

```

static void quicksort(int n[], int left, int right) {
    int dp;
    if (left < right) {
        dp = partition(n, left, right);
        quicksort(n, left, dp - 1);
        quicksort(n, dp + 1, right);
    }
}

static int partition(int n[], int left, int right) {
    int pivot = n[left];
    while (left < right) {
        while (left < right && n[right] >= pivot)
            right--;
        if (left < right)
            n[left++] = n[right];
        while (left < right && n[left] <= pivot)
            left++;
        if (left < right)
            n[right--] = n[left];
    }
}

```



李瑞生 何珍祥 主编

# Java SE

# 程序设计

Java SE CHENGXU SHEJI

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

# Java SE 程序设计

李瑞生 何珍祥 主 编

张 静 副主编

## 内 容 简 介

本书从实用的角度出发,全面介绍 Java SE 程序设计知识,通过大量案例引导读者系统掌握 Java SE 程序设计技能。全书共有 15 章内容:第 1 章和第 2 章,主要介绍 Java 的发展、特点、Java 程序运行机制与虚拟机、Java SE 程序开发环境、Java 语法等内容;第 3~5 章,主要介绍 Java 面向对象编程基础、Java 继承与多态、Java 抽象类与接口等内容;第 6~8 章,主要介绍 Java 数组与枚举、Java 常用工具类、异常处理机制等内容;第 9~14 章,主要介绍 Java 组件与事件编程、Java 数据库编程、Java 输入输出流、Java 多线程编程、Java 网络编程、Java 泛型与集合类等 Java SE 高级程序设计内容;第 15 章主要介绍 Java Applet 编程基础知识。

本书编者在总结长期 Java 程序设计教学经验的基础上,不断凝练内容和案例,内容安排上环环相扣,案例丰富实用,所有例题代码均在 JDK+Eclipse 环境下调试运行成功,同时给出了详尽的注释。

本书适合作为 Java 软件开发从业人员的入门培训教材和参考书,也可作为普通高等院校计算类、信息类本科专业 Java SE 程序设计教材。

### 图书在版编目(CIP)数据

Java SE 程序设计 / 李瑞生,何珍祥主编. —北京 :  
中国铁道出版社, 2016.12  
ISBN 978-7-113-22015-0

I. ①J… II. ①李… ②何… III. ①JAVA 语言—程序  
设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 188332 号

书 名: Java SE 程序设计  
作 者: 李瑞生 何珍祥 主编

策 划: 潘晨曦 孙晨光  
责任编辑: 秦绪好 鲍 闻  
封面设计: 刘 颖  
封面制作: 白 雪  
责任校对: 汤淑梅  
责任印制: 郭向伟

读者热线: (010) 63550836

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 虎彩印艺股份有限公司

版 次: 2016 年 12 月第 1 版 2016 年 12 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 22.5 字数: 558 千

书 号: ISBN 978-7-113-22015-0

定 价: 52.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659

Java 不仅是一种程序设计语言,而且是一种软件开发的标准、规范、平台和技术的集合。Java 包括 Java SE、Java EE、Java ME 三种平台,各个平台下都包含相关的 API、服务组件和基础服务等,允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序;支持 Web 应用程序及面向服务体系结构的企业级应用;支持移动应用开发和用于家电、智能设备、手持设备、通信设备等嵌入式应用程序的开发等。Java 以其纯面向对象、跨平台、高性能、分布性和可移植性等特点,已成为目前网络编程、手机应用开发等领域的主流编程技术。随着网络向着云计算、物联网的方向发展,Java 技术具有更加广阔的应用市场和应用前景。

近年来,社会对 Java 技术人才的需求持续增长,因此,Java 程序设计系列课程已成为各个普通高等院校计算机类、信息类专业学生的程序设计主干课程之一,Java 程序设计能力成为计算机类、信息类专业软件技术人才的基本能力要求。

