

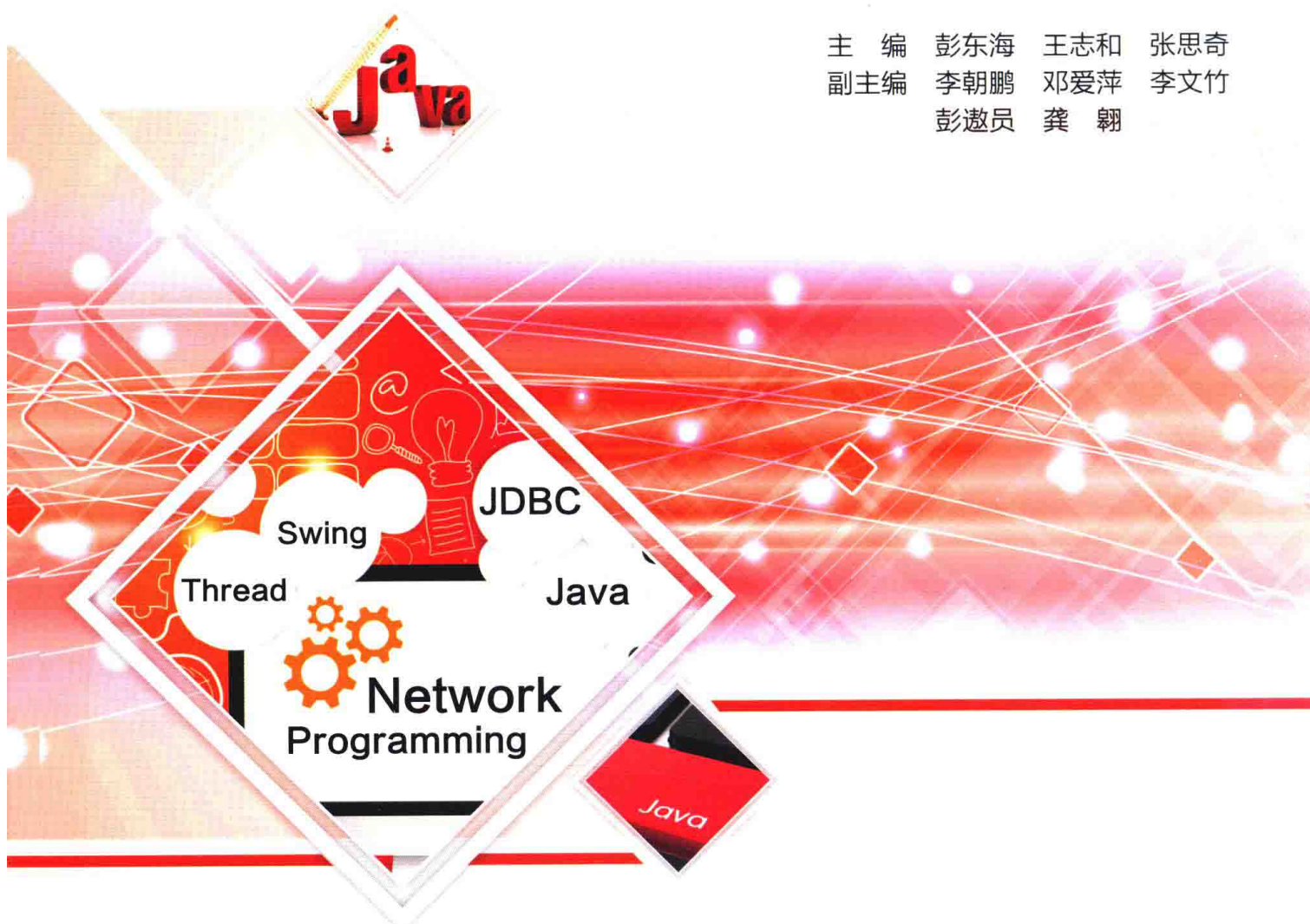
▪ 应用型高等院校校企合作创新示范教材 ▪



# Java

## 基础应用与实战

主 编 彭东海 王志和 张思奇  
副主编 李朝鹏 邓爱萍 李文竹  
彭邀员 龚 翱



中国水利水电出版社  
www.waterpub.com.cn

应用型高等院校校企合作创新示范教材

# Java 基础应用与实战

主 编 彭东海 王志和 张思奇

副主编 李朝鹏 邓爱萍 李文竹 彭遨员 龚 翱



中国水利水电出版社  
www.waterpub.com.cn

• 北京 •

## 内 容 提 要

本书是一本关于 Java 的计算机编程教材,共分 12 章,每个章节里有大量的实例来介绍 Java 语言和面向对象程序设计方法。本书的主要内容:Java 语法基础,Java 环境搭建,Java 书写规范,Java 变量与函数、数据类型、运算符、数组、控制语句、类与对象、方法与包、继承与多态,Java 常用类、抽象类与接口、泛型与集合、异常处理、图形界面设计、多线程、Swing 程序设计和数据库编程。

本书可作为高等院校本专科计算机相关专业的程序设计课程教材,也可作为 Java 程序开发的技术人员提供一定的参考。

## 图书在版编目(CIP)数据

Java 基础应用与实战 / 彭东海, 王志和, 张思奇主  
编. — 北京: 中国水利水电出版社, 2019.3  
应用型高等院校校企合作创新示范教材  
ISBN 978-7-5170-7541-7

