

普通高等教育软件工程专业“十二五”规划教材

# Java基础程序设计

主编 吴正江



科学出版社

普通高等教育软件工程专业“十二五”规划教材

# Java 基础程序设计

主编 吴正江

副主编 张 浩 郑 浩 赵龙德

科学出版社

## 内 容 简 介

本书以 Java SE 核心技术为支点, 全书共分为 9 章, 主要包括 Java 概论与 Eclipse 使用方法, Java 基础语法, 类、对象与接口, 异常处理, 常用类简介, 集合框架, I/O 系统, 多线程技术, GUI 编程。每章后附有习题, 并按章节设定相应实验。全书内容丰富、通俗易懂。

本书可作为普通高等院校本、专科 Java 语言程序设计课程的教学用书, 也可作为计算机等级及计算机培训班等相关 Java 课程的教材和参考书, 还可以作为有关工程技术人员的工具参考书。

### 图书在版编目(CIP)数据

Java 基础程序设计 / 吴正江主编. —北京: 科学出版社, 2013  
普通高等教育软件工程专业“十二五”规划教材  
ISBN 978-7-03-037565-0

I. ①J… II. ①吴… III. ①Java 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 109745 号

责任编辑: 于海云 / 责任校对: 张小霞

责任印制: 闫 磊 / 封面设计: 迷底书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京佳艺恒彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

\*

2013 年 6 月第 一 版 开本: 787×1092 1/16

2013 年 6 月第一次印刷 印张: 21

字数: 530 000

定价: 47.00 元

(如有印装质量问题, 我社负责调换)

# 普通高等教育软件工程专业“十二五”规划教材

## 编 委 会

### 主任委员

李占波 郑州大学软件技术学院副院长

### 副主任委员

车战斌 中原工学院软件学院院长  
刘黎明 南阳理工学院软件学院院长  
刘建华 华北水利水电大学软件学院院长  
乔保军 河南大学软件学院副院长

### 委 员

高 岩 河南理工大学计算机科学与技术学院副院长  
邓璐娟 郑州轻工业学院软件学院副院长  
史玉珍 平顶山学院软件学院副院长  
周文刚 周口师范学院计算机科学与技术学院副院长  
席 磊 河南农业大学信息与管理科学学院系主任  
陈建辉 郑州航空工业管理学院计算机科学与技术系副主任  
张永强 河南财经政法大学计算机与信息工程学院副院长  
郑延斌 河南师范大学计算机与信息工程学院副院长  
谭营军 河南职业技术学院信息工程系副主任  
赵素萍 洛阳师范学院信息技术学院软件工程系主任  
潘 红 新乡学院计算机与信息工程学院院长

## 《Java 基础程序设计》编委会

主 编 吴正江

副主编 张 浩 郑 浩 赵龙德

参 编 刘俊男 孙 楠 窦桂琴 毋 东

# 前 言

Java 是一种可以撰写跨平台应用软件的面向对象的程序设计语言，Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网。Java 已经无处不在，从互联网网站到 Android 手机，Java 开发者们锐意创新，使得 Java 技术已经成为世界上最卓越企业应用和移动应用的开发平台之一。

本书作为 Java 语言的入门教程，立足于 J2SE，对知识进行优化组合，针对无程序设计功底的人群，力争突出重点、循序渐进，使读者可以快速掌握核心技术，并为后续学习打下基础。

本书特点如下：

(1) 脉络清晰，易学好用。全书共分为 9 章，各章安排前后有序，突出各部分核心知识，力争给读者营造一个更纯洁的知识环境，方便快速掌握 J2SE 核心知识。

(2) 案例丰富，注释合理。全书共有近 200 个程序示例，均在核心部分给予合理注释，并给出到位的分析。

(3) 适用面广，实用性强。本书兼顾了零起点和有语言基础的读者，从目标上说，本书适合 Java 程序设计基础目标和进阶目标的读者。

本书由河南理工大学吴正江和毋东、河南农业大学张浩、郑州师范学院赵龙德、平顶山学院郑浩、河南大学刘俊男、河南财经政法大学孙楠、中原工学院窦桂琴 8 位老师共同合作完成。毋东负责第 1 章、第 5 章及第 9 章第 9 节的编写，孙楠负责第 2 章的编写，吴正江负责第 3 章和本书其他部分的编写，窦桂琴负责第 4 章的编写，郑浩负责第 6 章的编写，刘俊男负责第 7 章的编写，张浩负责第 8 章及本书实验的编写，赵龙德负责第 9 章其余部分的编写。全书由吴正江统稿。