Java SE 程序设计是学习 Java 编程技术的起点和基础。编者在总结多年教学经验的基础上,按照学生的认知规律和学习特点精心组织了本书的内容,并通过大量的实例由浅入深地讲解了 Java SE 程序设计的相关知识和编程方法。全书共分为 15 章:第 1 章 Java 简介与 Java SE 程序开发环境;第 2 章 Java 语法基础;第 3 章 Java 面向对象程序设计基础;第 4 章 Java 继承与多态;第 5 章 Java 抽象类与接口;第 6 章 Java 数组与枚举;第 7 章 Java 常用工具类;第 8 章 异常处理机制;第 9 章 Java 组件与事件编程;第 10 章 Java 数据库编程;第 11 章 Java 输入、输出流;第 12 章 Java 多线程编程;第 13 章 Java 网络编程;第 14 章 Java 泛型与集合类;第 15 章 Java Applet 编程简介。各章节的例题都在 JDK+Eclipse 环境下调试成功。各章节的安排和内容取舍注重可读性和实用性,理论知识点以实用、够用为原则,减少冗余繁复;每章节的例题和习题都经过精心考虑,做到既能帮助理解编程理论,又具有实战性、启发性。

本书章节层次合理清晰、内容充实、实例丰富、易学易用。本书既可作为普通高等院校计算机类、信息类本科专业 Java SE 程序设计的教材,也可作为 Java 软件开发从业人员的入门培训教材。

本书由甘肃政法学院李瑞生、何珍祥任主编,张静任副主编,由李瑞生对全书进行统稿,何珍祥负责本书的校对工作。本书中第 1、3、4、5、9、10、12、13、14 章由李瑞生编写,第 2 章由何珍祥编写,第 6、7、8、11、15 章由张静编写。

在本书的编写过程中，得到了甘肃政法学院信息工程学院院长安德智教授、陆军副教授、武光利副教授的指导和帮助。同时，得到了中国铁道出版社编辑部老师们的大力支持和帮助。在此，对以上各位老师的指导和帮助表示真挚的感谢。

由于编者水平有限，缺点和欠妥之处难免，恳请读者指正。

编者

2016年9月

## 第 1 章 Java 简介与 Java SE 程序

## 开发环境 ..... 1

## 1.1 Java 的发展 ..... 1

## 1.2 Java 的 3 种平台 ..... 2

## 1.3 Java 的特点 ..... 2

1.4 Java 程序的运行机制与 Java  
虚拟机 ..... 3

## 1.4.1 Java 程序的运行机制 ..... 3

## 1.4.2 Java 虚拟机 ..... 3

## 1.5 Java 程序分类 ..... 4

## 1.6 JDK 的安装与配置 ..... 4

## 1.6.1 JDK 的安装 ..... 4

## 1.6.2 JDK 环境变量的配置 ..... 5

## 1.7 JDK 环境下开发 Java 程序 ..... 6

## 1.7.1 Java Application 的开发 ..... 6

## 1.7.2 Java Applet 的开发 ..... 10

## 1.8 Eclipse 环境下开发 Java 程序 ..... 12

1.8.1 在 Eclipse 环境下开发 Java  
Application ..... 121.8.2 在 Eclipse 环境下开发 Java  
Applet ..... 14

