



全国信息技术人才培养工程指定培训教材  
Web应用开发工程师职业教育系列规划教材

# Java语言程序设计

工业和信息化部教育与考试中心 组编  
清华大学计算机与信息管理中心 主编

严谨

勤奋

求实

创新



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

全国信息技术人才培养工程指定培训教材  
Web 应用开发工程师职业教育系列规划教材

# Java 语言程序设计

工业和信息化部教育与考试中心 组编  
清华大学计算机与信息管理中心 主编

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书系统介绍了 Java 语言与面向对象程序设计的概念、方法与基本技术。书中首先介绍 Java 语言的特点和基础语法知识；然后介绍本书的核心内容——面向对象程序设计的主要技术与编程思路，其中包括类与对象、属性与方法的定义及应用。本书的后半部分从应用出发，讲述 Java 编程的几个重要专题，其中包括 Java 语言基础类库、异常处理、输入/输出以及数据库编程接口等。本书附录介绍 Java 编程环境，并重点介绍业界广泛使用的集成开发环境 Eclipse。

本书从 Web 应用开发的需求出发，省略了在编程中很少使用的一些内容（如图形界面及多线程等），而用较多的篇幅介绍了 Java 面向对象的核心技术。

本书的讲解条理清楚，内容深浅适中，并特别注重提高读者运用 Java 语言和面向对象技术解决问题的能力。书中给出了大量经过调试运行的实例，以便于初学者入门。

作为 Web 应用开发的系列教材之一，本书适合作为高职高专院校 Java 程序设计课程的教材，也可作为 Java 语言的社会培训教材及 Java 语言爱好者的自学用书。

### 图书在版编目（CIP）数据

Java 语言程序设计/清华大学计算机与信息管理中  
心主编. —北京：中国铁道出版社，2010.1  
（全国信息技术人才培养工程指定培训教材）  
（Web 应用开发工程师职业教育系列规划教材）  
ISBN 978-7-113-10951-6

I. ①J… II. ①清… III. JAVA 语言—程序设计—  
教材 IV. ①TP312

中国版本图书馆 CIP 数据核字（2010）第 001788 号

书 名：Java 语言程序设计

作 者：工业和信息化部教育与考试中心 组编  
清华大学计算机与信息管理中心 主编

策划编辑：秦绪好 王春霞

责任编辑：翟玉峰

编辑部电话：（010）63560056

编辑助理：邱雪姣

封面设计：付 巍

封面制作：李 路

责任印制：李 佳

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号 邮政编码：100054）

印 刷：北京市兴顺印刷厂

版 次：2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷

开 本：787mm×1092mm 1/16 印张：20 字数：492 千

印 数：3 000 册

书 号：ISBN 978-7-113-10951-6

定 价：32.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签，无标签者不得销售

凡购买铁道版图书，如有印制质量问题，请与本社计算机图书批销部联系调换

当今世界，随着信息技术在经济社会各领域不断深化的应用，信息技术对生产力以至于人类文明发展的巨大作用越来越明显。党的“十七大”提出要“全面认识工业化、信息化、城镇化、市场化、国际化深入发展的新形势新任务”，“发展现代产业体系，大力推进信息化与工业化融合”，明确了信息化的发展趋势，首次鲜明地提出了信息化与工业化融合发展的崭新命题，赋予了我国信息化全新的历史使命。

近年来，日新月异的信息技术呈现出新的发展趋势，信息技术与其他技术的结合更加紧密，信息技术应用的深度、广度和专业化程度不断提高。我国的信息产业作为国民经济的支柱产业正面临着有利的国际、国内形势，电子信息产业的规模总量已进入世界大国行列。但是我们也清楚地认识到，与国际先进水平相比，我们在产业结构、核心技术、管理水平、综合效益、普及程度等方面，还存在较大差距，缺乏创新能力与核心竞争力，“大”而不强。国际国内形势的发展，要求信息产业不仅要做大，而且要做强，要从制造大国向制造强国转变，这是信息产业今后的重点工作。要实现这一转变，人才是基础。机遇难得，人才更难得，要抓住本世纪头二十年的重要战略机遇期，加快信息行业发展，其关键在于培养和使用好人才资源。《中共中央、国务院关于进一步加强人才工作的决定》指出，人才问题是关系党和国家事业发展的关键问题，人才资源已成为最重要的战略资源，人才在综合国力竞争中越来越具有决定性意义。

