

普.通.高.等.学.校  
计算机教育“十二五”规划教材

# Java 程序设计 实用教程

(第2版)

THE JAVA PROGRAMMING LANGUAGE  
(2<sup>nd</sup> edition)

耿祥义 张跃平 ◆ 编著

 人民邮电出版社  
POSTS & TELECOM PRESS

普.通.高.等.学.校  
计算机教育“十二五”规划教材

# Java 程序设计 实用教程

(第2版)

---

THE JAVA PROGRAMMING LANGUAGE  
(2<sup>nd</sup> edition)

耿祥义 张跃平 ◆ 编著

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

Java程序设计实用教程 / 耿祥义, 张跃平编著. --  
2版. -- 北京: 人民邮电出版社, 2015. 4  
普通高等学校计算机教育“十二五”规划教材  
ISBN 978-7-115-38417-1

I. ①J… II. ①耿… ②张… 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(2015)第040088号

## 内 容 提 要

Java 语言具有面向对象、与平台无关、安全、稳定、多线程等优良特性,是目前软件设计中极为强大的编程语言。本书注重结合实例,每章分别配有相应的上机实训,循序渐进地向读者介绍了 Java 语言的重要知识点,特别强调 Java 面向对象编程的思想。全书分为 15 章,分别讲解了简单数据类型、运算符、表达式和语句、类与对象、子类与继承、接口与多态、数组与枚举、内部类与异常类、常用实用类、Java 输入输出流、JDBC 数据库操作、泛型与集合框架、Java 多线程机制、Java 网络基础、图形用户界面设计等内容。

本书适合作为高等院校计算机相关专业“Java 语言程序设计”以及“面向对象语言”课程的教材。

- 
- ◆ 编 著 耿祥义 张跃平  
责任编辑 刘 博  
责任印制 沈 蓉 彭志环
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京昌平百善印刷厂印刷
  - ◆ 开本: 787×1092 1/16  
印张: 24.75 2015 年 4 月第 2 版  
字数: 650 千字 2015 年 4 月北京第 1 次印刷
- 

定价: 54.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316  
反盗版热线: (010)81055315

## 第 2 版前言

---

---

---

---

---

---

---

---

Java 是一种纯面向对象的程序设计语言，具有跨平台、分布性、高性能、可移植等优点，是目前被广泛使用的编程语言之一，很多新的计算机技术领域也都涉及 Java 语言。

本书第 2 版对某些章节的内容作了适度的修改，并在每一章增加了相应的上机实训，对第 11 章做了全面改动，将原有的 JDBC 操作 Access 数据库更改为 JDBC 操作 Derby 数据库，删除了已经不再使用的第 16 章的关于 Java Applet 内容。

全书在内容和语言组织上注重 Java 语言的面向对象特性，强调面向对象的程序设计思想，在实例上注重实用性和启发性，在内容的深度和广度方面都进行了深入考虑，在类、对象、继承、接口等重要的基础知识上侧重深度，而在实用类、输入输出流、Java 网络技术、JDBC 数据库操作等实用技术方面的讲解上侧重广度。通过本书的学习，读者可以掌握 Java 面向对象编程的思想和 Java 编程中的一些重要技术。

本书语言通俗易懂，例子生动实用，配备的实训内容不仅有利于知识的掌握和运用，而且对提高编程能力也非常有帮助。每一章的后面还提供了习题，方便老师和同学及时检验学习效果。

全书共分 15 章。第 1 章主要介绍了 Java 产生的背景和 Java 平台，读者可以了解到 Java 是怎样做到“一次写成，处处运行”的。第 2 章通过学习一个简单的对象，初步了解对象的结构，并讲解了简单的数据类型。第 3 章主要介绍了 Java 运算符和控制语句。第 4 章、第 5 章和第 6 章是本书的重点内容，讲述了类与对象、子类与继承、接口与多态等内容。第 7 章和第 8 章是对第 4 章、第 5 章知识的总结升华。第 7 章讲解了数组与枚举，特别讲解了与数组相关的一些实用技术。第 8 章讲解了内部类和匿名类，特别强调了使用内部类的原则以及学习自定义异常的重要性。第 9 章讲解了常用的实用类，包括字符串、日期、正则表达式、模式匹配、数学计算等实用类，特别讲解了如何使用 Scanner 类解析字符串。第 10 章讲解了 Java 中的输入/输出流技术，这部分特别介绍了如何使用输入/输出流来克隆对象，Java 的文件锁技术以及使用 Scanner 解析文件等重要内容。第 11 章主要讲解 Java 如何使用 JDBC 操作 Derby 数据库，讲解了预处理、事务处理、批处理等重要技术。第 12 章讲解泛型和集合框架，强调如何使用集合框架提供的类来有效、合理地组织程序中的数据。第 13 章讲解了多线程技术，通过许多有启发的例子来帮助读者理解多线程编程。第 14 章讲解 Java 在网络编程中的一些重要技术，涉及 URL、Socket、InetAddress、DatagramPacket 等重要的类，而且特别讲解了 Java 远程调用 (RMI)。第 15 章是基于 Java Swing 的 GUI 图形用户界面设计，讲解了常用的组件和容器，特别详细讲解了事件处理。

在学习本书之前，读者最好具有 C 语言基础。掌握一门语言最好的方式就是实践，本书的着眼点是将基础的理论知识讲解和实践应用相结合，使读者在理解面向对象思想的基础上，快速掌握 Java 编程技术。

本书实例的源程序以及电子教案可以在人民邮电出版社教学服务与资源网 ([www.ptpedu.com.cn](http://www.ptpedu.com.cn)) 上免费下载，以供读者学习使用。

