型程序。

本书的内容

第 1 章: 首先介绍了 Java 语言的发展历史与特点、面向对象基础知识, 并介绍了 Java 语言的发展前景。

第 2 章: 介绍了 JDK 的安装与配置, 并分析了第一个 Java 语言程序。

第3章：主要介绍了基本数据类型（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：主要介绍了条件语句、循环语句、break语句和continue语句。

第5章：主要介绍了一维数组、二维数组的概念，数组的基本操作，还介绍了排序算法。

第6章：主要介绍了String类、StringTokenizer类、StringBuffer类、Character类及字符串的相关操作。

第7章：主要介绍了面向对象编程基础。首先介绍了对象和类的概念，然后介绍了包的概念。

第8章：主要介绍了继承和多态。首先介绍了继承的概念，然后介绍了类与类之间的关系，最后介绍了多态概念。

第9章：主要介绍了接口和抽象。首先介绍了接口的产生、接口的声明与实现、接口的继承，然后介绍了浅拷贝和深拷贝，最后介绍了内部类与抽象。

第10章：主要介绍了异常与处理。

第11章：主要介绍了Java Applet小程序的开发过程。

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

第13章：主要介绍了JDBC及其应用。首先介绍了JDBC相关技术，然后介绍了如何查询、建表、创建数据库。

第14章：主要介绍了Java的多线程机制。首先介绍了Java的线程概念、多线程的实现方法，最后介绍了线程的控制。

第15章：主要介绍了网络编程。首先介绍了网络编程的概念，然后介绍了基于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。

第16章：首先介绍了组织数据结构的两个接口——Collection和Iterator，然后介绍了链表、散列表、树集、Vector类。

本书由陈锐（高级程序员）、李绍华（大连外国语学院）主编，曾贤灏（兰州工业学院）、魏东（海军航空兵学院）、冯晶莹（辽宁警官高等专科学校）担任副主编，张西广（中原工学院）、卢香清（南阳师范学院）、唐红杰（辽宁警官高等专科学校）、孙亚非（沧州职业技术学院）、郎薇薇（北京信息职业技术学院）、安强强（榆林学院）参编。全书由陈锐负责统稿。

由于作者水平有限，加之时间仓促，错误和疏漏之处在所难免，恳请广大读者批评指正。在使用本书的过程中，如遇任何问题，或想索取本书的例题代码，请从博客 <http://blog.csdn.net/crcr> 下载，或通过电子邮件 nwuchenrui@126.com 索取。

编者

目 录

第 1 章 Java 语言综述	(1)
1.1 Java 语言发展史	(1)
1.2 面向对象的初步介绍	(2)
1.3 Java 语言的特点	(4)
1.4 Java 虚拟机	(6)
1.5 Java 的不同版本	(7)
1.6 Java 的发展前景	(8)
1.7 学好 Java 的关键	(8)
第 2 章 Java 语言开发环境	(11)
2.1 JDK 的安装配置	(11)
2.2 环境变量的配置与测试	(13)
2.3 第一个 Java 程序	(17)
2.4 命名与保存	(22)
2.5 Java 程序的开发流程	(25)
第 3 章 数据类型与运算符	(30)
3.1 数据类型	(30)
3.2 变量	(37)
3.3 基本类型变量的初始化	(45)
3.4 常量	(49)
3.5 运算符和表达式	(51)
第 4 章 流程控制	(72)
4.1 作用域	(72)
4.2 条件语句	(77)
4.3 循环语句	(87)