为抓住机遇，迎接挑战，实施人才强业战略，原信息产业部于2004年启动了“全国信息技术人才培养工程”。根据工业和信息化部人才工作要点关于“继续组织实施全国信息技术人才培养工程”的要求，工业和信息化部教育与考试中心将继续推进全国信息技术人才培养工程二期工作的开展。该项工程旨在通过政府政策引导，充分发挥全行业和社会教育培训资源的作用，建立规范的信息技术教育培训体系、科学的培训课程体系、严谨的信息技术人才评测服务体系，培养造就大批行业急需的、结构合理的高素质信息技术应用型人才，以促进信息产业持续快速协调健康发展。

根据信息产业对技术人才素质与能力的需求，在充分吸取国内外先进信息技术培训课程优点的基础上，工业和信息化部教育与考试中心组织各方专家精心编写了信息技术系列培训教材。这些教材注重提升信息技术人才分析问题和解决问题的能力，对各层次信息技术人才的培养工作具有现实的指导意义。我们谨向参与本系列教材规划、组织、编写的同志们致以诚挚的感谢，并希望该系列教材在全国信息技术人才培养工作中发挥有益的作用。

工业和信息化部教育与考试中心

2009年10月

面向对象技术被称为程序设计方法学的一场革命，它已经逐步替代了面向过程的程序设计技术，成为软件开发的主流技术。面向对象技术比较符合人们观察世界和处理问题的思维模式，而将数据与处理数据的操作封装在一起的机制也符合现代大规模软件开发的要求，并易于实现软件的复用。

进入 21 世纪以来，社会信息化的进程明显加快。而社会信息化的显著标志就是计算机信息系统在各行各业得到了普及。电子政务、电子商务等术语对大家来说已不陌生，基于 Web 的应用系统得到广泛应用。就大家所熟悉的大学校园来说，教务管理系统、选课系统、网络教学系统等在学校教学活动中发挥了重要作用。而开发这些信息系统的主流技术是基于 Java 语言及其技术架构的。

Java 语言是面向对象技术成功应用的典范。诞生于 1995 年的 Java 语言在短短的几年间便席卷全球。

由于 Java 语言所具有的简洁性、纯面向对象、跨平台等特征，使得它不但在 Web 应用开发中得到广泛应用，而且也非常适合高校中面向对象程序设计的教学。所以，本书从 Web 应用开发的实际需求出发，并考虑到面向对象程序设计的教学要求，对教学内容的选取、编排及例题（案例）设计都做了仔细的斟酌，确保全书深度和广度适中，并遵循由浅入深、循序渐进的组织原则。本书适合作为高职高专院校 Java 程序设计课程的教材，也可作为 Java 语言学习者的自学用书。

学习本书之前应该对计算机操作有一定的认识，但不必具有编程经验。

下面简要介绍一下本书的主要内容与教学安排。

**第 1 章 Java 语言概述。**介绍 Java 语言的特点、Java 程序基本开发环境，并通过几个简单示例使读者对 Java 程序有一个感性认识。本章介绍 Java 的入门知识。

**第 2 章 Java 语言基础。**系统介绍了 Java 语言的语法特征，并依次介绍 Java 的数据类型、表达式和流程控制语句，为后续章节的学习提供了语言编程基础。本章还介绍了两个重要的数据类型——数组与字符串。但在 Java 语言中，它们都是建立在类与对象的基础上。本章预先介绍它们是为了便于讲解流程控制语句，也便于在随后章节的例题中使用。在学完面向对象程序设计的有关内容之后，读者可进一步加深对数组与字符串的理解。

**第 3 章 Java 面向对象程序设计（上）。**面向对象程序设计的内容属本书核心内容。因知识点较多，所以将其分在两章中讲解。本章以 Java 面向对象编程为主线，首先介绍面向对象程序设计的基本思想，然后介绍类与对象的基本内容，包括类的定义、类与对象的使用等。本章还将详细讨论面向对象技术的几个重要特征，例如继承和多态的概念及实现方法。

**第 4 章 Java 面向对象程序设计（下）。**继上一章之后，本章继续介绍面向对象程序设计的核心内容，其中包括静态变量和静态方法、抽象类和抽象方法、最终类和最终方法等。本章还将介绍 Java 中用于解决多重继承及编程规范的“接口”概念及应用。本章最后介绍编程中常用到的包及修饰符的有关概念。

通过 3、4 章的学习，读者可以理解并初步掌握 Java 面向对象编程技术。

要掌握好 Java 语言并具有利用它解决实际问题的能力，仅仅学习语法规则是不够的，还需要

掌握 Java 的应用程序编程接口,即 Java 的类库。本书从第 5 章开始介绍 Java 的常用标准类库及一些重要的编程技术。

第 5 章 Java 语言基础类库。在实际编程中,用户不但需要抽象、定义自己的类,还应该学会如何充分利用系统或开发环境中提供的类。而对 Java 基础类库的了解及其中常用类的使用无疑是最基本的要求。本章介绍了 Java 类库的基本情况,并依次介绍了 4 个常用的包及其中部分常用类的含义与应用。