编者

2014 年 12 月

---

---

# 目 录

|                                 |                                  |
|---------------------------------|----------------------------------|
| <b>第 1 章 初识 Java</b> ..... 1    | 2.7 标识符与关键字..... 20              |
| 1.1 Java 诞生的原因..... 1           | 2.7.1 标识符..... 20                |
| 1.2 Java 的地位..... 3             | 2.7.2 关键字..... 21                |
| 1.2.1 网络地位..... 3               | 2.8 简单数据类型..... 21               |
| 1.2.2 语言地位..... 4               | 2.8.1 逻辑类型..... 21               |
| 1.2.3 需求地位..... 4               | 2.8.2 整数类型..... 22               |
| 1.3 安装 JDK..... 4               | 2.8.3 字符类型..... 22               |
| 1.3.1 三种平台简介..... 4             | 2.8.4 浮点类型..... 23               |
| 1.3.2 安装 Java SE 平台..... 5      | 2.9 简单数据类型的级别与数据转换..... 25       |
| 1.3.3 设置环境变量..... 6             | 2.10 从命令行窗口输入、输出数据..... 26       |
| 1.4 Java 程序的开发步骤..... 7         | 2.10.1 输入基本型数据..... 26           |
| 1.5 一个简单的 Java 应用程序..... 8      | 2.10.2 输出基本型数据..... 26           |
| 1.5.1 编写源文件..... 8              | 2.11 编程风格..... 27                |
| 1.5.2 编译..... 9                 | 2.11.1 Allmans 风格..... 28        |
| 1.5.3 运行..... 9                 | 2.11.2 Kernighan 风格..... 28      |
| 1.6 Java 的语言特点..... 10          | 2.11.3 注释..... 28                |
| 1.6.1 简单..... 10                | 2.12 上机实践..... 29                |
| 1.6.2 面向对象..... 10              | 2.12.1 实验 1 联合编译..... 29         |
| 1.6.3 多线程..... 10               | 2.12.2 实验 2 输出希腊字母表..... 30      |
| 1.6.4 安全..... 11                | 2.12.3 实验 3 从键盘输入数据..... 31      |
| 1.6.5 动态..... 11                | 习题 2..... 32                     |
| 1.7 上机实践..... 11                | <b>第 3 章 运算符、表达式和语句</b> ..... 34 |
| 习题 1..... 12                    | 3.1 运算符与表达式..... 34              |
| <b>第 2 章 初识对象和简单数据类型</b> ... 14 | 3.1.1 算术运算符与算术表达式..... 34        |
| 2.1 问题的提出..... 14               | 3.1.2 自增, 自减运算符..... 34          |
| 2.2 简单的矩形类..... 15              | 3.1.3 算术混合运算的精度..... 35          |
| 2.3 使用矩形类创建对象..... 16           | 3.1.4 关系运算符与关系表达式..... 35        |
| 2.3.1 用类声明对象..... 16            | 3.1.5 逻辑运算符与逻辑表达式..... 35        |
| 2.3.2 为对象分配变量..... 16           | 3.1.6 赋值运算符与赋值表达式..... 36        |
| 2.3.3 使用对象..... 17              | 3.1.7 位运算符..... 36               |
| 2.4 在 Java 应用程序中使用矩形对象..... 17  | 3.1.8 instanceof 运算符..... 37     |
| 2.5 Java 应用程序的基本结构..... 18      | 3.1.9 运算符综述..... 37              |
| 2.6 一个源文件中编写多个类..... 19         | 3.2 语句概述..... 37                 |
|                                 | 3.3 if 条件分支语句..... 38            |

|                          |           |                                  |           |
|--------------------------|-----------|----------------------------------|-----------|
| 3.3.1 if 语句              | 38        | 4.7.3 实例方法和类方法的定义                | 71        |
| 3.3.2 if-else 语句         | 38        | 4.7.4 实例方法和类方法的区别                | 72        |
| 3.3.3 if-else if-else 语句 | 39        | 4.8 方法重载与多态                      | 73        |
| 3.4 switch 开关语句          | 41        | 4.9 this 关键字                     | 74        |
| 3.5 循环语句                 | 43        | 4.10 包                           | 75        |
| 3.5.1 for 循环语句           | 43        | 4.10.1 包语句                       | 75        |
| 3.5.2 while 循环           | 44        | 4.10.2 有包名的类的存储目录                | 75        |
| 3.5.3 do-while 循环        | 44        | 4.10.3 运行有包名的主类                  | 76        |
| 3.6 break 和 continue 语句  | 45        | 4.11 import 语句                   | 77        |
| 3.7 上机实践                 | 46        | 4.11.1 引入类库中的类                   | 77        |
| 3.7.1 实验 1 计算电费          | 46        | 4.11.2 引入自定义包中的类                 | 78        |
| 3.7.2 实验 2 猜数字游戏         | 47        | 4.11.3 使用无包名的类                   | 80        |
| 习题 3                     | 48        | 4.11.4 避免类名混淆                    | 81        |
| <b>第 4 章 类与对象</b>        | <b>50</b> | 4.12 访问权限                        | 81        |
| 4.1 从抽象到类                | 50        | 4.12.1 何谓访问权限                    | 81        |
| 4.2 类                    | 51        | 4.12.2 私有变量和私有方法                 | 82        |
| 4.2.1 类声明                | 51        | 4.12.3 共有变量和共有方法                 | 83        |
| 4.2.2 类体                 | 51        | 4.12.4 友好变量和友好方法                 | 83        |
| 4.2.3 成员变量               | 52        | 4.12.5 受保护的成员变量和方法               | 84        |
| 4.2.4 方法                 | 53        | 4.12.6 public 类与友好类              | 84        |
| 4.2.5 需要注意的问题            | 54        | 4.13 基本类型的类包装                    | 85        |
| 4.2.6 类的 UML 类图          | 55        | 4.13.1 Double 和 Float 类          | 85        |
| 4.3 构造方法与对象的创建           | 55        | 4.13.2 Byte、Short、Integer、Long 类 | 85        |
| 4.3.1 构造方法               | 55        | 4.13.3 Character 类               | 85        |
| 4.3.2 创建对象               | 56        | 4.14 反编译和文档生成器                   | 86        |
| 4.3.3 使用对象               | 58        | 4.14.1 javap 反编译                 | 86        |
| 4.3.4 对象的引用和实体           | 59        | 4.14.2 javadoc 制作文档              | 86        |
| 4.4 参数传值                 | 60        | 4.15 上机实践                        | 86        |
| 4.4.1 传值机制               | 61        | 4.15.1 实验 1 用类描述坦克               | 86        |
| 4.4.2 基本数据类型参数的传值        | 61        | 4.15.2 实验 2 学校与教师                | 88        |
| 4.4.3 引用类型参数的传值          | 62        | 4.15.3 实验 3 共同的森林                | 90        |
| 4.4.4 可变参数               | 63        | 习题 4                             | 91        |
| 4.5 有理数的类封装              | 65        | <b>第 5 章 子类与继承</b>               | <b>94</b> |
| 4.6 对象的组合                | 67        | 5.1 子类与父类                        | 94        |
| 4.6.1 圆锥体                | 68        | 5.2 子类的继承性                       | 95        |
| 4.6.2 关联关系和依赖关系的 UML 图   | 69        | 5.2.1 子类和父类在同一包中的继承性             | 95        |
| 4.7 实例成员与类成员             | 69        | 5.2.2 子类和父类不在同一包中的继承性            | 97        |
| 4.7.1 实例变量和类变量的声明        | 69        | 5.2.3 protected 的进一步说明           | 97        |
| 4.7.2 实例变量和类变量的区别        | 70        |                                  |           |