4.4	break 语句与 continue 语句	(96)
4.5	return 语句	(100)
第 5 章	数组	(105)
5.1	数组的概念	(105)
5.2	一维数组	(105)
5.3	二维数组	(112)
5.4	数组的基本操作	(114)
5.5	排序	(119)
第 6 章	字符串	(126)
6.1	String 类	(126)
6.2	StringTokenizer 类	(133)
6.3	StringBuffer 类	(135)
6.4	Character 类	(137)
6.5	字符串与字符数组、字节数组	(139)
第 7 章	面向对象编程基础	(144)
7.1	对象	(144)
7.2	类	(146)
7.3	使用已有的类	(168)
7.4	包	(177)
7.5	注释及嵌入文档	(183)
第 8 章	继承和多态	(196)
8.1	继承的概念	(196)
8.2	继承的实现	(197)
8.3	构造函数的调用	(211)
8.4	类之间的关系	(213)
8.5	设计好继承的几点建议	(215)
8.6	final	(215)
8.7	多态	(216)
8.8	超类	(223)
8.9	数组列表	(225)
8.10	对象包装器	(232)

第 9 章 接口和抽象	(235)
9.1 接口	(235)
9.2 克隆	(248)
9.3 内部类	(259)
9.4 抽象行为	(278)
9.5 抽象类	(279)
9.6 抽象与接口的区别	(284)
第 10 章 异常与处理	(286)
10.1 异常处理	(286)
10.2 异常的捕获	(293)
10.3 finally 子句	(299)
10.4 自定义异常	(301)
第 11 章 Java Applet 编程	(305)
11.1 编写 Applet 小程序	(305)
11.2 Applet 的执行过程	(307)
第 12 章 图形用户界面设计	(311)
12.1 AWT 工具集简介	(311)
12.2 AWT 基本组件	(313)
12.3 AWT 容器	(321)
12.4 布局管理器	(324)
12.5 AWT 事件处理模型	(331)
12.6 Swing 简介	(344)
第 13 章 JDBC 及其应用	(348)
13.1 JDBC 基本编程概念	(348)
13.2 预查询	(367)
第 14 章 多线程机制	(369)
14.1 Java 中的线程	(369)
14.2 通过继承 Thread 类实现多线程	(371)
14.3 通过 Runnable 接口实现多线程	(373)

14.4	线程的调度	(374)
14.5	线程的同步	(376)
第 15 章	网络编程	(381)
15.1	网络编程的基本概念	(381)
15.2	基于 URL 的 Java 网络编程	(383)
15.3	基于套接字的 Java 网络编程	(387)
15.4	数据报	(391)
第 16 章	Java 中的数据结构	(397)
16.1	Collection 接口与 Iterator 接口	(397)
16.2	链表	(400)
16.3	散列表	(404)
16.4	树集	(407)
16.5	Vector 类	(411)
参考文献		(415)

第1章

Java 语言综述

■ 本章导读

Java 语言自诞生到现在,已经成为目前流行的网络编程语言之一,是当今计算机业界不可忽视的力量,代表着重要的发展潮流与方向。在学习 Java 语言之前,要先了解 Java 语言的发展历程及特点,初步建立面向对象的概念,从而对面向对象的核心思想有个较清晰的认识。通过本章学习,读者可以全面认识 Java 语言,并了解学习 Java 语言应该注意的事项。

■ 学习目标

- (1) 了解 Java 语言及其发展史;
- (2) 了解面向对象的概念;
- (3) 掌握 Java 语言的基本特点;
- (4) 掌握学好 Java 语言的关键。

1.1 Java 语言发展史

Java 语言是由 Sun 公司推出的。早在 1991 年, Sun 公司的 Patrick Naughton 和 James Gosling 就成立了 Green 项目组,专门为消费类电子产品开发分布式代码系统。该项目组的研究是以 C++ 为基础的,但 C++ 太复杂,且安全性差,最后开发了基于 C++ 的一种新语言 Oak (Java 的前身),Oak 是一种精巧而安全的网络语言。

1994 年下半年,Internet 的快速发展促进了 Java 语言研制的进展。Green 项目组成员用 Java 编制了 HotJava 浏览器,发起了 Java 向 Internet 的进军,使其逐渐成为最受欢迎的开发语言与编程语言。现在,Java 已发展成为一种重要的 Internet 平台。