第 6 章 Java 程序中的异常处理。程序的安全性和健壮性是 Java 语言设计的重要目标之一。Java 程序通过异常处理机制,加强了程序应对各种复杂错误情况的处理能力,使程序的安全性与稳定性得到加强。本章介绍 Java 的异常处理机制,包括异常的概念、异常的定义、异常的处理等内容。

第 7 章 Java 程序的输入/输出。输入/输出是计算机的基本操作,程序通过输入/输出与外部信息进行交互。Java 采用“输入/输出流”实现了 I/O 操作,它的特点是从“流”读取数据或者向“流”写入数据。Java 语言的 java.io 包中封装了丰富的 I/O 流类,供程序员实现 I/O 操作。Java 流类具有清晰的继承关系,使用方法也是类似的。通过学习最基本的输入/输出流类,可以掌握各种流类的基本使用方法。

第 8 章 Java 数据库编程接口。本章首先介绍了数据库的基础知识及 SQL 语言,以及在 Microsoft Access 中建立简单的实例数据库的方法,然后介绍了 Java 数据库编程接口——JDBC,这是编写数据库应用程序的基础。

第 9 章 综合实例。本章通过一个具体的实例,综合运用并复习前面所介绍的知识,其中包括 Java 语言基础知识、类的定义与继承、Java 类库、异常处理、输入/输出、数据库操作等内容。通过这个实例,读者能够了解运用 Java 语言开发实用程序的一般流程。

由于本书未涉及用户图形界面,因此仍以“命令提示符”作为操作界面。

程序设计课程是一门实践性很强的课程。读者只有在学习书本内容的同时辅以相应的实际练习和实验环节,才能真正掌握书中介绍的知识和技能。为此,本书引入了大量的例题,并配有习题解答和上机实验指导书。只要读者能够按照书中的要求边学边练,就一定能很快登堂入室,在 Java 语言和面向对象技术所构造的无限畅想空间中享受遨游的乐趣。

本教材由清华大学计算机与信息管理中心主编,参加编写的人员有:龙明盛、王行言、武莹、王强等。

由于作者水平所限,书中难免有疏漏之处,欢迎各位同行和广大读者对本书提出建议和修改意见。

编 者

2010 年 2 月 于清华园



|       |                          |    |
|-------|--------------------------|----|
| 第 1 章 | Java 语言概述                | 1  |
| 1.1   | Java 语言的形成与发展            | 1  |
| 1.2   | Java 语言特点                | 2  |
| 1.3   | Java 程序工作方式              | 3  |
| 1.4   | JDK 实用程序                 | 4  |
| 1.5   | Java 程序类型                | 7  |
| 1.5.1 | Java Application 程序      | 7  |
| 1.5.2 | Java Applet 程序           | 11 |
| 1.5.3 | Java 交互式程序               | 13 |
| 1.6   | Java 语言编码规范              | 14 |
|       | 小结                       | 16 |
|       | 习题                       | 16 |
| 第 2 章 | Java 语言基础                | 17 |
| 2.1   | Java 程序结构                | 17 |
| 2.2   | 表达式规则                    | 19 |
| 2.2.1 | 标识符                      | 19 |
| 2.2.2 | 基本数据类型 (primitive types) | 21 |
| 2.2.3 | 变量、常量与文字量                | 22 |
| 2.2.4 | 数据类型转换                   | 26 |
| 2.2.5 | 基本数据类型包裹类                | 29 |
| 2.2.6 | 引用类型 (reference types)   | 31 |
| 2.2.7 | 运算符                      | 33 |
| 2.2.8 | 表达式和语句                   | 40 |
| 2.3   | 流程控制语句                   | 42 |
| 2.3.1 | 结构化编程                    | 42 |
| 2.3.2 | 条件语句                     | 43 |
| 2.3.3 | 循环语句                     | 46 |
| 2.3.4 | 转移语句                     | 51 |
| 2.4   | 数组与字符串                   | 54 |
| 2.4.1 | 数组                       | 54 |
| 2.4.2 | 字符串                      | 58 |
|       | 小结                       | 63 |
|       | 习题                       | 64 |