## 1.9 其他几个问题 ..... 15

## 小结 ..... 16

## 习题 1 ..... 16

## 第 2 章 Java 语法基础 ..... 18

2.1 Java Application 程序的基本输入、  
输出方法 ..... 182.1.1 Java 向显示器 (控制台)  
输出数据 ..... 18

## 2.1.2 Java 从键盘读入数据 ..... 19

## 2.1.3 用 Scanner 灵活读入数据 ... 20

## 2.2 Java 注释 ..... 20

## 2.3 标识符与关键字 ..... 21

## 2.3.1 标识符 ..... 21

## 2.3.2 关键字 ..... 22

## 2.4 Java 数据类型 ..... 22

## 2.4.1 基本数据类型 ..... 22

## 2.4.2 常量与变量 ..... 23

## 2.4.3 数据类型转换 ..... 26

## 2.4.4 引用数据类型 ..... 26

## 2.5 运算符与表达式 ..... 26

## 2.5.1 算术运算符及其表达式 ... 27

## 2.5.2 关系运算符 ..... 28

## 2.5.3 逻辑运算符 ..... 28

## 2.5.4 位运算符 ..... 28

## 2.5.5 赋值运算符及其表达式 ... 29

2.5.6 复合赋值运算符及其  
表达式 ..... 29

## 2.5.7 条件运算符及其表达式 ... 30

2.5.8 instanceof 运算符及其  
表达式 ..... 30

## 2.6 Java 语句 ..... 30

## 2.7 Java 程序流程控制 ..... 31

## 2.7.1 顺序结构 ..... 31

## 2.7.2 选择结构 ..... 31

## 2.7.3 循环结构 ..... 36

## 2.7.4 流程跳转语句 ..... 37

## 小结 ..... 39

## 习题 2 ..... 39

## 第 3 章 Java 面向对象程序设计基础 ... 41

## 3.1 面向对象概述 ..... 41

## 3.1.1 类与对象的概念 ..... 41

## 3.1.2 面向对象的主要特性 ..... 42

## 3.2 Java 类与对象 ..... 42

|                        |                             |    |                                    |     |
|------------------------|-----------------------------|----|------------------------------------|-----|
| 3.2.1                  | Java 类的定义 .....             | 42 | 习题 4 .....                         | 85  |
| 3.2.2                  | 构造方法与对象的创建 ....             | 43 | 第 5 章 Java 抽象类与接口 .....            | 87  |
| 3.2.3                  | 对象声明及创建的内存<br>模型 .....      | 45 | 5.1 抽象方法与抽象类 .....                 | 87  |
| 3.2.4                  | 对象间的赋值 .....                | 46 | 5.2 Java 接口 .....                  | 89  |
| 3.3                    | 方法重载 .....                  | 47 | 5.3 Java 接口回调 .....                | 91  |
| 3.4                    | this 关键字 .....              | 48 | 5.4 内部类 .....                      | 92  |
| 3.5                    | static 与静态成员 .....          | 50 | 5.5 匿名类与匿名对象 .....                 | 95  |
| 3.5.1                  | 静态变量 .....                  | 50 | 5.6 面向抽象（接口）编程 .....               | 99  |
| 3.5.2                  | 静态方法 .....                  | 51 | 5.7 接口的一个应用——工厂模式 ...              | 101 |
| 3.5.3                  | 静态代码块 .....                 | 52 | 5.8 Java 内置注解简介 .....              | 103 |
| 3.6                    | 参数传递 .....                  | 54 | 小结 .....                           | 106 |
| 3.7                    | 类的关联与依赖关系 .....             | 56 | 习题 5 .....                         | 106 |
| 3.8                    | package 与 import 关键字 .....  | 57 | 第 6 章 Java 数组与枚举 .....             | 108 |
| 3.8.1                  | Java 中的包 .....              | 57 | 6.1 一维数组 .....                     | 108 |
| 3.8.2                  | 用 package 关键字<br>自定义包 ..... | 58 | 6.2 数组间赋值 .....                    | 110 |
| 3.8.3                  | 用 import 关键字导入包 ....        | 58 | 6.3 Arrays 类中处理数组的系统<br>方法 .....   | 111 |
| 3.9                    | JavaBean .....              | 60 | 6.4 Java 二维数组的定义及使用 .....          | 112 |
| 3.10                   | jar 命令的用法 .....             | 62 | 6.5 Java 对象数组 .....                | 115 |
| 3.10.1                 | 将应用程序打包为 jar<br>文件 .....    | 63 | 6.6 Java 枚举类型 .....                | 117 |
| 3.10.2                 | 生成 jar 文件扩展类库 ....          | 64 | 6.6.1 用 enum 定义枚举类型 ....           | 117 |
| 3.10.3                 | Eclipse 环境下的文件<br>打包 .....  | 65 | 6.6.2 枚举类型的构造方法 .....              | 118 |
| 小结 .....               |                             | 66 | 6.6.3 在 switch 结构中使用枚举<br>类型 ..... | 118 |
| 习题 3 .....             |                             | 66 | 小结 .....                           | 119 |
| 第 4 章 Java 继承与多态 ..... |                             | 69 | 习题 6 .....                         | 119 |
| 4.1                    | Java 继承 .....               | 69 | 第 7 章 Java 常用工具类 .....             | 121 |
| 4.2                    | 权限修饰符 .....                 | 70 | 7.1 Object 类及其常用方法 .....           | 121 |
| 4.3                    | 子类继承性 .....                 | 72 | 7.2 基本数据类型包装类 .....                | 125 |
| 4.4                    | 变量覆盖与方法重写 .....             | 74 | 7.3 String 类 .....                 | 126 |
| 4.5                    | super 的用法 .....             | 76 | 7.4 StringBuffer 类 .....           | 131 |
| 4.6                    | 对象的上下转型 .....               | 80 | 7.5 正则表达式 .....                    | 132 |
| 4.7                    | instanceof 关键字 .....        | 82 | 7.5.1 正则表达式简介 .....                | 132 |
| 4.8                    | 多态 .....                    | 83 | 7.5.2 Pattern 与 Matcher 类 .....    | 133 |
| 4.9                    | final 修饰符 .....             | 84 | 7.6 字符串解析方法 .....                  | 135 |
| 小结 .....               |                             | 85 | 7.7 日期时间类 .....                    | 138 |
|                        |                             |    | 7.8 Math 类 .....                   | 139 |