本书在编写过程中得到了河南省计算机协会、河南理工大学高岩教授的支持和鼓励。

限于编者水平，书中难免存在欠妥之处，恳请读者和同仁批评指正。

编 者

2013 年 5 月

# 目 录

## 前言

|       |                         |    |
|-------|-------------------------|----|
| 第 1 章 | Java 语言基础               | 1  |
| 1.1   | 面向对象编程                  | 1  |
| 1.1.1 | 面向对象的程序设计思想             | 1  |
| 1.1.2 | 面向对象编程的主要特征             | 2  |
| 1.2   | Java 程序设计语言概述           | 3  |
| 1.2.1 | Java 发展历史               | 3  |
| 1.2.2 | Java 语言特点               | 5  |
| 1.3   | Java 开发环境搭建             | 6  |
| 1.3.1 | 下载 JDK                  | 6  |
| 1.3.2 | 安装 JDK                  | 7  |
| 1.3.3 | 环境配置和测试                 | 8  |
| 1.4   | Java 集成开发环境             | 10 |
| 1.4.1 | Eclipse 简介              | 11 |
| 1.4.2 | Eclipse 安装配置            | 11 |
| 1.4.3 | 利用 Eclipse 编写 Java 程序步骤 | 13 |
|       | 小结                      | 16 |
|       | 习题                      | 16 |
| 第 2 章 | Java 基础语法               | 17 |
| 2.1   | Java 应用程序               | 17 |
| 2.1.1 | 如何开始你的第一个程序             | 17 |
| 2.1.2 | 如何输出                    | 18 |
| 2.1.3 | 常见错误                    | 19 |
| 2.1.4 | 万事皆对象                   | 20 |
| 2.2   | 标识符、关键字与注释              | 20 |
| 2.2.1 | 标识符                     | 20 |
| 2.2.2 | 关键字                     | 21 |
| 2.2.3 | 注释                      | 22 |
| 2.3   | Java 数据类型               | 23 |
| 2.3.1 | 基础数据类型                  | 23 |
| 2.3.2 | 变量                      | 23 |
| 2.3.3 | 语句与变量的生命周期              | 24 |
| 2.3.4 | 常量                      | 26 |
| 2.3.5 | 转义字符                    | 26 |

|            |                     |           |
|------------|---------------------|-----------|
| 2.4        | 运算法与表达式             | 27        |
| 2.4.1      | 运算符的优先级与结合性         | 27        |
| 2.4.2      | 算术运算符               | 28        |
| 2.4.3      | 关系运算符               | 31        |
| 2.4.4      | 逻辑运算符               | 32        |
| 2.4.5      | 条件运算符               | 33        |
| 2.4.6      | 按位运算符               | 34        |
| 2.4.7      | 移位运算符               | 34        |
| 2.4.8      | 赋值运算符               | 35        |
| 2.4.9      | 类型转换                | 36        |
| 2.5        | 控制结构                | 38        |
| 2.5.1      | 顺序结构                | 38        |
| 2.5.2      | 选择结构                | 39        |
| 2.5.3      | 循环结构                | 44        |
| 2.6        | 方法                  | 52        |
| 2.6.1      | 方法概述                | 52        |
| 2.6.2      | 简单的方法定义与调用          | 53        |
| 2.7        | 枚举                  | 56        |
|            | 小结                  | 57        |
|            | 习题                  | 58        |
| <b>第3章</b> | <b>类、对象与接口</b>      | <b>60</b> |
| 3.1        | 面向对象的程序设计概述         | 60        |
| 3.1.1      | 面向对象程序设计方法概述        | 60        |
| 3.1.2      | 面向对象程序设计方法的原则       | 61        |
| 3.1.3      | 面向对象程序设计的特点         | 61        |
| 3.2        | 类与对象                | 63        |
| 3.2.1      | 建立 Java 中的类         | 63        |
| 3.2.2      | 建立引用类型变量、对象及它们之间的关联 | 66        |
| 3.2.3      | 访问对象中的成员            | 69        |
| 3.2.4      | 类定义中的多态——重载         | 72        |
| 3.2.5      | 类定义中的其他问题           | 73        |
| 3.3        | 继承                  | 80        |
| 3.3.1      | 继承的功能               | 80        |
| 3.3.2      | 继承的定义               | 81        |
| 3.3.3      | 成员的隐藏和重写            | 84        |
| 3.3.4      | 引用类型的转型             | 86        |
| 3.3.5      | 继承中的构造方法            | 88        |
| 3.4        | 抽象类                 | 93        |
| 3.4.1      | 抽象类的功能              | 93        |