|                             |            |
|-----------------------------|------------|
| <b>第3章 Java 面向对象程序设计（上）</b> | <b>65</b>  |
| 3.1 面向对象求解概述                | 65         |
| 3.1.1 利用计算机解决问题的方式          | 65         |
| 3.1.2 面向过程的程序设计             | 66         |
| 3.1.3 面向对象的方法               | 66         |
| 3.1.4 抽象与封装                 | 68         |
| 3.1.5 面向对象程序设计语言的特点         | 70         |
| 3.2 类与对象                    | 70         |
| 3.2.1 类的声明                  | 71         |
| 3.2.2 类属性的声明                | 74         |
| 3.2.3 类方法的声明                | 77         |
| 3.2.4 方法的重载                 | 86         |
| 3.2.5 构造方法                  | 88         |
| 3.2.6 类与对象应用实例              | 92         |
| 3.3 类的组合与嵌套                 | 96         |
| 3.3.1 类的组合                  | 96         |
| 3.3.2 类的嵌套                  | 98         |
| 3.4 继承                      | 99         |
| 3.4.1 继承的概念                 | 99         |
| 3.4.2 继承的实现                 | 100        |
| 3.4.3 属性的继承、扩展和隐藏           | 103        |
| 3.4.4 方法的继承、扩展和覆盖           | 106        |
| 3.4.5 类的继承与构造方法             | 108        |
| 3.4.6 组合与继承的比较              | 110        |
| 3.5 多态                      | 113        |
| 3.5.1 多态的概念                 | 113        |
| 3.5.2 对象引用的多态               | 113        |
| 3.5.3 多态中的方法覆盖与重载           | 114        |
| 3.5.4 动态绑定                  | 115        |
| 小结                          | 118        |
| 习题                          | 118        |
| <b>第4章 Java 面向对象程序设计（下）</b> | <b>119</b> |
| 4.1 静态变量和静态方法               | 119        |
| 4.1.1 静态变量                  | 119        |
| 4.1.2 静态方法                  | 123        |
| 4.2 抽象类和抽象方法                | 126        |
| 4.2.1 抽象类                   | 126        |
| 4.2.2 抽象方法                  | 127        |



|              |                                |            |
|--------------|--------------------------------|------------|
| 4.3          | 最终类和最终方法 .....                 | 129        |
| 4.3.1        | 最终类 .....                      | 129        |
| 4.3.2        | 最终方法 .....                     | 130        |
| 4.4          | 包 .....                        | 132        |
| 4.4.1        | 包和名字空间 .....                   | 132        |
| 4.4.2        | 包的创建与包的应用 .....                | 133        |
| 4.5          | 修饰符和访问控制 .....                 | 136        |
| 4.5.1        | 类与成员的修饰符 .....                 | 137        |
| 4.5.2        | 类的访问控制 .....                   | 138        |
| 4.6          | 接口 .....                       | 143        |
| 4.6.1        | 接口的定义 .....                    | 144        |
| 4.6.2        | 接口的实现 .....                    | 146        |
| 4.6.3        | 接口的使用 .....                    | 149        |
| 小结           | .....                          | 151        |
| 习题           | .....                          | 152        |
| <b>第 5 章</b> | <b>Java 语言基础类库 .....</b>       | <b>153</b> |
| 5.1          | Java 类库 .....                  | 153        |
| 5.1.1        | Java 类库概述 .....                | 153        |
| 5.1.2        | Java 类库结构 .....                | 154        |
| 5.2          | Java 语言包 ( java.lang ) .....   | 155        |
| 5.2.1        | Object 类 .....                 | 155        |
| 5.2.2        | Math 类 .....                   | 159        |
| 5.2.3        | System 类 .....                 | 160        |
| 5.3          | Java 实用程序包 ( java.util ) ..... | 163        |
| 5.3.1        | 日期类 .....                      | 163        |
| 5.3.2        | 随机数类 ( Random ) .....          | 166        |
| 5.3.3        | 数组类 ( Arrays ) .....           | 167        |
| 5.3.4        | 向量类 ( Vector ) .....           | 170        |
| 5.4          | Java 文本包 ( java.text ) .....   | 177        |
| 5.4.1        | 日期的格式化 .....                   | 177        |
| 5.4.2        | 数字的格式化 .....                   | 179        |
| 5.5          | Java 数学包 ( java.math ) .....   | 180        |
| 小结           | .....                          | 182        |
| 习题           | .....                          | 183        |
| <b>第 6 章</b> | <b>Java 程序中的异常处理 .....</b>     | <b>184</b> |
| 6.1          | 程序异常概念 .....                   | 184        |
| 6.2          | 异常与异常类 .....                   | 186        |
| 6.2.1        | 系统异常类 .....                    | 186        |