|  |     |  |     |
|--|-----|--|-----|
| 5.2.4 继承关系 (Generalization) 的<br>UML 图 ..... | 97  | <b>第 7 章 数组与枚举</b> .....               | 140 |
| 5.2.5 关于 instanceof 运算符 .....                | 98  | 7.1 创建数组 .....                         | 140 |
| 5.3 子类对象的特点 .....                            | 98  | 7.1.1 声明数组 .....                       | 140 |
| 5.4 成员变量的隐藏和方法重写 .....                       | 99  | 7.1.2 为数组分配元素 .....                    | 141 |
| 5.4.1 成员变量的隐藏 .....                          | 99  | 7.1.3 数组元素的使用 .....                    | 142 |
| 5.4.2 方法重写 (Override) .....                  | 100 | 7.1.4 length 的使用 .....                 | 142 |
| 5.5 super 关键字 .....                          | 105 | 7.1.5 数组的初始化 .....                     | 142 |
| 5.5.1 用 super 操作被隐藏的成员<br>变量和方法 .....        | 105 | 7.1.6 数组的引用 .....                      | 143 |
| 5.5.2 使用 super 调用父类的构造方法 .....               | 106 | 7.2 遍历数组 .....                         | 144 |
| 5.6 final 关键字 .....                          | 107 | 7.2.1 基于循环语句的遍历 .....                  | 144 |
| 5.6.1 final 类 .....                          | 107 | 7.2.2 使用 toString() 方法遍历数组 .....       | 145 |
| 5.6.2 final 方法 .....                         | 108 | 7.3 复制数组 .....                         | 145 |
| 5.6.3 常量 .....                               | 108 | 7.3.1 arraycopy 方法 .....               | 146 |
| 5.7 对象的上转型对象 .....                           | 108 | 7.3.2 copyOf 和 copyOfRange() 方法 .....  | 146 |
| 5.8 继承与多态 .....                              | 110 | 7.4 排序与二分查找 .....                      | 148 |
| 5.9 abstract 类和 abstract 方法 .....            | 111 | 7.5 枚举 .....                           | 149 |
| 5.10 面向抽象编程 .....                            | 112 | 7.5.1 枚举类型的定义 .....                    | 149 |
| 5.11 开-闭原则 .....                             | 115 | 7.5.2 枚举变量 .....                       | 149 |
| 5.12 上机实践 .....                              | 118 | 7.5.3 枚举类型与 for 语句和 switch<br>语句 ..... | 150 |
| 5.12.1 实验 1 猫与狗 .....                        | 118 | 7.6 上机实践 .....                         | 151 |
| 5.15.2 实验 2 图形的面积和 .....                     | 119 | 7.6.1 实验 1 遍历与复制数组 .....               | 151 |
| 习题 5 .....                                   | 121 | 7.6.2 实验 2 公司与薪水 .....                 | 152 |
| <b>第 6 章 接口与多态</b> .....                     | 124 | 习题 7 .....                             | 153 |
| 6.1 接口 .....                                 | 124 | <b>第 8 章 内部类与异常类</b> .....             | 155 |
| 6.1.1 接口的声明与使用 .....                         | 124 | 8.1 内部类 .....                          | 155 |
| 6.1.2 理解接口 .....                             | 127 | 8.2 匿名类 .....                          | 156 |
| 6.1.3 接口的 UML 图 .....                        | 128 | 8.2.1 和子类有关的匿名类 .....                  | 156 |
| 6.2 接口回调 .....                               | 129 | 8.2.2 和接口有关的匿名类 .....                  | 158 |
| 6.2.1 接口变量与回调机制 .....                        | 129 | 8.3 异常类 .....                          | 159 |
| 6.2.2 接口的多态性 .....                           | 130 | 8.3.1 try-catch 语句 .....               | 159 |
| 6.2.3 abstract 类与接口的比较 .....                 | 131 | 8.3.2 自定义异常类 .....                     | 161 |
| 6.3 面向接口编程 .....                             | 131 | 8.4 断言 .....                           | 163 |
| 6.4 上机实践 .....                               | 134 | 8.5 上机实践 .....                         | 164 |
| 6.4.1 实验 1 作战命令 .....                        | 134 | 8.5.1 实验 1 内部购物卷 .....                 | 164 |
| 6.4.2 实验 2 小狗的状态 .....                       | 136 | 8.5.2 实验 2 检查危险品 .....                 | 165 |
| 习题 6 .....                                   | 138 | 习题 8 .....                             | 167 |

**第 9 章 常用实用类**..... 169

|                           |     |
|---------------------------|-----|
| 9.1 String 类              | 169 |
| 9.1.1 构造字符串对象             | 169 |
| 9.1.2 String 类的常用方法       | 170 |
| 9.1.3 字符串与基本数据的相互转化       | 174 |
| 9.1.4 对象的字符串表示            | 175 |
| 9.1.5 字符串与字符、字节数组         | 176 |
| 9.1.6 正则表达式及字符串的替换<br>与分解 | 178 |
| 9.2 StringBuffer 类        | 182 |
| 9.2.1 StringBuffer 对象的创建  | 182 |
| 9.2.2 StringBuffer 类的常用方法 | 183 |
| 9.3 StringTokenizer 类     | 184 |
| 9.4 Scanner 类             | 185 |
| 9.5 Date 类                | 187 |
| 9.5.1 构造 Date 对象          | 187 |
| 9.5.2 日期格式化               | 188 |
| 9.6 Calendar 类            | 189 |
| 9.7 Math 和 BigInteger 类   | 192 |
| 9.7.1 Math 类              | 192 |
| 9.7.2 BigInteger 类        | 192 |
| 9.8 DecimalFormat 类       | 193 |
| 9.8.1 格式化数字               | 193 |
| 9.8.2 将格式化字符串转化为数字        | 195 |
| 9.9 Pattern 与 Match 类     | 196 |
| 9.9.1 模式对象                | 196 |
| 9.9.2 匹配对象                | 196 |
| 9.10 System 类             | 198 |
| 9.11 上机实践                 | 198 |
| 9.11.1 实验 1 检索简历          | 198 |
| 9.11.2 实验 2 购物小票          | 200 |
| 9.11.3 实验 3 成绩单           | 202 |
| 习题 9                      | 203 |