I. ①J… II. ①彭… ②王… ③张… III. ①JAVA 语  
言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第051227号

策划编辑: 周益丹

责任编辑: 周益丹

封面设计: 梁 燕

|      |                                                                                                                                                                                                                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 书 名  | 应用型高等院校校企合作创新示范教材<br>Java 基础应用与实战<br>Java JICHU YINGYONG YU SHIZHAN                                                                                                                                                                                                                                 |
| 作 者  | 主 编 彭东海 王志和 张思奇<br>副主编 李朝鹏 邓爱萍 李文竹 彭邀员 龚 翱                                                                                                                                                                                                                                                          |
| 出版发行 | 中国水利水电出版社<br>(北京市海淀区玉渊潭南路 1 号 D 座 100038)<br>网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a><br>E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水)<br><a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a><br>电话: (010) 68367658 (营销中心)、82562819 (万水) |
| 经 售  | 全国各地新华书店和相关出版物销售网点                                                                                                                                                                                                                                                                                  |
| 排 版  | 北京万水电子信息有限公司                                                                                                                                                                                                                                                                                        |
| 印 刷  | 三河市鑫金马印装有限公司                                                                                                                                                                                                                                                                                        |
| 规 格  | 184mm×260mm 16 开本 16.75 印张 409 千字                                                                                                                                                                                                                                                                   |
| 版 次  | 2019 年 3 月第 1 版 2019 年 3 月第 1 次印刷                                                                                                                                                                                                                                                                   |
| 印 数  | 0001—3000 册                                                                                                                                                                                                                                                                                         |
| 定 价  | 39.00 元                                                                                                                                                                                                                                                                                             |

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换  
版权所有·侵权必究

# 前 言

Java 语言是快速发展的计算机程序语言，它展示了程序编写的精髓，其简明严谨的结构及简洁的语法编写为它的发展及维护提供了保障。Java 技术应用广泛，从大型复杂的企业级开发到小型移动设备的开发，现在随处可见 Java 活跃的身影。目前，受到来自 Java 社团和 IBM 等全球技术合作伙伴两方面的支持，Java 技术在创新和社会进步上继续发挥强有力的重要作用。并且随着其程序编写难度的降低，更多专业人员将精力放在 Java 语言的编写与框架结构的设计中。

本书大部分章节有代码实例，将一些难以理解的知识融入到实例里，让讲解更加清晰明了，使读者能够轻松理解，快速掌握书中的知识。本书采用一个“聊天室”项目，将所有章节重点技术贯穿起来，每章项目代码层层迭代、不断完善，最终形成一个完整的系统。通过贯穿项目以点连线，多线成面，使读者能够快速理解并掌握各项重点知识，全面提高分析问题、解决问题以及动手编程的能力。

聊天室项目是一个基于 C/S (Client/Server, 客户/服务器) 架构的系统，由 ChatClient 客户端和 ChatServer 服务器端两部分组成。客户端作为系统的一部分，主要是让用户进行设置、登录及聊天的，通过用户的匹配登录，把匹配成功的数据发送到服务器端；服务器端用于接收客户端发送来的数据，将数据保存到数据库中，并为接收保存的数据提供监控和查询功能。项目的分解如下：

- 贯穿项目 (1) 熟悉 Eclipse 的使用，用 Eclipse 搭建项目目录层次。
- 贯穿项目 (2) 通过 String 类可存储要提示的帮助信息机用户名，并可接受用户输入用户名。
- 贯穿项目 (3) 通过 if 判断语句，判断用户是否想要使用默认用户名。若是“否”的话，则自己输入。
- 贯穿项目 (4) 建立一个用户链表的节点类，再建一个用户链表，然后在主类中调用。
- 贯穿项目 (5) 完善代码，把之前项目里面的变量设置为私有变量。
- 贯穿项目 (6) 通过继承 JDialog 类实现帮助信息的窗体化，并通过继承来设置程序图标。
- 贯穿项目 (7) 通过使用接口 ActionListener 来实现各种监听。
- 贯穿项目 (8) 把之前的链表改成泛型模式，并把贯穿项目 (2) 改成集合表示。
- 贯穿项目 (9) 完善之前的贯穿项目，在其中加入异常处理。
- 贯穿项目 (10) 界面框架的设计与实现。
- 贯穿项目 (11) 生成用户信息输入对话框的类，让用户输入自己的用户名和生成连接信息输入的对话框，让用户输入连接服务器的 IP 和端口。
- 贯穿项目 (12) 通过数据库实现记录聊天信息与查看历史聊天信息的功能。

在本书编写过程中，湖南卓景京信息技术有限公司的 CTO（首席技术官）张思奇先生根据自身积累的开发经验，编写了其中的大量案例，在此深表感谢。

由于时间仓促及作者水平有限，本书难免有纰漏和不妥之处，敬请读者提出宝贵的意见与建议，希望能与各位读者共同交流，共同成长。

编者

2019 年 1 月

# 目 录

## 前言

|                       |    |
|-----------------------|----|
| 第 1 章 了解 Java         | 1  |
| 1.1 了解 Java           | 1  |
| 1.1.1 Java 语言的由来与发展   | 1  |
| 1.1.2 Java 语言的特性      | 2  |
| 1.1.3 Java 的运行机制      | 4  |
| 1.1.4 Java 的版本        | 4  |
| 1.2 开发工具与环境配置         | 6  |
| 1.2.1 开发工具            | 6  |
| 1.2.2 环境配置            | 7  |
| 1.2.3 编写简单的 Java 程序   | 8  |
| 1.2.4 编写比较复杂的 Java 程序 | 9  |
| 1.2.5 编写比较完整的 Java 程序 | 10 |
| 1.2.6 Eclipse 开发工具    | 12 |
| 1.3 Java 文档           | 13 |
| 1.3.1 查阅 JDK 文档       | 13 |
| 1.3.2 生成自己程序的文档       | 13 |
| 1.4 Java 开发规范         | 14 |
| 1.4.1 Java 格式规范       | 15 |
| 1.4.2 Java 注释规范       | 16 |
| 1.5 贯穿项目 (1)          | 17 |
| 1.6 本章小结              | 19 |
| 第 2 章 Java 数据处理       | 20 |
| 2.1 变量与函数             | 20 |
| 2.2 数据类型              | 22 |
| 2.2.1 Java 的基本数据类型    | 23 |
| 2.2.2 基本数据类型的封装类      | 27 |
| 2.2.3 复杂数据类型——类和接口    | 28 |
| 2.2.4 数据类型转换          | 30 |
| 2.3 运算符               | 32 |
| 2.3.1 算术运算符           | 32 |
| 2.3.2 位运算符            | 33 |
| 2.3.3 关系运算符           | 35 |
| 2.3.4 逻辑运算符           | 36 |