|                                          |            |
|------------------------------------------|------------|
| 6.2.2 用户自定义异常类 .....                     | 187        |
| 6.3 异常处理 .....                           | 189        |
| 6.3.1 抛出异常 .....                         | 189        |
| 6.3.2 异常的捕获与处理 .....                     | 194        |
| 6.3.3 多异常处理 .....                        | 197        |
| 6.4 异常处理原则 .....                         | 202        |
| 小结 .....                                 | 205        |
| 习题 .....                                 | 205        |
| <b>第 7 章 Java 程序的输入/输出 .....</b>         | <b>206</b> |
| 7.1 Java 流式输入/输出概述 .....                 | 206        |
| 7.1.1 I/O 流的层次 .....                     | 207        |
| 7.1.2 I/O 流的分类 .....                     | 209        |
| 7.2 字节流类 .....                           | 211        |
| 7.2.1 基本字节流类 .....                       | 211        |
| 7.2.2 用字节流读/写文件 .....                    | 214        |
| 7.2.3 用字节流读/写二进制数据 .....                 | 217        |
| 7.3 字符流类 .....                           | 221        |
| 7.3.1 基本字符流类 .....                       | 222        |
| 7.3.2 用字符流读/写文件 .....                    | 226        |
| 7.3.3 用数据包裹类读取基本类型数据 .....               | 228        |
| 7.4 缓冲流类 .....                           | 229        |
| 7.4.1 字节流缓冲流 .....                       | 230        |
| 7.4.2 字符流缓冲流 .....                       | 231        |
| 7.5 对象流类 .....                           | 233        |
| 7.6 文件操作 .....                           | 236        |
| 7.6.1 文件与目录操作 (File 类) .....             | 236        |
| 7.6.2 文件的随机访问 (RandomAccessFile 类) ..... | 239        |
| 小结 .....                                 | 241        |
| 习题 .....                                 | 241        |
| <b>第 8 章 Java 数据库编程接口 .....</b>          | <b>242</b> |
| 8.1 数据库基础知识 .....                        | 242        |
| 8.1.1 数据库技术概述 .....                      | 242        |
| 8.1.2 数据模型和关系表 .....                     | 243        |
| 8.2 建立 Access 数据库实例 .....                | 245        |
| 8.2.1 在 Access 中创建表 .....                | 246        |
| 8.2.2 创建 ODBC 数据源 .....                  | 248        |
| 8.3 SQL 语言入门 .....                       | 250        |
| 8.3.1 SQL 语言简介 .....                     | 250        |

参考文献 ..... 308

# 第 1 章

## Java 语言概述

Java 是 Sun Microsystems 公司研制的一种网络程序设计语言。在高级语言已经非常丰富的背景下, Java 语言脱颖而出, 不仅成为一门流行的计算机语言, 而且形成了一种专门的技术, 拥有独特的历史背景和独树一帜的品质。

### 学习目标

在学习完本章之后, 读者应该能够:

- 了解 Java 语言的几个主要特点;
- 能够独立安装 JDK 开发工具及完成环境设置;
- 编写并运行一个 Java Application 程序, 理解相关的 Java 文件。

### 1.1 Java 语言的形成与发展

Java 语言是一种高级网络编程语言, 由 Sun 公司的 Green 小组开发。随着第一个支持 Java 的浏览器 Mosaic 的诞生, Java 进入了网络应用。1995 年 5 月 23 日, JDK (Java Development Kit, Java 开发工具) 1.02a 版本正式发布, Java 语言正式诞生。

“Java”这个名字既不是根据语言本身的特色来命名的, 也不是由几个英文单词的首字母拼成, 更不是由名人或典故而来, 而是印度尼西亚的一个生产咖啡的岛屿的名字, 岛屿的中文名叫“爪哇”。正是因为许多程序设计师从所钟爱的热腾腾的香浓咖啡中得到了灵感, 所以热腾腾的香浓咖啡也就成了 Java 语言的标志 (见图 1-1)。



图 1-1 Sun 公司 Java 语言的 Logo

Java 语言是当今流行的网络编程语言。它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的计算概念。它使互联网从最初的单纯发布静态信息, 到现在的提供各种各样的动态服务。Java 强大的网络功能能够把整个 Internet 作为一个统一的运行平台。



Java 不仅是一门高级网络编程语言，还是一个优秀的开发平台。Java 平台代表了当今开源技术的发展方向，具有强大的平台扩展能力。Java 不断扩展的平台，包括以企业计算为主的 Java EE (Java Platform Enterprise Edition)，以桌面计算为主的 Java SE (Java Platform Standard Edition) 和以嵌入式计算为主的 Java ME (Java Platform Micro Edition)。Java 平台正显示着其广阔的应用前景。