**第 10 章 输入、输出流**..... 206

|                 |     |
|-----------------|-----|
| 10.1 File 类     | 206 |
| 10.1.1 文件的属性    | 207 |
| 10.1.2 目录       | 208 |
| 10.1.3 文件的创建与删除 | 209 |

|   |     |
|---|-----|
| 10.1.4 运行可执行文件                          | 209 |
| 10.2 字节流与字符流                            | 210 |
| 10.2.1 InputStream 类与<br>OutputStream 类 | 210 |
| 10.2.2 Reader 类与 Writer 类               | 211 |
| 10.2.3 关闭流                              | 211 |
| 10.3 文件字节流                              | 211 |
| 10.3.1 文件字节输入流                          | 212 |
| 10.3.2 文件字节输出流                          | 213 |
| 10.4 文件字符流                              | 213 |
| 10.5 缓冲流                                | 214 |
| 10.6 随机流                                | 216 |
| 10.7 数组流                                | 219 |
| 10.8 数据流                                | 220 |
| 10.9 对象流                                | 223 |
| 10.10 序列化与对象克隆                          | 225 |
| 10.11 文件锁                               | 226 |
| 10.12 使用 Scanner 解析文件                   | 227 |
| 10.13 使用 Console 流读取密码                  | 229 |
| 10.14 上机实践                              | 230 |
| 10.14.1 实验 1 分析成绩单                      | 230 |
| 10.14.2 实验 2 统计英文单词                     | 232 |
| 习题 10                                   | 234 |

**第 11 章 JDBC 操作 Derby  
数据库**..... 236

|                               |     |
|-------------------------------|-----|
| 11.1 Derby 数据库                | 236 |
| 11.1.1 准备工作                   | 237 |
| 11.1.2 内置 Derby 数据库           | 237 |
| 11.1.3 网络 Derby 数据库           | 239 |
| 11.1.4 Derby 数据库常用的基本<br>数据类型 | 241 |
| 11.2 JDBC                     | 241 |
| 11.3 连接 Derby 数据库             | 242 |
| 11.3.1 连接内置 Derby 数据库         | 242 |
| 11.3.2 连接网络 Derby 数据库         | 243 |
| 11.4 查询操作                     | 245 |
| 11.4.1 顺序查询                   | 246 |
| 11.4.2 控制游标                   | 248 |
| 11.4.3 条件查询                   | 250 |



|                         |            |  |            |
|-------------------------|------------|--|------------|
| 11.4.4 排序查询             | 250        | 12.8 上机实践                                      | 284        |
| 11.4.5 模糊查询             | 250        | 12.8.1 实验 1 搭建流水线                              | 284        |
| 11.5 更新、添加与删除操作         | 251        | 12.8.2 实验 2 排序与查找                              | 286        |
| 11.6 使用预处理语句            | 252        | 习题 12  | 287        |
| 11.6.1 预处理语句优点          | 253        | <b>第 13 章 Java 多线程机制</b>                       | <b>289</b> |
| 11.6.2 使用通配符            | 254        | 13.1 进程与线程                                     | 289        |
| 11.7 事务                 | 256        | 13.1.1 操作系统与进程                                 | 289        |
| 11.7.1 事务及处理            | 256        | 13.1.2 进程与线程                                   | 290        |
| 11.7.2 JDBC 事务处理步骤      | 257        | 13.2 Java 中的线程                                 | 290        |
| 11.8 批处理                | 258        | 13.2.1 Java 的多线程机制                             | 290        |
| 11.9 CachedRowSetImpl 类 | 259        | 13.2.2 线程的状态与生命周期                              | 291        |
| 11.10 上机实践              | 261        | 13.2.3 线程调度与优先级                                | 294        |
| 11.10.1 实验 1 抽取职员工资     | 261        | 13.3 Thread 的子类创建线程                            | 294        |
| 11.10.2 实验 2 用户转账       | 264        | 13.4 使用 Runnable 接口                            | 296        |
| 习题 11                   | 266        | 13.4.1 Runnable 接口与目标对象                        | 296        |
| <b>第 12 章 泛型与集合框架</b>   | <b>267</b> | 13.4.2 关于 run 方法启动的次数                          | 297        |
| 12.1 泛型                 | 267        | 13.4.3 在线程中启动其他线程                              | 298        |
| 12.1.1 泛型类              | 267        | 13.5 线程的常用方法                                   | 299        |
| 12.1.2 泛型类声明对象          | 268        | 13.6 线程同步                                      | 302        |
| 12.1.3 泛型接口             | 269        | 13.6.1 什么是线程同步                                 | 302        |
| 12.1.4 泛型的目的            | 270        | 13.6.2 通过同步避免切换的影响                             | 304        |
| 12.2 链表                 | 270        | 13.7 在同步方法中使用 wait()、notify<br>和 notifyAll()方法 | 305        |
| 12.2.1 LinkedList<E>泛型类 | 270        | 13.8 线程联合                                      | 307        |
| 12.2.2 常用方法             | 270        | 13.9 上机实践                                      | 308        |
| 12.2.3 遍历链表             | 271        | 13.9.1 实验 1 键盘操作练习                             | 308        |
| 12.2.4 排序与查找            | 273        | 13.9.2 实验 2 多线程猜数字                             | 310        |
| 12.2.5 洗牌与旋转            | 275        | 习题 13  | 312        |
| 12.3 堆栈                 | 277        | <b>第 14 章 Java 网络编程</b>                        | <b>314</b> |
| 12.4 散列映射               | 278        | 14.1 URL 类                                     | 314        |
| 12.4.1 HashMap<K,V>泛型类  | 278        | 14.1.1 URL 的构造方法                               | 314        |
| 12.4.2 常用方法             | 278        | 14.1.2 读取 URL 中的资源                             | 315        |
| 12.4.3 遍历散列映射           | 279        | 14.2 InetAddress 类                             | 316        |
| 12.4.4 基于散列映射的查询        | 279        | 14.2.1 地址的表示                                   | 316        |
| 12.5 树集                 | 280        | 14.2.2 获取地址                                    | 316        |
| 12.5.1 TreeSet<E>泛型类    | 280        | 14.3 套接字                                       | 317        |
| 12.5.2 节点的大小关系          | 280        | 14.3.1 套接字                                     | 317        |
| 12.5.3 TreeSet 类的常用方法   | 281        | 14.3.2 客户端套接字                                  | 318        |
| 12.6 树映射                | 282        |  |            |
| 12.7 自动装箱与拆箱            | 283        |  |            |