|                                      |            |                                                    |            |
|--------------------------------------|------------|----------------------------------------------------|------------|
| 7.9 BigInteger 类 .....               | 141        | 9.7.1 JButton .....                                | 184        |
| 7.10 Random 类 .....                  | 141        | 9.7.2 JLabel .....                                 | 184        |
| 7.11 其他常用类 .....                     | 142        | 9.7.3 JTextField .....                             | 185        |
| 7.12 Class 类与 Java 的反射机制<br>简介 ..... | 144        | 9.7.4 JTextArea .....                              | 185        |
| 小结 .....                             | 150        | 9.7.5 JRadioButton .....                           | 185        |
| 习题 7 .....                           | 150        | 9.7.6 JCheckBox .....                              | 185        |
| <b>第 8 章 异常处理机制 .....</b>            | <b>152</b> | 9.7.7 JComboBox .....                              | 186        |
| 8.1 Java 异常处理概述 .....                | 152        | 9.7.8 JList .....                                  | 186        |
| 8.2 Java 异常类 .....                   | 152        | 9.7.9 JMenuBar、JMenu 与<br>JMenuItem .....          | 189        |
| 8.3 Java 异常处理语法 .....                | 153        | 9.7.10 JPopupMenu .....                            | 190        |
| 8.4 强制检查异常和非强制检查异常 .....             | 156        | 9.7.11 JToolBar .....                              | 190        |
| 8.5 用户自定义异常 .....                    | 159        | 9.7.12 JTable .....                                | 193        |
| 小结 .....                             | 161        | 9.7.13 JTabbedPane .....                           | 196        |
| 习题 8 .....                           | 161        | 9.7.14 JTree .....                                 | 198        |
| <b>第 9 章 Java 组件与事件编程 .....</b>      | <b>162</b> | 9.7.15 JDialog、JOptionPane 与<br>JFileChooser ..... | 200        |
| 9.1 Java AWT 与 Swing 简介 .....        | 162        | 9.7.16 JSlider .....                               | 203        |
| 9.2 容器和组件 .....                      | 162        | 9.7.17 JProgressBar .....                          | 205        |
| 9.3 Java 布局管理器 .....                 | 164        | 9.7.18 Timer .....                                 | 205        |
| 9.3.1 FlowLayout .....               | 165        | 9.7.19 键盘事件示例 .....                                | 207        |
| 9.3.2 BorderLayout .....             | 166        | 9.8 其他几个应用 .....                                   | 209        |
| 9.3.3 GridLayout .....               | 166        | 9.8.1 更换窗体标题栏图标 .....                              | 209        |
| 9.3.4 CardLayout .....               | 167        | 9.8.2 让窗体在屏幕上<br>居中显示 .....                        | 209        |
| 9.3.5 GridBagLayout .....            | 169        | 9.8.3 将窗体显示为<br>任务栏图标 .....                        | 210        |
| 9.3.6 BoxLayout .....                | 171        | 9.9 字体与颜色 .....                                    | 211        |
| 9.3.7 空布局 .....                      | 172        | 9.9.1 Font 类 .....                                 | 211        |
| 9.4 Java Swing 常用的中间容器 .....         | 173        | 9.9.2 Color 类 .....                                | 211        |
| 9.5 Java 事件编程机制 .....                | 175        | 9.10 GUI 图形绘制 .....                                | 214        |
| 9.6 事件监听器对象的几种实现 .....               | 180        | 9.10.1 Graphics 类 .....                            | 214        |
| 9.6.1 窗体类自身实现相应<br>事件监听器接口的方式 .....  | 180        | 9.10.2 Canvas 类 .....                              | 215        |
| 9.6.2 自定义外部类实现相应<br>事件监听接口的方式 .....  | 181        | 小结 .....                                           | 218        |
| 9.6.3 自定义外部类继承相应的<br>事件适配器类的方式 ..... | 182        | 习题 9 .....                                         | 218        |
| 9.6.4 匿名类实现事件监听器 .....               | 183        | <b>第 10 章 Java 数据库编程 .....</b>                     | <b>220</b> |
| 9.7 Swing 的常用组件及其事件<br>编程 .....      | 184        | 10.1 JDBC 简介 .....                                 | 220        |
|                                      |            | 10.2 JDBC API .....                                | 220        |