Java 已经形成一门技术，主要体现在：

- Java 软件技术：Java JDK、Java Runtime Environment、HotJava、Java OS、JDBC、Java Beans 等。
- Java 虚拟机及 JRE 规范等。
- Java 嵌入技术：Java 芯片（如 MicroJava701），基于 Java 技术的 NC、Java Station、WebTV（机顶盒）、Java 汽车、Java 手机、Java 掌上机等。
- Java Computing：基于处理功能的综合处理系统，如 Java EE Server 或 Web Application Server，其典型产品有 Weblogic、Websphere 等。
- Java+XML：电子商务解决方案。

## 1.2 Java 语言特点

Sun 公司对 Java 的定义是：“Java: A simple, object-oriented, distributed, robust, secure, architecture-neutral, portable, high-performance, multi-threaded and dynamic language.” 即 Java 是一种简单的、面向对象的、分布式的、健壮的、安全的、体系结构中立的、可移植的、高性能的、多线程的动态语言。这个定义概括地说明了 Java 语言的特点。

下面简单介绍一下 java 语言的几个主要特点：

### 1. 简单性

Java 的基本语法部分与 C 语言几乎完全一致。Java 语言出于安全性的考虑，去除了 C/C++ 中不容易理解和掌握的部分，如指针操作、多重继承等，降低了学习难度。

### 2. 面向对象

有别于传统语言，Java 是完全面向对象的语言。Java 语言提供了类的机制，在对象中封装了成员变量和方法，实现了数据的封装和信息的隐藏；类提供了一类对象的模型，通过继承和多态，实现了代码的复用。

### 3. 分布性

Java 是一个分布式语言。它有一套很齐全的通信及相关功能的程序库，可以处理 TCP/IP 协议和其他协议，用户可以用 URL 地址在网络上方便地访问其他对象。利用 Java 来开发分布式网络程序是 Java 的主要应用之一。

### 4. 健壮性

内存管理是 C 和 C++ 程序中最容易产生错误的地方，如果内存分配与内存释放不符，就可能消耗系统资源直至将其耗尽，最后造成程序异常终止。Java 中所有的对象都是通过动态内存分配建立的，Java 对内存自动进行管理并进行垃圾回收，防止了程序员失误而导致的内存分配错误，进而更好地利用系统资源。

### 5. 安全性

安全性是网络环境下需要面对的最重要的问题。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实现，防止了以不法手段访问对象的私有成员，同时避免了指针操作中容易引起的错误。Java 的内部安全机制保证 Java 程序在 Java 虚拟机（Java Virtual Machine，简称 JVM）规则下操作，防止未授权的程序访问含有专有信息的系统资源或危及客户机的完整性。

#### 6. 体系结构中立

Java 编译器能够产生一种与计算机体系结构无关的字节码（bytecodes），只要安装了 Java 虚拟机，Java 源代码就可以在相应的处理器上执行。

#### 7. 可移植性

Java 编译器生成与平台无关的字节码指令，这些字节码对应于 Java 虚拟机中的表示。Java 解释器得到字节码后，对其进行解释，使之能够在不同的平台上运行。不同的操作系统有不同的虚拟机。Java 编译器主要由 Java 语言本身实现，Java 的运行系统由标准 C 实现，因而整个 Java 系统都具有可移植性。

#### 8. 高性能

Java 的字节码能够迅速地转换成机器码，充分利用硬件平台资源，从而可以得到较高的整体性能。

#### 9. 多线程机制

多线程是一种程序并行运行的机制。Java 是第一个在编程语言中提供内置多线程支持的高级语言，它的同步机制保证了对共享数据的共享操作。这大大简化了多线程程序的编写，而一些其他语言要通过调用操作系统的原语来支持多线程。

#### 10. 动态性

Java 比 C++语言有更强大的动态性，更能适应不断变化的环境。Java 不会因程序库某些部分的更新而需要全部重新编译程序。所以，在 Java 类库中可以自由地加入新的内容而不会影响用户程序的执行。Java 通过接口（interface）机制支持多重继承，使之比严格的类继承更有灵活性和扩展性。

## 1.3 Java 程序工作方式

程序的运行需要一定的硬件和软件环境，这个环境称为平台（platform），平台可以看做操作系统和硬件的集合。但 Java 平台不同，它是运行于计算机硬件系统和操作系统之上的软平台。

Java 平台包括 Java 的应用程序接口（Application Programming Interface，API）和 Java 虚拟机。Java 应用程序接口是经过编译的，可以在程序中直接使用的 Java 代码标准库，是程序员和 Java 系统交互的接口。Java 虚拟机负责解释和执行 Java 程序。