|                        |                             |     |            |                  |     |
|------------------------|-----------------------------|-----|------------|------------------|-----|
| 14.3.3                 | ServerSocket 对象与服务<br>器端套接字 | 318 | 15.4.1     | 事件处理模式           | 349 |
| 14.3.4                 | 使用多线程技术                     | 321 | 15.4.2     | ActionEvent 事件   | 350 |
| 14.4                   | UDP 数据报                     | 324 | 15.4.3     | ItemEvent 事件     | 354 |
| 14.4.1                 | 发送数据包                       | 324 | 15.4.4     | DocumentEvent 事件 | 356 |
| 14.4.2                 | 接收数据包                       | 325 | 15.4.5     | MouseEvent 事件    | 358 |
| 14.5                   | 广播数据报                       | 328 | 15.4.6     | 焦点事件             | 362 |
| 14.6                   | Java 远程调用                   | 330 | 15.4.7     | 键盘事件             | 362 |
| 14.6.1                 | 远程对象及其代理                    | 330 | 15.4.8     | 匿名类实例或窗口做监视器     | 365 |
| 14.6.2                 | RMI 的设计细节                   | 331 | 15.4.9     | 事件总结             | 366 |
| 14.7                   | 上机实践                        | 334 | 15.5       | 使用 MVC 结构        | 367 |
|                        | 习题 14                       | 337 | 15.6       | 对话框              | 369 |
| <b>第 15 章 图形用户界面设计</b> |                             |     | <b>338</b> |                  |     |
| 15.1                   | Java Swing 概述               | 338 | 15.6.1     | 消息对话框            | 370 |
| 15.2                   | 窗口                          | 339 | 15.6.2     | 输入对话框            | 371 |
| 15.2.1                 | JFrame 常用方法                 | 339 | 15.6.3     | 确认对话框            | 373 |
| 15.2.2                 | 菜单条、菜单、菜单项                  | 340 | 15.6.4     | 颜色对话框            | 374 |
| 15.3                   | 常用组件与布局                     | 342 | 15.6.5     | 文件对话框            | 375 |
| 15.3.1                 | 常用组件                        | 342 | 15.6.6     | 自定义对话框           | 377 |
| 15.3.2                 | 常用容器                        | 344 | 15.7       | 发布 GUI 程序        | 378 |
| 15.3.3                 | 常用布局                        | 345 | 15.8       | 上机实践             | 379 |
| 15.4                   | 处理事件                        | 349 | 15.8.1     | 实验 1 算术测试        | 379 |
|                        |                             |     | 15.8.2     | 实验 2 华容道         | 382 |
|                        |                             |     |            | 习题 15            | 386 |

# 第 1 章

## 初识 Java

### 主要内容

- Java 诞生的原因
- Java 的地位
- 安装 JDK
- 一个简单的 Java 应用程序
- Java 的语言特点

### 难点

- 安装 JDK

在学习 Java 语言之前，部分读者可能学习过 C 语言，熟悉计算机的一些基础知识。学习过 Java 语言之后，可以继续学习和 Java 相关的一些重要内容，比如，如果希望从事编写与数据库相关的软件，可以深入学习 Java Database Connection(JDBC)；如果希望从事 Web 程序的开发，可以学习 Java Server Page(JSP)；如果希望从事手机应用程序的设计，可以学习 Android 手机程序设计；如果希望从事与网络信息交换有关的软件设计，可以学习 eXtensible Markup Language(XML)；如果希望从事大型网络应用程序的开发与设计，可以学习 Java Enterprise Edition(Java EE)，如图 1.1 所示。

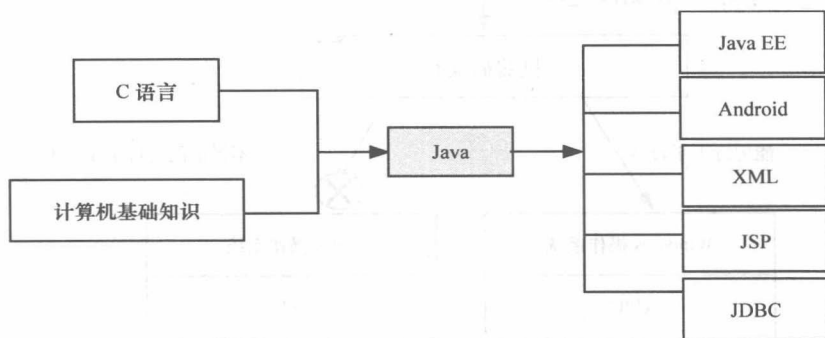


图 1.1 Java 的先导知识与后继技术

## 1.1 Java 诞生的原因

在 Java 诞生之前已经出现了许多优秀的编程语言，比如大家熟悉的 C 语言和 C++ 语言等，那

么是什么原因导致了 Java 语言的诞生呢? Java 语言相对于其他语言, 比如 C 语言和 C++ 语言, 到底有着怎样的特殊优势呢?

Java 语言相对于其他语言的最大优势就是所谓的平台无关性, 即跨平台性, 这也是 Java 最初风靡全球的主要原因。以下通过讲解平台与机器指令, 以及程序的编译、执行来理解 Java 的平台无关性。

### 1. 平台与机器指令

无论哪种编程语言编写的应用程序都需要经过操作系统和处理器来完成程序的运行, 因此这里所指的平台是由操作系统 (OS) 和处理器 (CPU) 所构成。与平台无关是指软件的运行不因操作系统、处理器的变化导致发生无法运行或出现运行错误。

所谓平台的机器指令就是可以被该平台直接识别、执行的一种由 0 和 1 组成的序列代码。需要注意的是相同的 CPU 和不同的操作系统所形成的平台的机器指令可能是不同的, 因此, 每种平台都会形成自己独特的机器指令。比如, 某个平台可能用 8 位序列代码 1000 1111 表示一次加法操作, 以 1010 0000 表示一次减法操作; 而另一种平台可能用 8 位序列代码 1010 1010 表示一次加法操作, 以 1001 0011 表示一次减法操作。