|            |                               |     |
|------------|-------------------------------|-----|
| 3.4.2      | 抽象类的定义                        | 93  |
| 3.5        | 接口                            | 95  |
| 3.5.1      | 接口的作用                         | 96  |
| 3.5.2      | 接口的定义                         | 96  |
| 3.5.3      | 接口的使用                         | 98  |
| 3.6        | 多态                            | 100 |
| 3.6.1      | 多态的概念                         | 101 |
| 3.6.2      | 重载与多态                         | 101 |
| 3.6.3      | 重写与多态                         | 102 |
| 3.7        | 内部类                           | 103 |
| 3.7.1      | 成员内部类                         | 104 |
| 3.7.2      | 局部内部类                         | 105 |
| 3.7.3      | 静态内部类                         | 106 |
| 3.7.4      | 匿名内部类                         | 107 |
| 小结         |                               | 109 |
| 习题         |                               | 110 |
| <b>第4章</b> | <b>异常处理</b>                   | 112 |
| 4.1        | 异常基础                          | 112 |
| 4.2        | 异常类型                          | 113 |
| 4.3        | 异常处理                          | 115 |
| 4.3.1      | 主动处理异常——关键字 try、catch、finally | 116 |
| 4.3.2      | 被动处理异常——关键字 throws            | 118 |
| 4.4        | 异常主动抛出                        | 119 |
| 4.5        | 自定义异常                         | 121 |
| 小结         |                               | 123 |
| 习题         |                               | 123 |
| <b>第5章</b> | <b>常用基础类库</b>                 | 125 |
| 5.1        | 包                             | 125 |
| 5.1.1      | 包的概念                          | 125 |
| 5.1.2      | 包的创建                          | 126 |
| 5.1.3      | 包与目录                          | 126 |
| 5.1.4      | 包的引用                          | 127 |
| 5.1.5      | 包与访问权限                        | 127 |
| 5.2        | Object 类                      | 128 |
| 5.2.1      | toString()方法                  | 128 |
| 5.2.2      | equals()方法                    | 129 |
| 5.3        | 基本数据类型包装类                     | 130 |
| 5.3.1      | 自动装箱与拆箱                       | 131 |
| 5.3.2      | 自动装拆箱注意事项                     | 132 |

|              |                          |            |
|--------------|--------------------------|------------|
| 5.4          | 字符串类 .....               | 133        |
| 5.4.1        | String 类 .....           | 133        |
| 5.4.2        | StringBuffer 类 .....     | 137        |
| 5.4.3        | StringBuilder 类 .....    | 139        |
| 5.5          | Calendar 类与 Date 类 ..... | 140        |
| 5.5.1        | Calendar 类 .....         | 140        |
| 5.5.2        | Date 类 .....             | 142        |
| 5.6          | Math 类 .....             | 143        |
|              | 小结 .....                 | 143        |
|              | 习题 .....                 | 143        |
| <b>第 6 章</b> | <b>集合框架 .....</b>        | <b>145</b> |
| 6.1          | 数组 .....                 | 145        |
| 6.1.1        | 一维数组 .....               | 146        |
| 6.1.2        | 二维数组 .....               | 150        |
| 6.1.3        | 三维数组及三维以上数组 .....        | 152        |
| 6.1.4        | Arrays 类 .....           | 153        |
| 6.2          | 容器类 .....                | 154        |
| 6.2.1        | Collection 接口 .....      | 155        |
| 6.2.2        | List 接口 .....            | 156        |
| 6.2.3        | Set 接口 .....             | 161        |
| 6.2.4        | Map 接口 .....             | 164        |
| 6.2.5        | Collections 类 .....      | 168        |
| 6.2.6        | Java 对象集合的遍历 .....       | 169        |
| 6.3          | 泛型 .....                 | 171        |
| 6.3.1        | 泛型的概念和作用 .....           | 171        |
| 6.3.2        | 泛型接口和泛型类 .....           | 172        |
|              | 小结 .....                 | 174        |
|              | 习题 .....                 | 175        |
| <b>第 7 章</b> | <b>I/O 系统 .....</b>      | <b>176</b> |
| 7.1          | I/O 系统概述 .....           | 176        |
| 7.1.1        | 流的概念 .....               | 176        |
| 7.1.2        | 流的分类 .....               | 177        |
| 7.2          | 文件的处理 .....              | 180        |
| 7.2.1        | File 类 .....             | 180        |
| 7.2.2        | 文件节点输入/输出流 .....         | 191        |
| 7.2.3        | 文件处理输入/输出流 .....         | 200        |
| 7.2.4        | 数据、对象的输入/输出流 .....       | 206        |
| 7.3          | 标准流 .....                | 214        |
| 7.3.1        | System.out .....         | 214        |