|                                  |    |
|----------------------------------|----|
| 2.4 数组                           | 38 |
| 2.4.1 一维数组                       | 39 |
| 2.4.2 多维数组                       | 40 |
| 2.5 String 类和 StringBuffer 类     | 43 |
| 2.5.1 String 类                   | 43 |
| 2.5.2 StringBuffer 类             | 49 |
| 2.5.3 String 类和 StringBuffer 类比较 | 51 |
| 2.5.4 Math 类和 Object 类           | 53 |
| 2.6 贯穿项目 (2)                     | 56 |
| 2.7 本章小结                         | 57 |
| 第 3 章 Java 程序控制                  | 58 |
| 3.1 循环控制                         | 58 |
| 3.1.1 while 循环语句                 | 58 |
| 3.1.2 do-while 循环                | 60 |
| 3.1.3 for 循环                     | 61 |
| 3.2 判断控制                         | 63 |
| 3.2.1 if 语句                      | 64 |
| 3.2.2 switch 语句                  | 65 |
| 3.3 跳转控制                         | 67 |
| 3.3.1 break 语句                   | 67 |
| 3.3.2 continue 语句                | 68 |
| 3.3.3 return 语句                  | 68 |
| 3.4 贯穿项目 (3)                     | 69 |
| 3.5 本章小结                         | 70 |
| 第 4 章 Java 类与对象                  | 71 |
| 4.1 理解面向对象                       | 71 |
| 4.1.1 唐僧成功创业的故事——从没钱的老板到百万富翁     | 71 |
| 4.1.2 面向过程到面向对象的思维转换             | 73 |
| 4.2 类                            | 76 |
| 4.2.1 类声明和类体                     | 76 |
| 4.2.2 类声明                        | 77 |
| 4.2.3 类体的构成                      | 77 |

|                              |     |                     |     |
|------------------------------|-----|---------------------|-----|
| 4.3 对象                       | 78  | 6.1.1 继承的基础知识       | 107 |
| 4.3.1 对象的创建                  | 78  | 6.1.2 子类的继承性        | 108 |
| 4.3.2 对象的使用                  | 80  | 6.2 关联关系            | 110 |
| 4.3.3 对象的销毁                  | 81  | 6.3 聚合关系            | 111 |
| 4.4 属性                       | 81  | 6.4 继承下的重写          | 112 |
| 4.4.1 类变量和实例变量               | 81  | 6.4.1 方法重写          | 112 |
| 4.4.2 final 变量               | 82  | 6.4.2 super 关键字     | 113 |
| 4.5 方法                       | 83  | 6.5 多态性             | 115 |
| 4.5.1 方法的声明和方法体              | 83  | 6.6 final 方法        | 116 |
| 4.5.2 构造方法                   | 83  | 6.7 Object 类        | 117 |
| 4.5.3 类方法与实例方法               | 84  | 6.8 贯穿项目 (6)        | 118 |
| 4.5.4 this 关键字               | 85  | 6.9 本章小结            | 123 |
| 4.6 包                        | 86  | 第7章 抽象类与接口          | 124 |
| 4.6.1 创建包                    | 86  | 7.1 抽象类             | 124 |
| 4.6.2 导入包                    | 87  | 7.1.1 抽象类介绍         | 124 |
| 4.7 贯穿项目 (4)                 | 88  | 7.1.2 抽象类的应用        | 125 |
| 4.8 本章小结                     | 91  | 7.2 接口              | 127 |
| 第5章 深入 Java 类                | 92  | 7.2.1 接口的声明与接口体     | 127 |
| 5.1 类的访问限制                   | 92  | 7.2.2 理解接口          | 129 |
| 5.1.1 私有变量和私有方法              | 92  | 7.2.3 接口的使用         | 129 |
| 5.1.2 公有变量和公有方法              | 93  | 7.3 接口应用            | 130 |
| 5.1.3 受保护变量和受保护方法            | 94  | 7.3.1 接口回调          | 130 |
| 5.1.4 友好变量和友好方法              | 94  | 7.3.2 接口作为参数        | 131 |
| 5.2 方法重载                     | 95  | 7.4 贯穿项目 (7)        | 133 |
| 5.3 static 关键字               | 96  | 7.5 本章小结            | 135 |
| 5.4 嵌套类与内部类                  | 97  | 第8章 泛型与集合           | 136 |
| 5.4.1 外部类中定义内部类              | 98  | 8.1 泛型              | 136 |
| 5.4.2 方法中定义内部类               | 100 | 8.2 泛型的应用           | 138 |
| 5.4.3 匿名内部类                  | 101 | 8.2.1 泛型类           | 138 |
| 5.5 JavaBean 组件技术            | 102 | 8.2.2 泛型通配符         | 139 |
| 5.5.1 JavaBean 组件的基本概念       | 102 | 8.2.3 泛型有界类型        | 140 |
| 5.5.2 JavaBean 组件的开发环境       | 102 | 8.2.4 泛型的限制         | 141 |
| 5.5.3 JavaBean 组件的任务         | 103 | 8.3 集合              | 141 |
| 5.5.4 JavaBean 组件的设计目标及其实现方式 | 103 | 8.3.1 集合概述          | 141 |
| 5.6 贯穿项目 (5)                 | 105 | 8.3.2 集合框架          | 142 |
| 5.7 本章小结                     | 106 | 8.3.3 迭代器接口         | 143 |
| 第6章 对象关系研究                   | 107 | 8.4 集合类             | 144 |
| 6.1 继承关系                     | 107 | 8.4.1 Collection 接口 | 144 |
|                              |     | 8.4.2 List 接口       | 144 |