Java 平台如图 1-2 所示。

Java 程序上机过程包括编译过程和执行过程（见图 1-3）。

① 编译过程：由编译器（compiler）把 Java 源文件（文件扩展名为 .java）编译成字节码文件（文件扩展名为 .class）。

② 执行过程：由解释器（interpreter）负责解释和执行字节码文件。

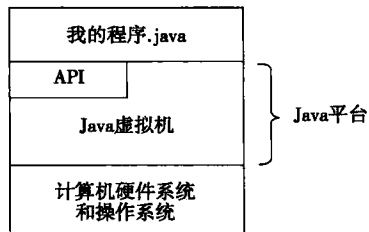


图 1-2 Java 平台示意图

字节码是一组类似机器码的指令，但它独立于具体的处理器（如 Intel Pentium 处理器，Mac Power 处理器等）。因此，相同的 Java 源文件经过不同的计算机系统（如 Windows, Linux, MacOS）编译后，将生成相同的字节码文件。可见字节码具有与平台无关的特性，用户可以容易地将 Java 字节码在不同的计算机系统之间移植。

显然，对不同的计算机系统，Java 编译器和解释器的工作方式是不一样的。因此，不同的计算机系统，需要安装相应的 Java 编译器和解释器。

Java 程序运行于 Java 平台之上，经过编译和解释的程序可以通过 Java 平台在各种操作系统中运行，具有优越的跨平台性能，如图 1-4 所示。

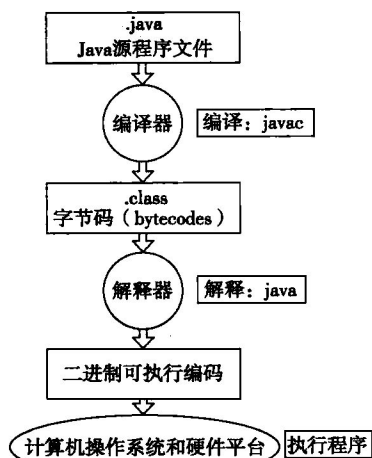


图 1-3 Java 语言的工作方式

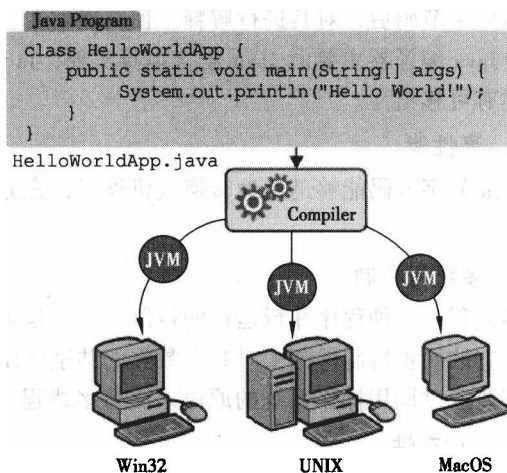


图 1-4 Java 程序在几种典型平台之间移植

## 1.4 JDK 实用程序

JDK (Java Developing Kit, Java 开发包) 是 Sun 公司提供的免费的 Java 开发工具和运行环境。下面介绍 JDK 的下载、安装和使用。

### 1. JDK 的下载

通过浏览器上网后，可以在 Sun 公司的站点找到 JDK 的页面，链接网址是：<http://java.sun.com/javase/downloads/index.jsp>。本书使用的 JDK 版本是 JDK 1.6。

Sun 公司提供对 Windows 操作系统的支持，也提供支持类似 Solaris SPARC 和 Macintosh 操作系统的版本。用户可以根据操作系统平台选取合适的 JDK 版本下载。

### 2. JDK 的安装与环境设置

下面是在 Windows XP 操作系统中，安装 JDK 和设置环境变量。其他操作系统平台的安装与此类似。

在 Windows 环境下，双击下载的 JDK 压缩文件，即可自动进入安装过程。此时可按照安装向导，逐步完成安装。安装时需要选择一个目录，为方便起见，保持默认的路径即可。默认路径一般为“C:\Program Files\Java\jdk1.6.0”，这个目录较复杂，会给 DOS 命令带来不便，这个问题可以在环境变量的设置中得到解决。



JDK 安装后，会在安装目录下生成若干文件夹，主要的文件介绍如下：

- bin 目录：存放一些可执行文件，Java 的编译工具 `javac` 和 Java 解释器 `java` 工具就在该目录下。
- lib 目录：存放 Java 开发类库。
- demo 目录：存放一些 Applet（Java 小程序）的实例。
- sample 目录：存放一些 Java 程序演示。这是程序初学者学习一门语言的快速途径。
- jre 目录：存放 Java 的运行环境，包括 Java 虚拟机和运行类库等。