### 2. C/C++ 程序依赖平台

现在, 让我们分析一下为何 C/C++ 语言编写的程序可能因为操作系统的变化、处理器升级导致程序出现错误或无法运行。

C/C++ 语言提供的编译器对 C/C++ 源程序进行编译时, 将针对当前 C/C++ 源程序所在的特定平台进行编译和连接, 然后生成机器指令, 即根据当前平台的机器指令生成机器码文件 (可执行文件)。这样一来, 就无法保证 C/C++ 编译器所产生的可执行文件在所有的平台上都能正确地运行, 这是因为不同平台可能具有不同的机器指令 (如图 1.2 所示)。因此, 如果更换了平台, 可能需要修改源程序, 并针对新的平台重新编译源程序。

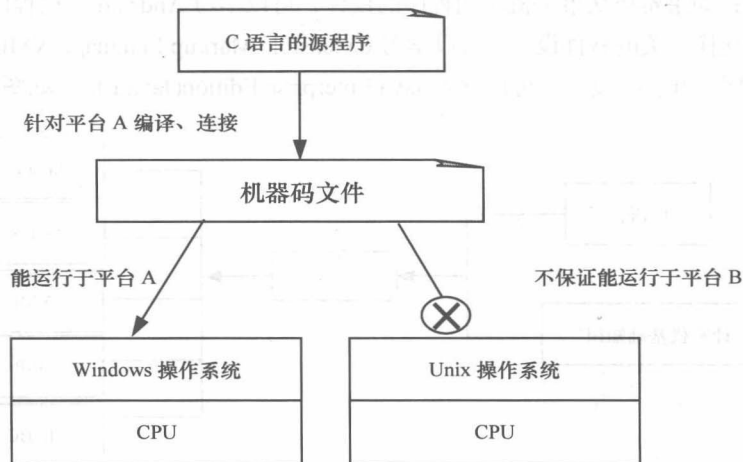


图 1.2 C/C++ 生成的机器码文件依赖平台

### 3. Java 程序不依赖平台

与其他语言相比, Java 语言最大的优势就是它的平台无关性, 这是因为 Java 可以在平台之上再提供一个 Java 运行环境 (Java Runtime Environment, JRE)。该 Java 运行环境由 Java 虚拟机 (Java Virtual Machine, JVM)、类库以及一些核心文件组成。Java 虚拟机的核心是所谓的字节码指令, 即可以被 Java 虚拟机直接识别、执行的一种由 0 和 1 组成的序列代码。字节码并不是机器指令,

因为它不和特定的平台相关，不能被任何平台直接识别、执行。Java 针对不同平台提供的 Java 虚拟机的字节码指令都是相同的，比如所有的虚拟机都将 1111 0000 识别、执行为加法操作。

和 C/C++ 不同的是，Java 语言提供的编译器不针对特定的操作系统和 CPU 芯片进行编译，而是针对 Java 虚拟机把 Java 源程序编译为称作字节码的一种“中间代码”，比如，Java 源文件中的“+”被编译成字节码指令 1111 0000。字节码是可以被 Java 虚拟机识别、执行的代码，即 Java 虚拟机负责解释运行字节码，其运行原理是 Java 虚拟机负责将字节码翻译成虚拟机所在平台的机器码，并让当前平台运行该机器码，如图 1.3 所示。

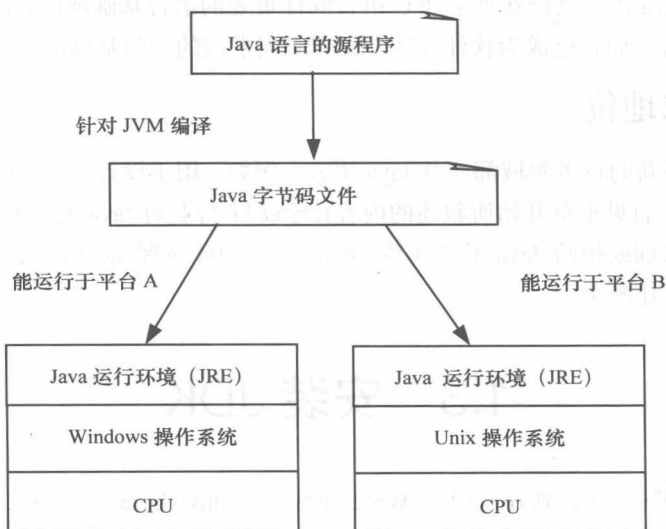


图 1.3 Java 生成的字节码文件不依赖平台

#### 4. Java 之父—James Gosling

1990 年 Sun 公司成立了由詹姆斯·高斯林 (James·Gosling) 领导的开发小组，开始致力于开发一种可移植的、跨平台语言，该语言能生成正确运行于各种操作系统、各种 CPU 芯片上的代码。他们的精心研究和努力促成了 Java 语言的诞生。1995 年 5 月 Sun 公司推出 Java Development Kit(JDK)1.0a2 版本，标志着 Java 的诞生，而 Java 的快速发展得益于 Internet 和 Web 的出现。Internet 上的各种不同计算机可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，Java 的出现标志着真正的分布式系统的到来。



印度尼西亚有一个重要的盛产咖啡的岛屿叫 Java，中文译名为爪哇，开发人员为这种新的语言起名为 Java，其寓意是为世人端上一杯热咖啡。

## 1.2 Java 的地位

### 1.2.1 网络地位

网络已经成为信息时代最重要的交互媒介，那么基于网络的软件设计就成为软件设计领域的

核心。Java 的平台无关性让 Java 成为编写网络应用程序的佼佼者,而且 Java 也提供了许多以网络应用为核心的技术,使得 Java 特别适合于网络应用软件的设计与开发。

## 1.2.2 语言地位

Java 是面向对象编程,并涉及网络、多线程等重要的基础知识,是一门很好的面向对象语言。通过学习 Java 语言不仅可以学习怎样使用对象来完成某些任务,而且可以掌握面向对象编程的基本思想,为今后进一步学习设计模式奠定一个较好的语言基础。C 语言无疑是非常基础和实用的语言之一,目前,Java 语言已经获得了和 C 语言同样重要的语言基础地位,即不仅是一门正在被广泛使用的编程语言,而且已成为软件设计开发者应当掌握的一门基础语言。