|                        |     |                               |     |
|------------------------|-----|-------------------------------|-----|
| 8.4.3 Set 接口           | 148 | 10.6.1 绘图概述                   | 194 |
| 8.4.4 Map 接口           | 148 | 10.6.2 三种绘图方法                 | 195 |
| 8.4.5 Queue 接口         | 149 | 10.6.3 绘图示例                   | 196 |
| 8.5 集合工具类              | 150 | 10.7 贯穿项目 (10)                | 197 |
| 8.6 贯穿项目 (8)           | 154 | 10.8 本章小结                     | 203 |
| 8.7 本章小结               | 157 | 第 11 章 Swing 程序设计             | 204 |
| 第 9 章 异常处理与垃圾回收        | 158 | 11.1 Swing 简介                 | 204 |
| 9.1 异常处理机制             | 158 | 11.1.1 Swing 类的层次结构           | 205 |
| 9.1.1 try-catch        | 158 | 11.1.2 Swing 程序结构简介           | 206 |
| 9.1.2 finally          | 160 | 11.1.3 JFrame 与 Frame         | 206 |
| 9.1.3 throw 关键字        | 161 | 11.2 常用控件                     | 207 |
| 9.1.4 throws 关键字       | 162 | 11.2.1 按钮控件 (JButton)         | 207 |
| 9.1.5 正确处理异常           | 163 | 11.2.2 标签控件 (JLabel)          | 208 |
| 9.2 垃圾回收               | 168 | 11.2.3 文本域控件 (JTextField)     | 210 |
| 9.2.1 透视 Java 垃圾回收     | 169 | 11.2.4 选择框控件 (JComboBox)      | 211 |
| 9.2.2 finalize() 方法    | 169 | 11.3 对话框                      | 212 |
| 9.3 贯穿项目 (9)           | 172 | 11.3.1 消息对话框                  | 213 |
| 9.4 本章小结               | 174 | 11.3.2 确认对话框                  | 214 |
| 第 10 章 图形界面设计 (AWT)    | 175 | 11.3.3 输入对话框                  | 215 |
| 10.1 图形界面设计简介          | 175 | 11.3.4 对话框的应用                 | 217 |
| 10.2 窗体框架 Frame 类      | 176 | 11.3.5 文件选择对话框 (JFileChooser) | 217 |
| 10.2.1 Frame 类简介       | 176 | 11.3.6 文件选择对话框的使用示例           | 218 |
| 10.2.2 Frame 类的创建及主要方法 | 176 | 11.4 视图与模型机制 (MVC)            | 220 |
| 10.3 控件类               | 177 | 11.4.1 视图与模型机制 (MVC) 简介       | 220 |
| 10.3.1 按钮控件 Button     | 177 | 11.4.2 体会视图与模型机制 (MVC)        | 221 |
| 10.3.2 标签控件 Label      | 179 | 11.5 List 控件及其应用              | 221 |
| 10.3.3 文本域控件 TextField | 180 | 11.5.1 JList 控件简介             | 221 |
| 10.3.4 列表框控件 List      | 181 | 11.5.2 JList 的构造函数和常用方法       | 221 |
| 10.3.5 选择框控件 Choice    | 182 | 11.5.3 JList 控件的创建            | 222 |
| 10.3.6 面板控件 Panel      | 183 | 11.5.4 JList 控件的事件处理及应用       | 225 |
| 10.4 窗体布局              | 185 | 11.6 JTable 控件及应用             | 227 |
| 10.4.1 为什么要使用布局管理      | 185 | 11.6.1 JTable 控件简介            | 227 |
| 10.4.2 三种常用的布局         | 185 | 11.6.2 JTable 的构造函数和常用方法      | 227 |
| 10.5 事件                | 188 | 11.6.3 JTable 控件的创建           | 228 |
| 10.5.1 Java 事件的处理机制    | 188 | 11.6.4 JTable 控件的事件处理及应用      | 229 |
| 10.5.2 按钮事件            | 189 | 11.7 网络编程之套接字                 | 231 |
| 10.5.3 窗体事件            | 192 | 11.7.1 网络编程简介                 | 231 |
| 10.5.4 鼠标事件            | 192 | 11.7.2 TCP/IP 协议              | 231 |
| 10.6 窗体绘制              | 194 | 11.7.3 Socket                 | 232 |

|                                  |     |                               |     |
|----------------------------------|-----|-------------------------------|-----|
| 11.8 贯穿项目 (11) .....             | 232 | 12.3.2 Connection 的常用方法 ..... | 248 |
| 11.9 本章小结 .....                  | 244 | 12.3.3 Connection 的应用 .....   | 248 |
| 第 12 章 JDBC 数据库访问技术 .....        | 245 | 12.4 Statement 接口 .....       | 249 |
| 12.1 JDBC 技术概述 .....             | 245 | 12.4.1 Statement 接口 .....     | 249 |
| 12.1.1 JDBC 的概念 .....            | 245 | 12.4.2 Statement 的常用方法 .....  | 249 |
| 12.1.2 JDBC 的诞生 .....            | 245 | 12.4.3 Statement 的应用 .....    | 250 |
| 12.1.3 JDBC 的任务 .....            | 246 | 12.5 ResultSet 接口 .....       | 251 |
| 12.1.4 JDBC 驱动程序分类 .....         | 246 | 12.5.1 游标 .....               | 251 |
| 12.2 DriverManager 类 .....       | 247 | 12.5.2 ResultSet 接口 .....     | 252 |
| 12.2.1 DriverManager 类简介 .....   | 247 | 12.5.3 ResultSet 的常用方法 .....  | 252 |
| 12.2.2 DriverManager 的常用方法 ..... | 247 | 12.5.4 ResultSet 的应用 .....    | 252 |
| 12.3 Connection 接口 .....         | 247 | 12.6 贯穿项目 (12) .....          | 254 |
| 12.3.1 Connection 接口 .....       | 247 | 12.7 本章小结 .....               | 258 |

# 第 1 章 了解 Java



## 学习目标

本章学习下列知识:

- Java 的历史与发展。
- Java 语言的特性。
- Java 的开发环境及其配置。
- Java 文档。
- Java 开发规范。

使读者具备下述能力:

- 了解 Java 语言, 理解它的优越性。
- 熟悉 Java 开发工具和开发环境配置。
- 编写简单的 Java 程序。
- 为自己的程序生成说明文档。
- 规范化地编程。

## 1.1 了解 Java

### 1.1.1 Java 语言的由来与发展

20 世纪 90 年代中期, SUN 公司的 James Gosling 和其他开发人员致力于开发一个交互式的 TV 项目, Gosling 不满意他当时正在使用的 C++。C++ 是一种面向对象的编程语言, 它是由 AT&T 的贝尔实验室的 Bjarne Stroustrup 于 1979 年在 C 语言的基础上发展而来的, 最初将这种语言称为“带类的 C”, 1983 年改名为 C++。

Gosling 把自己关在办公室, 创建了一种适合其项目的语言, 该语言解决了 C++ 中一些令他失望的问题。虽然 SUN 的交互式 TV 项目最后仍以失败告终, 但出乎人们意料的是, 在此期间开发出来的新语言却适用于此时逐渐流行的互联网。这种语言就是 Java 的早期版本, 在 1992 年秋天问世的时候称为 Oak 语言, 于 1995 年正式更名为 Java。

1995 年秋天, SUN 以自由开发包的方式发布了 Java, 该开发包可从 SUN 公司的官方网站下载。虽然与当时的 C++ (以及当今的 Java) 相比, 该语言的大多数特性过于初级, 但被称作小程序 (Applet) 的 Java 程序可作为 Web 页的一部分运行在 Netscape Navigator 浏览器中。这种功能是第一种用于 Web 的交互式编程技术, 给这种新语言提供了极大的舆论攻势, 在短短的 6 个月内便吸引了数以十万计的开发人员。

在人们对 Java Web 编程技术的好奇过后, 该语言的整体优势逐渐清晰, 现在程序员们仍在继续使用它。调查表明, 当前职业 Java 程序员人数已经超过了 C++ 程序员。

Java 最早被正式应用于 Applet 小程序，它主要被嵌入到网页中运行，实现客户端漂亮完美的动态效果。目前很多电子地图、网页图表等都在采用 Applet 技术。我们将在后续章节中学习和应用 Applet 小程序。

随着 Java 技术本身不断的完善，Java 技术的应用几乎遍及到整个 Internet，其应用领域已经涉及到电子商务、电子政务、远程教学、远程医疗、科学研究、智能卡、遥控设备、移动通信以及最近风靡的手机游戏等。可以说，Java 技术已经深入到人们的日常生活了。

### 1.1.2 Java 语言的特性

Java 是一种面向对象的、独立于平台的安全语言，它比 C 和 C++ 更容易学习，且比 C 和 C++ 更能避免被误用。

面向对象编程（Object Oriented Programming, OOP）是一种软件开发方法，它将程序视为一组协同工作的对象。对象是使用被称作类的模板创建的，它们由数据和使用数据所需的语句组成。Java 是完全面向对象的编程语言，在后面的内容中，当你创建第一个类并使用它来生成对象时将会深刻地明白这一点。

独立于平台指的是无需修改程序便能够使其运行在不同的计算环境中。Java 程序被编译成一种名为字节码格式的“class”文件，字节码文件可以在任何带有 Java 解释器的操作系统、软件或设备上运行。可以在 Windows 环境下创建 Java 程序，然后在 Linux Web 服务器上使用 OS X 的 Applet Mac 和 Palm 个人数字助理环境下运行它。只要运行平台安装了 Java 解释器，便可以运行字节码程序。

Java 作为一种跨平台的编程语言，具有丰富的特性。

#### 1. 简单性

Java 最初的设计是源于对独立平台的需要，是为了对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java 语言的简单性主要体现在以下三个方面：

- Java 的风格类似于 C++，从某种意义上讲，Java 语言是 C 及 C++ 语言的一个变种，因此，学习过 C++ 的程序员可以很快就掌握 Java 编程技术。
- Java 摒弃了 C++ 中容易引发程序错误的地方，如指针和内存管理等。
- Java 提供了丰富的类库。

#### 2. 安全性

设计一门计算机语言时，安全性是权衡其市场占有量的一个很重要的标准。一方面，由于 Java 摒弃了 C++ 的指针和内存管理，所以避免了非法内存操作；另一方面，当 Java 用于网络编程时，Java 语言功能和浏览器本身提供的功能结合起来，使得它更为安全。

#### 3. 可移植性

可移植性是指不用修改程序代码就可在不同的软、硬件平台上运行。由于 Internet 将许多不同类型的计算机和操作系统连接在一起，要使连接到 Internet 上的各种平台都能正常地访问 Internet 资源，就需要生成可移植执行的代码。

可移植性一直是 Java 程序员的精神指标，也是 Java 之所以能够受到程序员喜爱的原因之一。Java 依靠 Java 虚拟机（Java Virtual Machine, JVM）实现了它的可移植性。大多数编译器产生的目标代码只能运行在一类 CPU 上（如 Intel 的×86 系列），即使那些能支持多种 CPU 的编译器也不能同时产生适合多种 CPU 的目标代码。如果你需要在三种 CPU（如×86、SPARC

和 MIPS) 上运行同一程序, 就必须编译三次。但 Java 编译器就不同了。Java 编译器产生的目标代码 J-Code 是针对一种并不存在的 CPU-Java 虚拟机, 而不是某一实际的 CPU。Java 虚拟机掩盖不同 CPU 之间的差别, 使 J-Code 能运行于任何具有 Java 虚拟机的机器上。

#### 4. 面向对象性

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的, 它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码继承及重用。单从面向对象的特性来看, Java 类似于 Small Talk, 但其他特性、尤其是其适用于分布式计算环境的特性远远超越了 Small Talk。