1995 年,以 James Gosling 为首的编程小组在 wicked.neato.org 网站上发布了 Java 技术,这门语言的名字也从 Oak 变为 Java。之所以叫做 Java,是因为印度尼西亚

有一个盛产咖啡的重要岛屿，中文译名为爪哇。而这门语言的开发者为其起名为 Java，寓意是为世人端上一杯热气腾腾的咖啡。所以，Sun 公司有关 Java 的产品上都会有一杯冒着热气的咖啡卡通图标。Java 技术一经发布，便被美国著名杂志《PC Magazine》评为 1995 年十大优秀科技产品之一。

1996 年，Sun 公司发布了包括运行环境和开发工具的 JDK (Java Development Kit) 1.0，之后陆续发布新的版本 JDK 1.1、JDK 1.2 等。

1998 年，Sun 公司发布了 Java 2。Java 2 是应用 Java 最新技术的核心品牌。JDK 1.2 支持 Java 2 技术，自 JDK 1.2 之后的 JDK 版本正式更名为 Java 2 SDK，由于多数程序员已经习惯了 JDK 的名字，所以可以将 Java 2 SDK 称为 JDK。

2010 年，发布了 JDK 1.7 (Oracle 官方称为 JDK 7)，增加了简单闭包功能。

2011 年，甲骨文公司发布 Java 7 的正式版。

Java 是一门优秀的程序设计语言，具有面向对象、与平台无关、安全、稳定和多线程等特点，是目前软件设计中极为健壮的编程语言之一。Java 语言被称为革命性的编程语言，这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 编写的程序却不必考虑这些问题。Java 语言在所有平台上的字节码是兼容的，即只要提供 Java 解释器，Java 编写的程序就能在计算机上运行，从而实现了 Java 语言编写者“一次编写，处处运行”的口号。

Java 不仅可以用来开发大型的应用程序，而且特别适合于 Internet 的应用开发。Java 已成为网络时代最重要的编程语言之一，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 曾有一些过火的宣传，但现在看来，Java 真正的用武之地应当是嵌入式开发。Java 自正式发布至今已经变得相当稳定、强健，现在它的类库仍然在不断地壮大、扩展中，相信这也必将使 Java 在网络世界的应用变得更加广泛。

1.2 面向对象的初步介绍

1.2.1 面向对象程序设计的分类

Java 是一种面向对象的编程语言，要想真正掌握它，首先必须明确什么是面向对象以及面向对象的核心思想。近几年来，面向对象编程在软件开发领域掀起了一阵狂热的风潮，得到迅猛发展的同时也受到越来越多的关注，同时也吸引了越来越多的人加入到程序开发的行列中。随着面向对象程序设计方法的提出，出现了不少面向对象的程序设计语言，如 Java、C++ 等。这些语言大致可分为以下两类：

(1) 开发全新的面向对象程序设计语言。其中，最具代表性的就是 Java、Smalltalk 和 Eiffel。Java 语言适合网络编程，Smalltalk 语言完整地体现了面向对象程序设计的核心思想，而 Eiffel 语言除了具有封装和继承之外，还具有了其他面向对象的特征，是一种很好的面向对象程序设计语言。

(2) 对传统语言进行面向对象程序设计语言扩展的语言。这类语言又称为“混合

型语言”，一般是在其他语言的基础上加入面向对象程序的概念开发出来的，最典型的就是 C++。

任何事物都能抽象为对象，面向对象程序设计是以对象为模型描述现实世界。面向对象程序设计（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；对于程序设计分析人员来说，利用 UML 建模更加直观、方便，完成的程序也更易于维护；对于程序员来说，能更快地理解并领会设计人员的意图，使开发过程变得不再枯燥无味。

在我们利用面向对象时，也必须为掌握它而付出努力。因为思考对象的过程需要用抽象思维，而不再是程序化的思维，这种思想的形成需要长时间的努力和实践。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，具体的过程是将对象实例化。