|       |                    |     |
|-------|--------------------|-----|
| 7.3.2 | System.in .....    | 216 |
| 7.4   | 其他流简介 .....        | 218 |
| 7.4.1 | 文件随机存取流 .....      | 218 |
| 7.4.2 | 打印流 .....          | 220 |
| 7.4.3 | 内存流 .....          | 221 |
| 7.4.4 | 线程流 .....          | 222 |
| 7.4.5 | 压缩流 .....          | 222 |
| 小结    | .....              | 224 |
| 习题    | .....              | 224 |
| 第 8 章 | 多线程 .....          | 225 |
| 8.1   | 线程简介 .....         | 225 |
| 8.1.1 | 进程与线程 .....        | 225 |
| 8.1.2 | 线程生命周期及其状态转换 ..... | 227 |
| 8.1.3 | 线程优先级与线程调度 .....   | 228 |
| 8.2   | 多线程的实现 .....       | 229 |
| 8.2.1 | Thread 类 .....     | 229 |
| 8.2.2 | Runnable 接口 .....  | 231 |
| 8.3   | 线程管理 .....         | 232 |
| 8.3.1 | 创建 .....           | 232 |
| 8.3.2 | 终止 .....           | 234 |
| 8.3.3 | 同步 .....           | 237 |
| 8.3.4 | 通信 .....           | 247 |
| 8.3.5 | 中断与阻塞 .....        | 249 |
| 8.3.6 | 死锁 .....           | 251 |
| 小结    | .....              | 256 |
| 习题    | .....              | 256 |
| 第 9 章 | 图形用户界面 .....       | 257 |
| 9.1   | GUI 概述 .....       | 257 |
| 9.1.1 | Java 基础类 .....     | 257 |
| 9.1.2 | AWT 工具包 .....      | 257 |
| 9.1.3 | Swing 工具包 .....    | 258 |
| 9.2   | Swing 容器 .....     | 259 |
| 9.2.1 | Swing 容器层次 .....   | 259 |
| 9.2.2 | JFrame 框架窗口 .....  | 260 |
| 9.3   | 颜色、字体和图形 .....     | 261 |
| 9.3.1 | Color 类 .....      | 262 |
| 9.3.2 | Font 类 .....       | 262 |
| 9.3.3 | Graphics 类 .....   | 262 |
| 9.4   | Swing 基本组件 .....   | 264 |