## 1.2.3 需求地位

目前,由于很多新的技术领域都涉及 Java 语言,例如,用于设计 Web 应用的 JSP、设计手机应用程序的 Android 等(见本章开始所叙述的内容),导致 IT 行业对 Java 人才的需求正在不断增长,经常可以看到许多培训或招聘 Java 软件工程师的广告,因此掌握 Java 语言及其相关技术意味着较好的就业前景和工作酬金。

# 1.3 安装 JDK

Java 要实现“编写一次,到处运行”(Write once, run anywhere)的目标,就必须提供相应的 Java 运行环境,即运行 Java 程序的平台。目前 Java 平台主要分为下列 3 个版本。

## 1.3.1 三种平台简介

### 1. Java SE

Java SE (曾称为 J2SE)称为 Java 标准版或 Java 标准平台。Java SE 提供了标准的 Java Development Kit(JDK)。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序,也可以开发 Java Applet 程序。当前最新的 JDK 版本为 JDK1.8, Sun 公司把这一最新的版本命名为 JDK8.0,但人们仍然习惯地称作 JDK1.8。

### 2. Java EE

Java EE (曾称为 J2EE)称为 Java 企业版或 Java 企业平台。使用 Java EE 可以构建企业级的服务应用,Java EE 平台包含了 Java SE 平台,并增加了附加类库,以便支持目录管理、交易管理和企业级消息处理等功能。

### 3. Java ME

Java ME (曾称为 J2ME)称为 Java 微型版或 Java 小型平台。Java ME 是一种很小的 Java 运行环境,用于嵌入式的消费产品中,如移动电话、掌上电脑或其他无线设备等。

目前,由于 Android 操作系统的出现,几乎没有手机厂商继续使用 Java ME 平台。Android 操作系统支持 Java 语言,即可以使用 Java 语言编写运行于 Android 操作系统上的应用程序,一款 Android 手机可以通过安装应用软件不断提高自己的应用性能和智能水平。

上述任何一种 Java 运行平台都包括了相应的 Java 虚拟机,虚拟机负责将字节码文件(包括程序使用的类库中的字节码)加载到内存,然后采用解释方式来执行字节码文件,即根据相应平

台的机器指令翻译一句执行一句。

## 1.3.2 安装 Java SE 平台

学习 Java 最好选用 Java SE 提供的 Java 软件开发工具箱 JDK。Java SE 平台是学习掌握 Java 语言的最佳平台，而掌握 Java SE 又是进一步学习 Java EE 和 Android 所必须的。

目前有许多很好的 Java 集成开发环境（IDE）可用，例如 NetBean，Eclipse 等。Java 集成开发环境都将 JDK 作为系统的核心，非常有利于快速地开发各种基于 Java 语言的应用程序。但学习 Java 最好直接选用 Java SE 提供的 JDK，因为 Java 集成开发环境（IDE）的目的是更好、更快地开发程序，不仅系统的界面往往比较复杂，而且也会屏蔽掉一些知识点。在掌握了 Java 语言之后，再去熟悉、掌握一个流行的 Java 集成开发环境（IDE）即可。

可以登录到 Sun 公司的网站（<http://java.sun.com>）免费下载 JDK 1.8。在网站的“Download”菜单中选择“Java SE”，在“Java Platform, Standard Edition”选择界面选择“JDK DOWNLOAD”，接受许可协议后，选择相应的 JDK 版本即可。本书将使用针对 Window 操作系统（32 位）平台的 JDK，因此下载的版本为 jdk-8u25-windows-i586.exe（见图 1.4），如果读者使用其他的操作系统，可以下载相应的 JDK。

| Product / File Description          | File Size | Download                        |
|-------------------------------------|-----------|---------------------------------|
| Linux x86                           | 135.24 MB | jdk-8u25-linux-i586.rpm         |
| Linux x86                           | 154.88 MB | jdk-8u25-linux-i586.tar.gz      |
| Linux x64                           | 135.6 MB  | jdk-8u25-linux-x64.rpm          |
| Linux x64                           | 153.42 MB | jdk-8u25-linux-x64.tar.gz       |
| Mac OS X x64                        | 209.13 MB | jdk-8u25-macosx-x64.dmg         |
| Solaris SPARC 64-bit (SVR4 package) | 137.01 MB | jdk-8u25-solaris-sparcv9.tar.Z  |
| Solaris SPARC 64-bit                | 97.14 MB  | jdk-8u25-solaris-sparcv9.tar.gz |
| Solaris x64 (SVR4 package)          | 137.11 MB | jdk-8u25-solaris-x64.tar.Z      |
| Solaris x64                         | 94.24 MB  | jdk-8u25-solaris-x64.tar.gz     |
| Windows x86                         | 157.26 MB | jdk-8u25-windows-i586.exe       |
| Windows x64                         | 169.62 MB | jdk-8u25-windows-x64.exe        |

图 1.4 选择下载 JDK

双击下载后的 jdk-8u25-windows-i586.exe 文件图标将出现安装向导界面，接受软件安装协议，出现选择安装路径界面。为了便于今后设置环境变量，建议修改默认的安装路径。在这里，我们将默认的安装路径：

```
C:\program Files\Java\jdk1.8.0_25
```

修改为：E:\jdk1.8，如图 1.5 所示。

需要注意的是，安装 JDK 的过程中，JDK 还额外提供一个 Java 运行环境-JRE（Java Runtime Environment），并提示是否修改 JRE 默认的安装路径：

```
C:\program Files\Java\jre1.8.0_25
```

建议采用默认的安装路径。如果修改该默认安装路径，修改后的安装路径不可以与 JDK 的安装路径相同。

将 JDK 安装到 E:\jdk1.8 目录下后，会形成如图 1.6 所示的目录结构。现在，就可以编写 Java 程序并进行编译、运行程序了，因为安装 JDK 的同时，计算机就安装上了 Java 运行环境。

JDK 主要目录内容如下。

- 开发工具：位于 bin 子目录中。指工具和实用程序，可帮助开发、执行、调试以 Java 编程语言编写的程序，例如，编译器 javac.exe 和解释器 java.exe 都位于该目录中。