#### 5. 健壮性

Java 致力于检查程序在编译和运行时的错误。类型检查可检测出许多开发初期出现的错误。Java 程序操纵内存减少了内存出错的可能性。数组越界会对内存中其他的数据产生覆盖, 造成程序崩溃, Java 实现了真数组, 避免了数据覆盖的可能, 如果数组越界将会抛出异常 Array Index Out Of Bounds Exception, 这样提高了程序的健壮性。Java 提供 Null 指针、数组边界、异常检测, 以及对字节代码的校验等, 这些功能大大缩短了 Java 应用程序的开发周期。

#### 6. 多线程性

多线程功能使得在一个程序里可同时执行多个小任务。线程有时也称小进程, 是一个大进程里分出来的小的独立进程。因为 Java 实现了多线程技术, 所以比 C 和 C++ 更健壮。多线程带来的更大的好处是具有更好的交互性和实时控制性。当然实时控制性还取决于系统本身 (UNIX, Windows, Macintosh 等), 在开发难易程度和性能上都比单线程更好。任何用过浏览器的人都觉得等待一幅图片的显示是一件烦恼的事情。在 Java 中, 你可用一个单线程来显示一幅图片, 同时你可以访问 HTML 里的其他信息而不必等待图片的下载过程。

#### 7. 体系结构中立

为了使 Java 作为网络的一个整体, Java 将它的程序编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。目前 Java 程序能运行的操作系统有 Solaris、Windows、Mac OS X 和 Linux 等。“\*.Java” 源程序被编译成一种高层次的与机器无关的字节码 (byte-code) 格式语言, 这种语言被设计在虚拟机上运行, 由与机器相关的运行调试器实现执行。

#### 8. 解释执行与高性能

Java 解释器 (运行系统) 能直接运行目标代码指令。链接程序通常比编译程序所需要资源少, 所以程序员可以在创建源程序时花费更多的时间。

如果解释器速度不慢, Java 可以在运行时直接将目标代码翻译成机器指令。SUN 用直接解释器一秒钟内可以调用 3000,000 个过程。翻译目标代码的速度与 C 和 C++ 没什么区别。

Java 语言的字节码经过仔细设计, 很容易使用即时编译技术 (Just In Time, JIT) 将字节码直接转换成高性能的本机代码。

#### 9. 分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 的子库。因此, Java 应用程序可凭借 URL 打开并访问网络中的对象, 其访问方式与访问本地对象几乎完全相同。为分布式环境尤其是 Internet 提供动态内容无疑是一项非常宏伟的任务, 但 Java 的语法特性却使我们很容易地完成这项任务。

## 10. 动态性

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类。Java 编译器不是将对实例变量和成员方法的引用编译为数值引用，而是将符号引用信息在字节码中保存下来传递给解释器，再由解释器完成动态链接类后，将符号引用信息转换为数值偏移量。这样，一个在存储器生成的对象不在编译过程中决定，而是延迟到运行时由解释器确定。这样对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时，这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次，随后代码便可以全速执行。在运行时确定引用的好处是，可以使用已被更新的类，而不必担心影响原有的代码。如果程序连接了网络中另一系统中的某一类，该类的所有者也可以自由地对该类进行更新，而不会使任何引用该类的程序崩溃。

### 1.1.3 Java 的运行机制

Java 程序的运行机制包括编写、编译、运行三个步骤。编写是指在 Java 开发环境中进行程序代码的开发，最终形成后缀为“`.java`”的 Java 源文件；编译是指使用 Java 编译器对源文件进行错误排查的过程，编译后将生成后缀为“`.class`”的字节码文件；运行是指使用 Java 解释器将字节码文件翻译成机器代码，执行并显示结果。这一过程如图 1-1 所示。

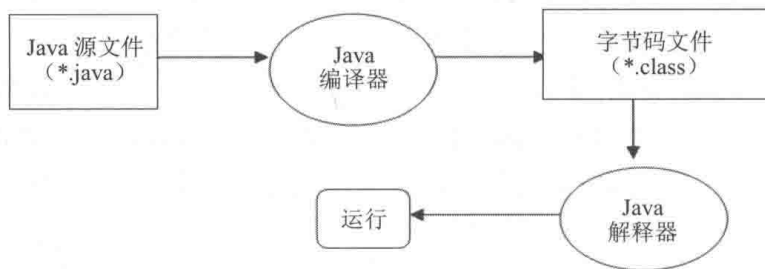


图 1-1 Java 程序运行流程图

字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码，它是一种二进制文件，是 Java 源文件经 Java 编译器编译后生成的目标代码文件。它必须由专用的 Java 解释器来解释执行。Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码，它运行在被称为 Java 虚拟机（JVM）的软件平台之上。

Java 虚拟机是运行 Java 程序的软件环境，在运行 Java 程序时，首先会启动 JVM，然后由它来负责解释执行 Java 的字节码，利用 JVM 可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来，只要在不同的计算机上安装了针对具体平台的 JVM，Java 程序就可以运行，而不用考虑当前具体的硬件平台及操作系统环境，也不用考虑字节码文件是在何种平台上生成的。JVM 是 Java 平台无关性的基础，Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。JVM 跨平台原理图如图 1-2 所示。

### 1.1.4 Java 的版本

Java 自发布 1.0 版本以来，便以飞快的速度向前发展着。Java2 标志着“Java 新时代”的开始，SUN 公司将 Java 产品重新组装为 J2SE（Java2 平台标准）。Java 发展过程中的几个阶段见表 1-1。