|        |                  |     |
|--------|------------------|-----|
| 9.4.1  | 标签               | 264 |
| 9.4.2  | 按钮               | 265 |
| 9.4.3  | 文本框              | 266 |
| 9.4.4  | 文本域              | 268 |
| 9.4.5  | 复选框和单选框          | 269 |
| 9.5    | 布局管理器            | 271 |
| 9.5.1  | 流布局管理器           | 272 |
| 9.5.2  | 边界布局管理器          | 273 |
| 9.5.3  | 网格布局管理器          | 274 |
| 9.5.4  | 网格包布局管理器         | 276 |
| 9.5.5  | 容器嵌套             | 278 |
| 9.6    | 事件驱动设计           | 280 |
| 9.6.1  | 委托事件模型           | 280 |
| 9.6.2  | 事件类              | 283 |
| 9.6.3  | ActionEvent 类    | 284 |
| 9.6.4  | KeyEvent 类       | 286 |
| 9.6.5  | MouseEvent 类     | 287 |
| 9.6.6  | WindowEvent 类    | 290 |
| 9.7    | Swing 其他常用组件     | 291 |
| 9.7.1  | 列表               | 291 |
| 9.7.2  | JComboBox 组件     | 292 |
| 9.7.3  | JScrollBar 组件    | 294 |
| 9.8    | 对话框的使用           | 295 |
| 9.8.1  | JDialog 对话框      | 295 |
| 9.8.2  | JOptionPane 对话框  | 297 |
| 9.8.3  | JFileChooser 对话框 | 299 |
| 9.9    | 菜单制作             | 301 |
| 9.10   | 观感效果             | 306 |
| 9.11   | Applet 小程序       | 308 |
| 9.11.1 | Applet 的基本工作原理   | 308 |
| 9.11.2 | Applet 生命周期      | 308 |
| 9.11.3 | Applet 与 HTML    | 310 |
| 小结     |                  | 313 |
| 习题     |                  | 314 |
| 附录     | 实验               | 315 |

# 第 1 章 Java 语言基础

## 本章内容:

在本章中, 介绍了面向对象 (Object-Oriented, OO) 的程序设计思想、Java 语言历史及特点、Java 面向对象程序设计, 以及 Java 开发环境搭建和 Java 开发工具 Eclipse。

## 学习目标:

- 了解面向对象程序设计思想
- 了解 Java 发展历史及特点
- 掌握 Java 开发环境搭建
- 了解主流的 Java 开发工具
- 掌握 Eclipse 的安裝配置

## 1.1 面向对象编程

### 1.1.1 面向对象的程序设计思想

面向对象 (Object-Oriented, OO) 是一种程序设计的思想, 它起源于 Kisten Nygaard 和 Ole Johan Dahl 于 20 世纪 60 年代开发的 Simula 语言, 然而真正的面向对象编程 (Object-Oriented Programming, OOP) 由 20 世纪 70 年代初 Palo Alto 研究中心的 Alan Kay 所在的研究小组开发出的 Smalltalk 语言奠基。正是通过 Smalltalk 的研制与推广应用, 使人们注意到面向对象方法所具有的模块化、信息封装与隐蔽、抽象性、继承性、多样性等独特之处, 这些优异特性为研制大型软件、提高软件可靠性、可重用性、可扩充性和可维护性提供了有效的手段和途径。

Smalltalk 对后来出现的面向对象语言, 如 Object-C、C++等都产生了深远的影响。与此同时, 随着面向对象语言的出现, 面向对象程序设计也就应运而生且得到迅速发展。由于面向对象编程也是 Java 编程语言的核心, 因此在系统地学习 Java 语言之前, 需要首先对面向对象的程序设计思想有一个初步的了解, 并且在以后的学习过程中要不断地深入理解并掌握面向对象的程序设计思想。那么, 什么是面向对象编程呢? 它具有什么样的优势呢?

面向对象编程是将客观事物看做具有状态和行为的对象, 并通过抽象找出同一类对象的共同状态 (静态特征) 和行为 (动态特征), 然后构成类的一种程序设计思想。这里需要指出的是, 对象是现实世界中随处可见的人、事、物体等多种实体, 如天上飞的鸟、地上走的人、水里游的鱼等, 即现实世界中的一切事物皆是对象。对象具有的状态和行为即对象自身具有的结构。在面向对象编程中, 对象是类的实例, 类是用来创建对象的模板。例如, 你和我作为对象, 都是人 (Person) 这个类的实例, 而且都具有一些共同的状态属性, 如姓名、籍贯、身高、年龄和体重等许多属性。同时我们每一个人还具有一些说话、工作、吃饭、睡觉、读