1.2.2 面向过程与面向对象的区别

在 Java 诞生以前，开发软件的方式基本上都是面向过程的。接下来介绍面向过程与面向对象的区别。

我们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：

(1) 黑白双方，这两方的行为是一模一样的；(2) 棋盘系统，负责绘制画面；(3) 规则系统，负责判定诸如犯规、输赢等操作。

玩家对象负责接受用户输入，并告知棋盘对象棋子的布局变化。棋盘对象接收到了棋子的变化就要在屏幕上面显示出这种变化，同时利用规则系统对棋局进行判定。

对于绘制棋局，在面向过程的设计中要分散在许多步骤中进行，很可能出现不同的绘制版本，因为设计人员通常会考虑到实际情况进行各种各样的简化；而在面向对象的设计中，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。例如要加入悔棋功能，若在面向过程的设计中，那么从输入、判断到显示这一连串的步骤都要改动，甚至步骤之间的顺序都要进行大规模地调整；而在面向对象的设计中，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，只需简单回溯即可。此时，其他功能不用修改，改动的只是局部。

当然，通过这几段文字不可能完全涵盖面向对象的思想，只是想让大家对面向对象有个初步认识，并在以后的学习中尝试着用面向对象的思维方式去思考问题、分析问题。

1.2.3 对象的概念

任何事物都可以看成是一个对象，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；一类东西也可以作为一个对象，如砖和积木。可以把不同形状的砖或积木作为一个对象；也可以把多个相同形状的砖或积木作为一个对象；还可以把所有的砖或积木作为一个对象。

对象的概念是面向对象技术的核心所在。以面向对象的观点来看，所有面向对象的程序都是由对象组成的。更确切地说，对象就是现实世界中某个具体的物理实体在计算机逻辑中的映射和体现。

1.2.4 对象的属性和行为

对象既具有静态属性又有动态行为。例如，每个人都有自己的姓名、年龄、身高等属性，又有吃饭、走路等行为。对象是由对象名、一组属性数据和一组操作封装在一起构成的实体。

对象的状态又称为对象的静态属性，主要指对象内部包含的各种信息，也就是变量。每个对象个体都具有自己专有的内部变量，这些变量的值标明了对象所处的状态。比如，每台电视机都具有以下信息：种类、品牌、尺寸、外观、颜色、所在频道等。当对象经过某种操作和行为而状态发生改变时，具体地就体现在它属性的改变。

行为又称为对象的操作，操作的作用是改变或设置对象的状态。比如，一台电视机可以有打开、关闭、调节音量、改变频道等行为或操作。

1.3 Java 语言的特点

Java 语言是一门很优秀的编程语言，它最大的优点是与平台无关，在 Windows 9x、Windows NT、Solaris、Linux、Mac OS 以及其他平台上，都可以使用相同的代码。Java 语言与 C++ 语言的语法结构十分相似，因此 C++ 程序员学习 Java 语言非常容易。

Java 语言的规范是公开的，可以在 <http://www.sun.com> 上找到，阅读 Java 语言的规范是提高技术水平的好方法。Java 语言的主要特点如下所示：

1. 简单性

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 语言中省去了运算符重载、多重继承等模糊概念，并且提供自动垃圾收集机制，大大简化了程序设计者的内存管理工作。

2. 面向对象

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类提供的方法，从而实

现了代码的复用。

3. 分布性

Java 是面向网络的语言。用户可以通过 Java 提供的类库处理 TCP/IP 协议,也可以通过 URL 地址在网络上很方便地访问其他对象。

4. 健壮性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾收集机制来进行内存管理,防止程序员在管理内存时产生错误。在编译时,通过集成的面向对象的例外处理机制提示可能出现但未被处理的例外,帮助程序员正确地进行选择,防止系统崩溃。另外,Java 还可捕获类型声明中许多常见错误,防止动态运行时不匹配问题的出现。

5. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员,同时也避免了在指针操作中产生错误。

6. 体系结构中立

体系结构中立也称平台无关。Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行系统,Java 程序就可以在任意处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示,Java 解释器得到字节码后,对它进行转换,使之能够在不同的平台运行,这也成为 Java 应用软件便于移植的良好基础。

7. 可移植性

Java 程序不必重新编译就能在任何平台上运行,具有很强的可移植性。同时,Java 的类库也实现了与不同平台的接口,从而使这些类库可以移植。另外,Java 编译器是由 Java 语言实现的,Java 运行时系统由标准 C 实现,这使得 Java 系统本身也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。Java 程序被编译成 Java 虚拟机 (Java Virtual Machine, JVM) 编码,即字节码。字节码本身携带了许多编译时的信息,它能够在任何具有 Java 解释器的机器上运行,连接过程更加简单。

9. 高性能

因为 Java 是解释型的,字节码不在系统上直接运行,而是通过解释器运行,所以字节码的运行速度较之 C++ 之类的编译语言稍逊。但随着 Sun 公司对 Java 技术的改进,Java 虚拟机的运行速度也在不断提高。Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码,从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操

作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样可以很容易地实现网络上的实时交互行为。

11. 动态性

Java 的设计使它适应一个不断发展的环境,在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。而且,Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。例如,对于一个 Circle 类来说,可以增加表示颜色的新属性,也可以增加表示获取周长的新方法。

基于以上这些特点,Java 语言备受程序员的青睐。

1.4 Java 虚拟机

Java 程序可以利用记事本编写,并保存为扩展名为 .java 的文件。经过编译后,生成 .class 文件,该文件为字节码文件。此时的 .class 文件并不能被处理器直接运行,必须通过虚拟机转换为二进制文件后才能被处理器运行。Java 程序的运行过程如图 1-1 所示。

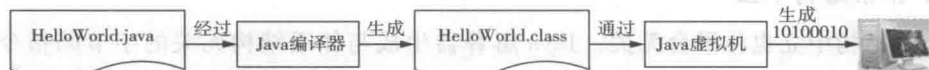


图 1-1 Java 程序在虚拟机上的运行过程

Java 是跨平台的高级编程语言,这里的平台是指操作系统平台,如 Windows、UNIX、Linux、Mac 等。正因为有 Java 虚拟机的存在,Java 语言编写的程序一次编译后才可以在上述所有平台上运行。Sun 提供了在各种操作系统平台上运行安装的 Java 虚拟机,虚拟机都可以执行 .class 文件。通过 Java 虚拟机,在不同平台上执行 Java 程序的过程如图 1-2 所示。

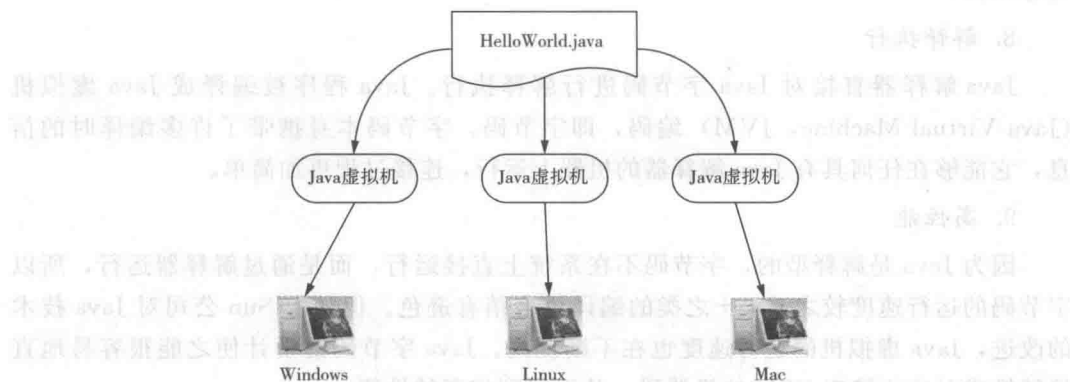


图 1-2 不同平台上通过虚拟机执行 Java 程序

注意：

不同平台上的 Java 虚拟机 (JVM) 是不同的, 可以根据需要到 Sun 网站下载适合自己机器操作系统平台的虚拟机。

作为软件平台的 Java 由下面两部分组成:

Java 虚拟机 (Java Virtual Machine, JVM): Java 虚拟机是 Java 软件平台的基础。基于不同的操作系统平台, 有相应的虚拟机。

Java 应用程序编程接口 (Java API): API 是一套编写好的软件组件集合。这些 API 按照功能通过包 (package) 来提供, 包是一系列具有相似功能的类和接口的集合。

Java 虚拟机和应用程序编程接口与各种操作系统的层级关系如图 1-3 所示。

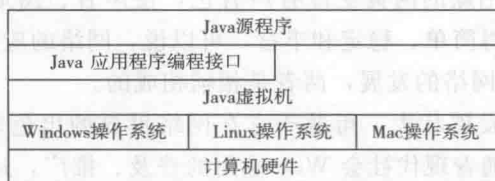


图 1-3 Java 虚拟机和 API 与操作系统之间的关系

说明：

Java 虚拟机使 Java 实现了平台的独立性, 这也使 Java 程序比 C 或 C++ 语言编写的程序运行速度要慢一些, 但是随着编译器技术的不断发展, 两者之间的差距会越来越小。

1.5 Java 的不同版本

Sun 公司为了实现“一次编写, 处处运行”的目标, 提供了相应的 Java 运行平台。目前, Java 主要有 3 种版本。

1. Java 2 Platform Standard Edition (J2SE)

自 Java 1.2 版本发布后, Java 改名为 Java 2, 相应的 JDK 也改名为 J2SE。J2SE 是 Java 2 标准版, 主要用于开发一般台式机应用程序。我们平时所说的 JDK 就是指 J2SE, 学习 Java 也是从学习 J2SE 开始, 这是本书主要讲述的内容。

2. Java 2 Platform Micro Edition (J2ME)

J2ME 是 Java 2 微型版, 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 2 Platform Enterprise Edition (J2EE)

J2EE 是 Java 2 企业版,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 公司的官方网站 <http://www.sun.com>。

1.6 Java 的发展前景

通过简单了解 Java 的发展史, 可以感受到 Java 的发展是随着网络及 Web 应用而壮大的。Java 使平淡、枯燥的网页变得有声有色, 使声音、图形、图像和动画等内容在网络上的动态交互变得简单、稳定和丰富。可以说, 网络的应用促进了 Java 的诞生, 而 Java 的诞生又推动了网络的发展, 两者是相辅相成的。

Java 语言从诞生到发展壮大, 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 也越来越展现出它的能力及优点。Java 语言的前景可以概括成以下几点:

- (1) 所有面向对象的应用开发, 包括面向对象的事件描述、处理和综合等。
- (2) 计算过程的可视化、可操作化的软件开发。
- (3) 动态画面的设计, 包括图形、图像的调用。
- (4) 交互操作设计。
- (5) Internet 的系统管理功能模块, 包括 Web 页面的动态设计、管理和交互操作设计等。
- (6) Internet 上的软件开发。
- (7) 与各类数据库连接、查询的 SQL 语句的实现。
- (8) 其他应用类型的程序。

1.7 学好 Java 的关键

Java 虽说是一门简单易学的语言, 但是要想学好这门语言, 还是需要花一些精力、注意一些问题的。下面介绍几点学好 Java 的注意事项。

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, 用实践去验证自己的思想。在本书中, 我们会列举大量的示例, 包括演示、示例分析等, 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。