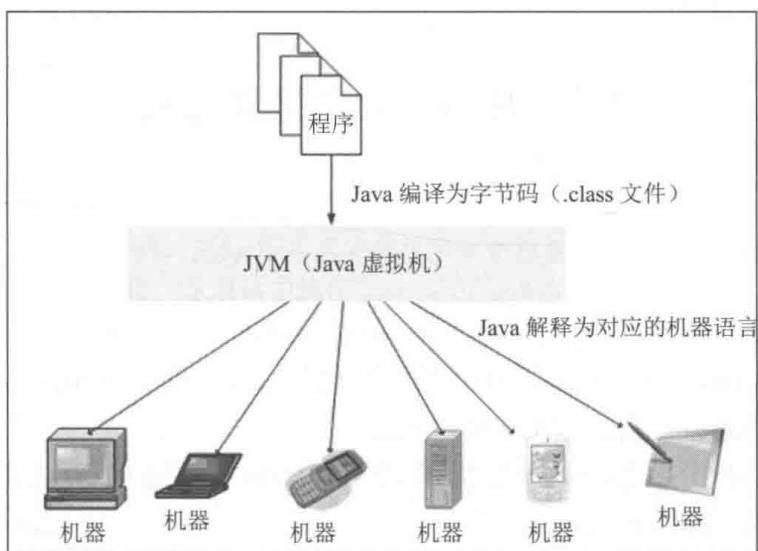


图 1-2 JVM 跨平台原理图

表 1-1 Java 的发展史

| 日期              | 版本号                    | 说明                                                                                                         |
|-----------------|------------------------|------------------------------------------------------------------------------------------------------------|
| 1995 年 5 月 23 日 | 无                      | Java 语言诞生                                                                                                  |
| 1996 年 1 月      | JDK1.0                 | JDK 1.0 诞生，还不能进行真正的应用开发                                                                                    |
| 1998 年 12 月 8 日 | JDK1.2                 | 企业平台 J2EE 发布，里程碑式的产品，性能有所提高，具有完整的 API                                                                      |
| 1999 年 6 月      | Java 三个版本              | 标准版 (J2SE)，企业版 (J2EE)，微型版 (J2ME)                                                                           |
| 2000 年 5 月 8 日  | JDK1.3                 | JDK1.3 发布，对 JDK1.2 进行改进，扩展标准类库                                                                             |
| 2000 年 5 月 29 日 | JDK1.4                 | JDK1.4 正式发布，提高了系统性能，修正一些 Bug                                                                               |
| 2001 年 9 月 24 日 | J2SE1.3                | J2SE 1.3 正式发布                                                                                              |
| 2002 年 2 月 26 日 | J2SE1.4                | 计算能力有了大幅提升                                                                                                 |
| 2004 年 9 月 30 日 | J2SE1.5<br>Java SE 5.0 | Java 语言发展史上的重要里程碑，从该版本开始，增加了泛型类、for-each 循环、可变元参数，自动打包、枚举、静态导入和元数据等技术。为了表示该版本的重要性，J2SE 1.5 更名为 Java SE 5.0 |
| 2005 年 6 月      | Java SE 6.0            | 发布 Java SE 6.0，此时 Java 的各种版本已更名，取消数字“2”分别更名为：Java EE、Java SE、Java ME                                       |
| 2006 年 12 月     | JRE 6.0                | SUN 公司发布 JRE 6.0                                                                                           |
| 2009 年 4 月 20 日 | 收购                     | 甲骨文 74 亿美元收购 SUN，获得 Java 版权                                                                                |
| 2011 年 7 月 28 日 | Java SE 7.0            | 甲骨文发布 Java SE 7.0 正式版                                                                                      |
| 2014 年 3 月 18 日 | Java SE 8.0            | 又一里程碑，甲骨文发布 Java SE 8.0，增加 Lambda、Default Method 等新特性                                                      |
| 2017 年 9 月 21 日 | Java SE 9.0            | 增加新特性：Jigsaw 项目，模块化源码，轻量级 JSON API，智能 Java 编译，代码分段缓存等。                                                     |
| 2018 年 3 月 26 日 | Java SE 10.0           | 局部变量类型推断，并行完整 GC，应用程序类数据共享等改进                                                                              |

## 1.2 开发工具与环境配置

### 1.2.1 开发工具

Java 程序的开发工具 JDK 的安装程序可以登录 Oracle 官方网站下载最新版本。Oracle 官方网站: <http://www.oracle.com>, JDK-1.10.0 的下载地址: <http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html>。

这里使用 JDK-1.10.0 版本。安装成功后, JDK 安装的默认路径是“C:\Program Files\Java\jdk-1.10.0”, 如图 1-3 所示。