书、写字等行为。然而，我们每一个人之所以都是独一无二的，是因为我们每个人的这些属性都有特定的值。

在 Java 程序中，所有对象都是由类创建的。一个 Java 程序中的类主要包括以下两部分：

- (1) 类的声明。
- (2) 类的主体。

在 Java 中定义上文 Person 类的部分代码如下：

```
//代码 1-1
class Person{
    String name;           // 定义姓名成员变量
    String nativePlace;    // 定义籍贯成员变量
    String height;         // 定义身高成员变量
    String age;            // 定义年龄成员变量
    String weight;         // 定义体重成员变量

    // ...

    public void Talk() {    // 定义说话行为
        // ...
    }

    // ...
    public void Eat() {     // 定义吃饭行为
        // ...
    }
}
```

### 1.1.2 面向对象编程的主要特征

在面向对象程序设计方法出现以前，人们都是采用以事件为中心的面向过程（Procedure-Oriented，PO）编程方法。面向过程编程方法的主要思路是按功能将程序结构划分为若干个基本模块，这些模块功能上相对独立，模块之间关系尽可能简单，并依照自顶向下、逐步求精的原则实现形成树状结构的各个模块。虽然面向过程的编程方法具有很多优点，但是它把数据和处理数据的过程分离为相互独立的实体，一旦数据结构发生改变，将导致所有相关程序都有修改，使得程序可重用性差。所以，相对于面向过程程序设计方法，面向对象程序设计方法具有以下主要特征和优势。

#### 1) 封装性

封装是一种信息隐蔽技术，是面向对象编程的核心思想之一。封装使数据和加工该数据的方法封装为一个整体，以实现独立性很强的模块，使得用户只能见到对象的外特性，而对象的内特性对用户是隐蔽的。封装的目的在于把对象的设计者和对象者的使用分开，使用者不必知晓行为实现的细节，有效避免了外部错误对内部数据的影响，实现了错误局部化。同时当一个对象的内部结构或实现方法改变时，由于只要保持对象的接口不改变，就不用改变程序的其他部分，从而提高了程序的可维护性。

#### 2) 继承性

继承性是子类自动共享父类之间数据和方法的机制。它由类的派生功能体现。一个子类

可以直接继承其父类的全部描述，同时还可修改和扩充。继承具有传递性。继承分为单继承（一个子类只有一父类）和多重继承（一个类有多个父类）。类的对象是各自封闭的，若没继承性机制，则类对象中的数据、方法就会出现大量重复。继承不仅支持系统的可重用性，而且还促进系统的可扩充性。

### 3) 多态性

对象根据所接收的消息而做出动作，同一消息被不同的对象接收时可产生完全不同的行动，这种现象称为多态性。在面向对象编程中可通过在派生类中重定义基类函数来实现多态性。利用多态性用户可发送一个通用的信息，而将所有的实现细节都留给接收消息的对象自行决定，也就是说，同一消息可调用不同的方法。例如，Print 消息被发送到一个图或表时所调用的打印方法与将同样的 Print 消息发送给一文本文件所调用的打印方法会完全不同。多态性的实现受到继承性的支持，利用类继承的层次关系，把具有通用功能的协议存放在类层次中尽可能高的地方，而将实现这一功能的不同方法置于较低层次，这样，在这些低层次上生成的对象就能给通用消息以不同的响应。

综上可知，在面对对象方法中，类和继承是适应人们一般思维方式的描述范式。通过继承可以大幅减少冗余的代码，方便地扩展现有代码，提高编码效率，降低出错概率，降低软件维护的难度。通过保持外部接口不变的情况下改变内部实现，减少甚至避免外界的干扰。

## 1.2 Java 程序设计语言概述

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的面向对象程序设计语言。Sun 公司在《Java 白皮书》中这样对 Java 进行定义：Java: A simple, object-oriented, distributed, interpreted, 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、解释型的、健壮的、安全的、结构中立的、可移植的、高性能的、多线程的动态语言。Java 语言具有优秀的跨平台特性，所编译的程序能够在多种操作系统上运行，即“一次编写，到处运行（Write Once, Run Anywhere）”。