安装好 JDK 后，需要设置系统环境变量。在 Windows NT/2000/XP 操作系统中，设置环境变量的方法是：

① 打开系统的【控制面板】，双击【系统】图标，在弹出的对话框中单击【高级】选项卡，单击【环境变量】按钮，在此可以设置环境变量。

② 在【系统变量】区中单击【新建】按钮，在【新建系统变量】对话框的【变量名】文本框中输入“`classpath`”，在【变量值】文本框中输入“`C:\Program Files\Java\jdk1.6.0_07\bin; C:\Program Files\Java\jdk1.6.0_07\lib\dt.jar; C:\Program Files\Java\jdk1.6.0_07\lib\tools.jar`”，最后单击【确定】按钮。注意：这里假定 JDK 的安装目录为“`C:\Program Files\Java\jdk1.6.0_07`”，若 JDK 为其他安装目录，则应在【变量值】文本框中输入相应的路径。

在 Windows XP 系统中，对 Java 编译环境的环境变量设置如图 1-5 所示。



图 1-5 设置环境变量

### 3. JDK 实用工具简介

这里将简单介绍 JDK 的环境工具，包括 Java 编译器、Java 解释器、Java 虚拟机、Applet 程序观察器、Java 文档生成器、Java 调试器等。

#### (1) `javac`——Java 程序的编译器

`javac` 的作用是将源程序（.java 文件）编译成字节码文件（.class 文件）。Java 源程序的扩展名必须是“`.java`”。`javac` 一次可以编译一个或多个源程序，对于源程序中定义的每个类，都会生成一个单独的字节码文件。因此，Java 源文件与生成的 .class 文件之间不存在一一对应的关系。

`javac` 的调用格式为：

`javac` [选项] 文件名列表



其中，“文件名列表”是一个或多个带有“.java”扩展名的文件名，也可以采用“\*.java”对当前目录下的所有 java 源文件进行编译。

“选项”可以在 JDK 的帮助中查到。输入“javac”，系统就会列出 javac 的选项及说明。

### (2) java——Java 程序的解释器

java 命令解释执行 Java 字节码，其调用格式为：

```
java [选项] 字节码文件名<参数表>
```

这里的“字节码文件名”就是编译后生成的带“.class”扩展名的类文件名，但在 java 命令中的“字节码文件名”不用带“.class”扩展名。这个类文件名必须是个独立的程序，不能是 Applet 小程序。

java 命令可以运行 Java Application 程序，它启动 java 运行环境，加载指定的类，并调用这个类的 main 方法。

### (3) jdb——Java 语言调试工具

jdb 可以调试 Java 程序。其格式为

```
jdb [选项] 类名
```

或：

```
jdb [-host 主机名] -password 密码(或口令)
```

jdb 装载指定的类，启动内嵌的 Java 虚拟机，然后等待用户发出相应的调试命令。通过调用 Java Debugger API 能够对本机或远程的 Java 虚拟机进行调试。

第一种命令格式，由 jdb 解释执行被调试的类。第二种格式，jdb 将被嵌入到一个正在运行的 Java 虚拟机中，这个虚拟机事先须运行在 debug 模式下。

### (4) javadoc——Java 文档生成器

javadoc 从 Java 源文件生成 HTML 格式的 API 文档，内容包括类和接口的描述、类的继承层次以及类中任何非私有域的索引和介绍。

javadoc 的调用格式为：

```
javadoc [选项] [包|文件名]
```

### (5) appletviewer——Java Applet 观察器

appletviewer 命令可以使 Applet 小程序不通过 Web 浏览器也可以执行。其格式为：

```
appletviewer [-debug] HTML 文件
```

appletviewer 下载并运行 HTML 文件中包含的 Applet，如果 HTML 文件中不包含任何 Applet，appletviewer 则不执行任何行为。上述命令使用了 -debug 选项，则 appletviewer 将从 jdb 内部启动，以调试模式运行 HTML 文件引用的 Applet。

## 4. Java 帮助文档

Java 帮助文档是程序员开发 Java 程序的重要参考资料。Java 帮助文档可以在 Sun 公司的官方网站上下载，地址是 <http://java.sun.com/javase/downloads/index.jsp>。

Java 帮助文档由 Java 类库的使用说明及简单例程两部分组成。Java 语言本身有一套约定俗成的编程规范，这些编程规范在 Java 帮助文档里都有体现。Java 初学者应学会阅读帮助文档，并养成遇到问题查阅帮助文档的习惯。Java 帮助文档中的 Java Platform API Specification 是 Java 类库帮助信息的入口，程序员可以从中找到 Java 各种类、接口的说明，其中包括属性、构造函数、方法的说明等。