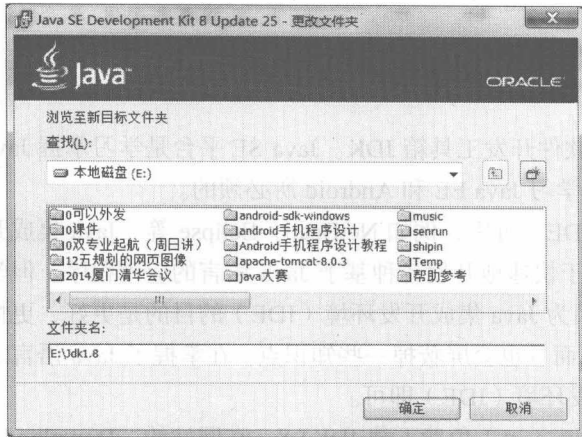


图 1.5 选择 JDK 的安装路径

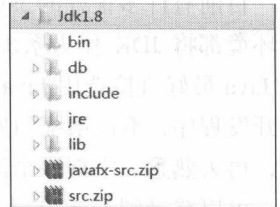


图 1.6 JDK 的目录结构

- **Java 运行环境:** 位于 `jre` 子目录中。Java Runtime Environment (JRE) 包括 Java 虚拟机 (JVM)、类库以及其他支持执行以 Java 编程语言编写的程序的文件。
- **附加库:** 位于 `lib` 子目录中。开发工具所需的其他类库和支持文件。
- **C 头文件:** 位于 `include` 子目录中。支持使用 Java 本机界面、JVM 工具界面以及 Java 平台的其他功能进行本机代码编程的头文件。
- **Derby 数据库:** Java 平台提供了 Derby 数据库管理系统, 或简称 Derby 数据库。Derby 是一个纯 Java 实现、开源的数据库管理系统。安装 JDK 之后 (版本 1.6 之后), 会在安装目录下找到一个名字是 `db` 的子目录, 在该目录下的 `lib` 子目录中提供连接 Derby 数据库所需要的类 (加载驱动程序的类)。
- **源代码:** 位于 JDK 安装目录之根目录中的 `src.zip` 文件是 Java 核心 API 的所有类的 Java 编程语言源文件 (即 `java.*`、`javax.*` 和某些 `org.*` 包的源文件, 但不包括 `com.sun.*` 包的源文件)。

### 1.3.3 设置环境变量

#### 1. 系统环境 path 的设置

JDK 平台提供的 Java 编译器 (`javac.exe`) 和 Java 解释器 (`java.exe`) 位于 Java 安装目录的 `bin` 文件夹中, 为了能在任何目录中使用编译器和解释器, 应在系统特性中设置 `path`。对于 Windows 7/Windows XP, 用鼠标右键单击“计算机”/“我的电脑”, 在弹出的快捷菜单中选择“属性”命令弹出“系统特性”对话框, 再单击该对话框中的“高级系统设置”/“高级选项”, 然后单击按钮“环境变量”, 添加系统环境变量。如果曾经设置过环境变量 `path`, 可单击该变量进行编辑操作, 将需要的值加入即可。需要注意的是, 在编辑环境变量的值时, 如果新加入的值不准备作为环境变量取值范围中的第一个值或最后一个值, 那么新加入的值要和已有的其他值用分号分隔 (如图 1.7 所示); 如果作为最后一个值, 需要和前面的值用分号分隔; 如果作为第一个值需要和后面的值用分号分隔。

#### 2. 系统环境 classpath 的设置

JDK 的安装目录的 `jre` 文件夹中包含着 Java 应用程序运行时所需的 Java 类库, 这些类库被包含在 `jre\lib` 中的压缩文件 `rt.jar` 中。安装 JDK 一般不需要设置环境变量 `classpath` 的值, 如果读者的计算机安装过一些商业化的 Java 开发产品或带有 Java 技术的一些产品, `classpath` 的值可能会被



修改了。那么运行 Java 应用程序时，系统可能加载这些产品所带的老版本的类库，可能导致程序要加载的类无法找到，使程序出现运行错误。读者可以重新编辑系统环境变量 `classpath` 的值。对于 Windows 7/Windows XP 系统，用鼠标右键单击“计算机”/“我的电脑”，在弹出的快捷菜单中选择“属性”命令弹出“系统特性”对话框，再单击该对话框中的“高级系统设置”/“高级选项”，然后单击按钮“环境变量”，添加如图 1.8 所示的系统环境变量。如果曾经设置过环境变量 `classpath`，可单击该变量进行编辑操作，将需要的值加入即可。

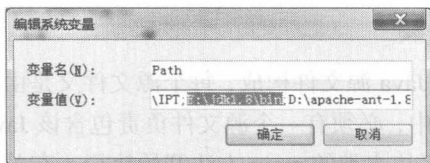


图 1.7 编辑环境变量 path

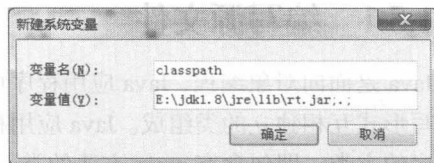


图 1.8 设置环境变量 classpath



环境变量 `classpath` 设置中的“.”是指可以加载应用程序当前目录及其子目录中的类。`rt.jar` 包含了 Java 运行环境提供的类库中的类。

### 3. 仅仅安装 JRE

如果一个平台只想运行 Java 程序，可以只安装 Java 运行环境 JRE。JRE 由 JVM、Java 的核心类以及一些支持文件组成。读者可以登录 Sun 的网站免费下载 JRE。

### 4. 帮助文档

建议下载类库帮助文档(Java SE 8 Documentation:), 如 `jdk-8u25-docs-all.zip`。

## 1.4 Java 程序的开发步骤

Java 程序的开发步骤如图 1.9 所示。

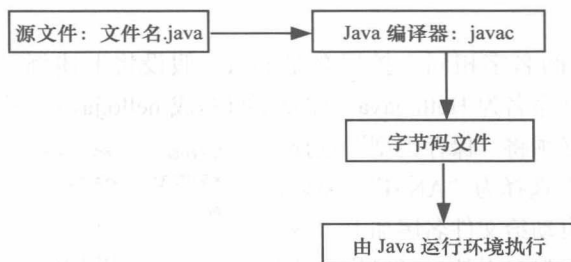


图 1.9 Java 程序的开发过程

### 1. 编写源文件

使用一个文本编辑器，如 Edit 或记事本，来编写源文件。不可使用 Word 编辑器，因为它含有不可见字符。将编写好的源文件保存起来，源文件的扩展名必须是 `.java`。

### 2. 编译 Java 源程序

使用 Java 编译器 (`javac.exe`) 编译源文件，得到字节码文件。