### 1.2.1 Java 发展历史

Java 语言源于 Oak 语言，这是 Sun 公司在一项消费性电子产品软件发展方案中采用的语言。Oak 的设计目标是用以开发可靠、紧凑、易于移植的分布式嵌入系统。尽管 Oak 语言在技术上颇为成功，但由于商业上的原因，却未能在市场的激烈竞争中站稳脚跟。然而，Internet 和 WWW 日新月异的发展却为 Oak 创造了新的生存空间。之后，James Gosling 对 Oak 进行了小规模的改造，成为一种非常适合网络开发的独特语言，这就是在 1995 年 5 月 23 日诞生的 Java。Java 的诞生标志着互联网时代的开始。由于 Java 恰逢其时地诞生于 Internet 和 WWW 蓬勃发展的时期，经过多年的蓬勃发展，致使如今的 Java 与当初的 Oak 语言已不可同日而语，使得它逐渐成为 Internet 上受欢迎的开发与编程语言，一些闻名的计算机公司纷纷购买了 Java 语言的使用权，如 Microsoft、IBM、Netscape、Novell、Apple 等。而且 Sun 公司采用了一种“开放”策略，将 Java 技术公之于众，更激发了第三方软件开发商的极大热情，纷纷发展 Java 技术，因此 Java 的前景是非常广阔的。

Java 的主要发展历史简述如下：

1995 年 5 月 23 日，Java 语言诞生。Java 的诞生标志着互联网时代的开始，它能够被应用在全球信息网络的平台之上编写互动性极强的 Applet 程序，而 1995 年的 Applet 无疑能够给人们无穷的视觉和脑力震荡。其实 Java 的诞生颇有那么一股“有心栽花花不开，无心插柳柳成荫”的味道。

1996 年 1 月，第一个 JDK1.0 诞生。

1996 年 2 月，Sun 发布 Java 芯片系列，包括 picoJava、microJava 和 UltraJava。并推出 Java 数据库连接 JDBC 数据库 API。

1996 年 4 月，10 个最主要的操作系统供应商声明将在其产品中嵌入 Java 技术。

1996 年 5 月，Sun 在大会上推出一系列 Java 平台新技术，如 JavaMediaAPIs、CommerceAPIs、SecurityAPIs、ServletAPIs、ManagementAPIs 等。

1996 年 9 月，约 8.3 万个网页应用了 Java 技术来制作，同时推出 Java 虚拟机规范和 Java 类库。

1996 年 10 月，Sun 完成 JavaBeans 规范并发布，并发布第一个 JavaJIT（Just-In-Time）编译器。

1997 年 2 月 18 日，JDK1.1 发布，它蕴藏着新的翻天覆地的革命。在 JDK1.1 时代，Java 平台分为 PersonalJava 与 EmbeddedJava。PersonalJava 比较适用于运算资源和内存丰富的设备，而 EmbeddedJava 适用于资源有限的设备。

1998 年 12 月 8 日，Sun 发布了 Java 历史上最重要的一个 JDK 版本，即 JDK1.2。这个版本标志着 Java 已经进入 Java2 时代，这个时期也是 Java 飞速发展的时期。JDK1.2 分成 J2EE、J2SE 和 J2ME 三大块，得到了市场的强烈反响。相对于之前的 JDK，JDK1.2 增加了最吸引眼球的 Swing。

2000 年 5 月 8 日，JDK1.3 发布。在 JDK1.3 时代，第一个稳定版本的 Tomcat3.x 得到了广泛的应用，WebLogic 等商业应用服务器也渐渐被接受。

2002 年 2 月 26 日，JDK1.4 发布。由于 Compaq、Fujitsu、SAS、Symbian、IBM 等公司的参与，使得 JDK1.4 发展很快。在此时间，.NET 平台和 Java 平台之间曾经发生了一次声势浩大的孰优孰劣的论战，焦点在于 Java 的性能问题。

2004 年 9 月 30 日 18 点，J2SE1.5 横空出世，这是 Java 语言发展史上的里程碑事件。为了表示这个版本的重要性，J2SE1.5 更名为 J2SE5.0。此版本增加了诸如泛型、增强的 for 语句、可变量数参数、注释、自动拆箱和装箱等功能，推出了 EJB3.0 规范及 JSF 等。利用 JSF 可以很快地建立起复杂的 JSP 界面。