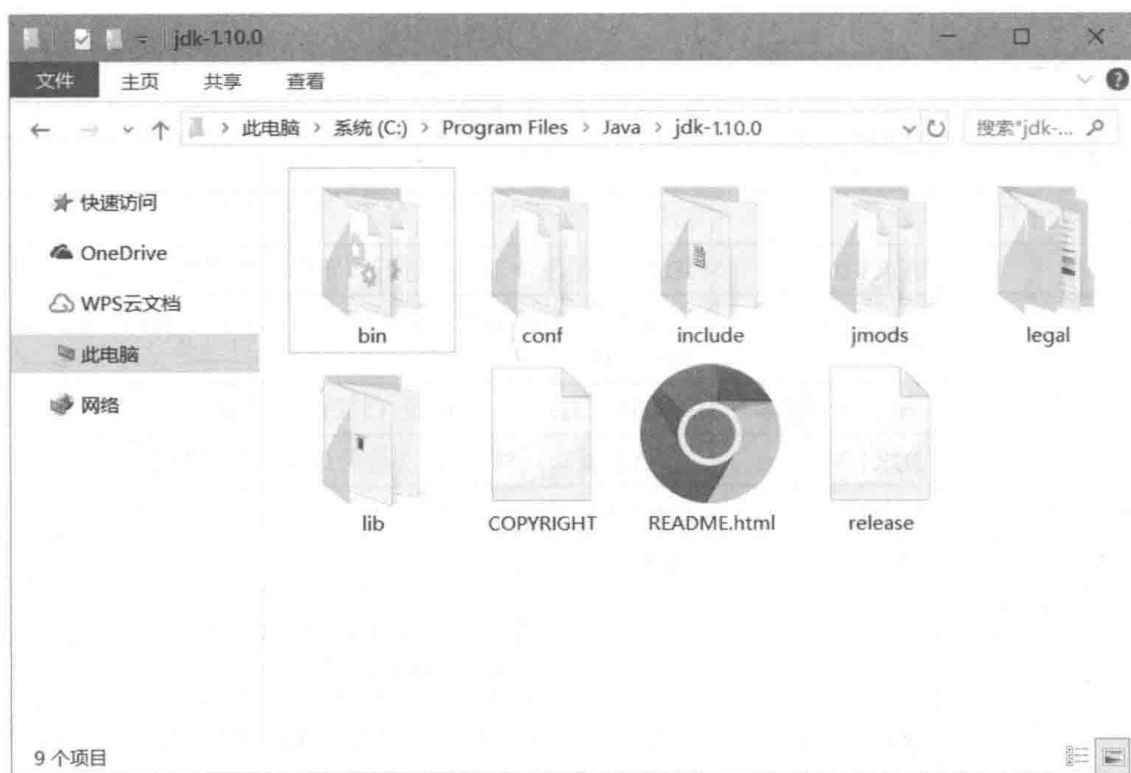


图 1-3 JDK-1.10.0 安装目录

在 JDK-1.10.0 安装目录下, 有些文件夹和文件是需要了解的, 具体说明见表 1-2。

表 1-2 JDK-1.10.0 安装目录下的文件夹说明

| 文件夹 | 说明                                                    |
|-----|-------------------------------------------------------|
| bin | 提供了 JDK 的工具程序, 包括 javac、java、javadoc、appletviewer 等程序 |
| lib | 提供了工具程序使用的 Java 工具类                                   |

在 bin 文件夹中, 有些程序也是需要熟练掌握其应用的, 具体见表 1-3。

表 1-3 bin 文件夹下的程序及其使用说明

| 程序               | 使用说明                                                                                                             |
|------------------|------------------------------------------------------------------------------------------------------------------|
| javac.exe        | Java 程序编译器。它读取 Java 源代码，并将其编译成字节码文件 (*.class)                                                                    |
| java.exe         | Java 程序执行器。它用来执行编译后的 “*.class” 文件                                                                                |
| jdb.exe          | Java 程序调试器。它为 Java 程序提供了一个命令行调试环境，既可在本地，也可在与远程的解释器的一次对话中执行                                                       |
| javadoc.exe      | Java 参考文档生成器。它从 Java 源程序中提取信息生成 HTML 格式的软件参考文档，这些 HTML 文件描述了 Java 类文件的类、变量、成员方法，所有 Java 类库的 API-HTML 文件都可以由此程序创建 |
| appletviewer.exe | Applet 程序观看器。用它来在浏览器中观看 Applet 小程序。它的简单用法是 appletviewer XXX.html，其中 XXX.html 是嵌入了 Applet 的 HTML 文档，扩展名不能省略       |
| jar.exe          | Java 类包程序生成器。它可以用来将 Java 应用程序压缩成 “.jar” 文件                                                                       |

## 1.2.2 环境配置

关于 JDK-10.0.1 的安装和配置，需要重点说明的是 Path 和 Classpath 这两个配置。

在配置 Path 和 CLASSPATH 前，新建一个系统变量 JAVA\_HOME，里面存储 JDK 位置，目的是为了以后 JDK 的升级。图 1-4 和图 1-5 为 Win10 打开环境变量界面进行参数配置的一般方法。



图 1-4 打开此电脑属性



图 1-5 打开环境变量

Path 配置主要指运行的程序所在的目录，在环境变量里配置该参数信息，如图 1-6 所示。CLASSPATH 配置主要解决 Java 程序编译或运行时找不到类的问题，也就是指定 Java 程序需要的类路径，如图 1-6 所示。

| 变量                   | 值                                                           |
|----------------------|-------------------------------------------------------------|
| CLASSPATH            | .;%JAVA_HOME%\bin;%JAVA_HOME%\lib\tools.jar;G:\sqljdbc4\... |
| ComSpec              | C:\WINDOWS\system32\cmd.exe                                 |
| GTK_BASEPATH         | C:\Program Files (x86)\GtkSharp\2.12\                       |
| JAVA_HOME            | C:\Program Files\Java\jdk-10.0.1                            |
| NUMBER_OF_PROCESSORS | 8                                                           |
| OS                   | Windows_NT                                                  |
| Path                 | %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;C:\Program Files (x...  |
| PATHEXT              | COMPILE BAT CMD EXE JAR JPS JS JSE MSC MFC MFC...           |

图 1-6 配置 CLASSPATH 和 Path

下面对 “.;%JAVA\_HOME%\bin;%JAVA\_HOME%\lib\tools.jar” 进行说明。

- “.” 代表当前目录，意味着编译时程序会从当前目录下寻找需要的类。
- “%JAVA\_HOME%\lib” 代表指定的目录，意味着编译时程序会从指定目录下寻找需要的类。（%JAVA\_HOME%表示 C:\Program Files\Java\jdk-10.0.1）。
- “%JAVA\_HOME%\lib\tools.jar” 代表指定的类库包，意味着编译时程序会解压缩 tools.jar 包，并从中寻找需要的类。

### 1.2.3 编写简单的 Java 程序

有了上面的基础知识之后，就可以编写一个简单的 Java 程序了。

#### 【例 1-1】

```
/**
 * 这是我编写的第一个 Java 程序
 */
public class FirstProgram {
    /**
     * main()方法是程序的入口
     * 注意这个方法的写法，注意大小写
     */
    public static void main(String[] args) {
        /* 让程序在命令窗口打印一句话*/
        System.out.println("我是长沙卓京的一名学子！");
    }
}
```

上面的这段程序可以用记事本编写，然后保存为 FirstProgram.java 文件。

实践提示：FirstProgram 是程序的一个 public 类，所以文件名必须和类名相同，并且大小写也要完全一致。接下来编译该程序，操作指令如图 1-7 所示。



图 1-7 编译程序

然后查看该目录下是否有 FirstProgram.class 文件生成，如果有，那么这就是编译好的类文件。运行该类文件并查看运行结果，具体操作如图 1-8 所示。