2005 年，J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

2006 年 12 月，发布了 Java SE6.0。此版本不仅在性能、易用性等方面得到了前所未有的提高，而且还提供了如脚本、全新的 API 的支持。

2009 年 4 月 20 日，Oracle 公司并购了 Sun 公司。

2011 年 7 月 28 日，Oracle 正式发布 Java 7。这是自 2006 年 12 月 Sun 发布 Java SE 6 以来的首个重要版本，也是 Sun 被 Oracle 收购以来发行的第一个版本，包含了 Java 诞生以来最大的改进。

现在，Java 已经成为开发和部署企业应用程序的首选语言。Java 2 的计算平台以 Java 语言为中心，其体系结构与平台无关。目前，它有三个独立的版本。

### 1) Java SE

Java 技术标准版 (Java Standard Edition)。Java SE 主要针对包含丰富的 GUI (图形用户界面)、复杂逻辑和高性能的桌面应用程序。

### 2) Java EE

Java 技术企业版 (Java Enterprise Edition)。Java EE 是一个标准的多层体系结构,适用于开发和部署分布式、基于组件、安全可靠、可伸缩和易于管理的企业应用程序,以减少开发分布式应用程序的复杂性和代价,简化开发和部署过程。Java EE 体系结构将一个企业应用程序分隔为客户层、表示层、业务层和数据层。以服务器端程序和企业软件的开发为目标。

### 3) Java ME

Java 技术微型版 (Java Micro Edition)。它是针对具有有限的连接、内存和用户界面能力的设备应用程序而设计,如移动电话、PDA、互联移动设备、嵌入式设备、能够接入电缆服务的机顶盒等。

## 1.2.2 Java 语言特点

Java 语言具有鲜明的特点,使它在分布式网络应用、多平台应用、图形用户界面、Web 应用、多线程应用等软件的开发中成为方便高效的工具。下面简要进行介绍。

(1) 开发和使用简单性。Java 的语法风格非常近似于 C++ 语言,但删改了 C++ 中的指针、操作符重载等一些易混淆的地方。在内存管理方面又提供了垃圾收集机制。这使程序员可以在实现程序功能方面投入更多的精力,而无须考虑诸如内存释放等枝节问题。C++ 中复杂而灵活的指针操作往往导致严重的错误,一向是开发调试人员深感棘手的问题,而这在 Java 中却不存在。Java 虚拟机还能程序链接本地甚至远程的类库,开发人员不必关注其细节。

(2) 分布式。Java 对 TCP/IP 协议 (如 HTTP、FTP) 的支持使 Java 程序可以轻易地建立网络连接,并通过统一资源定位器 (Uniform Resource Locator, URL) 访问远程文件,如同访问本地文件一样方便。Java 的运行系统能动态地通过网络装入字节码,动态使用新的协议控制软件。

(3) 面向对象性。Java 对面向对象的要求十分严格,不允许定义独立于类的变量和方法。Java 以类和对象为基础,任何变量和方法都只能包含于某个类的内部。这就使程序的结构更为清晰,为继承和重用带来便利。Java 提供了众多的一般对象的类,通过继承即可使用父类的方法。在 Java 中,类的继承关系是单一的非多重的,一个子类只有一个父类。Java 提供的 Object 类及其子类的继承关系如同一棵倒立的树。根类为 Object 类, Object 类功能强大,经常会使用到它及其派生的子类。

(4) 安全性。对于网络应用来说,这一点是极为重要的。Java 将安全性作为第一考虑,设置了层层防范。首先在编译时进行语法、语义的检查。链接时,还要再进行一遍编译级的类型检查,消除间接对象访问。运行时,Java 的运行系统将进行字节码检验,并记录对象的存储情况,将访问限制在安全范围之内。本地的类与远程的类分开运行,阻止远程系统对本地系统的破坏。支持 Java 的浏览器还允许用户控制 Java 软件对本地系统的访问。各种措施的综合使 Java 程序的安全性得到保证。

(5) 平台独立性和可移植性。Java 的应用程序接口 (API) 和运行时系统是可移植性的关键。Java 为支持它的各种操作系统提供了一致的 API。在 API 界面上,所有 Java 程序将都不依赖于平台。Java 的运行系统在解释执行程序时,将字节码转化为当前机器的机器码。程序开发人员无须考虑使用应用时的硬件条件和操作系统结构,用户只需有 Java 的运行系统,就可运行编译过的字节码。