|             |                                 |     |             |                            |     |
|-------------|---------------------------------|-----|-------------|----------------------------|-----|
| 10.3        | MySQL 简介 .....                  | 223 | 12.10       | 守护线程 .....                 | 283 |
| 10.4        | 数据库基本操作的 SQL 语法 ...             | 224 | 12.11       | 线程之间的通信流类 .....            | 284 |
| 10.5        | MySQL 的使用 .....                 | 224 | 小结 .....    | 286                        |     |
| 10.6        | JDBC 数据库基本操作 .....              | 226 | 习题 12 ..... | 286                        |     |
| 10.7        | 运用 JavaBean 进行<br>数据库操作 .....   | 233 | 第 13 章      | Java 网络编程 .....            | 288 |
| 10.8        | 数据库的批处理与事务操作 ....               | 237 | 13.1        | Java 网络编程概述 .....          | 288 |
| 10.9        | JDBC 操作 Access 数据库 .....        | 247 | 13.2        | InetAddress 类的应用 .....     | 289 |
| 小结 .....    | 249                             |     | 13.3        | URL 类的应用 .....             | 290 |
| 习题 10 ..... | 249                             |     | 13.3.1      | URL 简介 .....               | 290 |
| 第 11 章      | Java 输入、输出流 .....               | 250 | 13.3.2      | URL 类的常用方法 .....           | 290 |
| 11.1        | Java 输入、输出流概述 .....             | 250 | 13.4        | URLConnection 类 .....      | 292 |
| 11.2        | File 类的应用 .....                 | 251 | 13.5        | TCP、UDP、端口与套接字 .....       | 294 |
| 11.3        | 输入、输出流类 .....                   | 252 | 13.6        | 基于 TCP 的 Socket 网络编程 ..... | 294 |
| 11.4        | 文件字节输入、输出流类 .....               | 254 | 13.6.1      | ServerSocket 类 .....       | 295 |
| 11.5        | 文件字符输入、输出流类 .....               | 256 | 13.6.2      | Socket 类 .....             | 295 |
| 11.6        | 字节数组输入、输出流类 .....               | 257 | 13.6.3      | Socket 编程的一般流程 ...         | 296 |
| 11.7        | 过滤流类 .....                      | 257 | 13.7        | 基于 UDP 的网络编程 .....         | 305 |
| 11.8        | 随机访问文件 .....                    | 259 | 13.7.1      | DatagramPacket 类 .....     | 305 |
| 11.9        | Serializable 接口与<br>对象序列化 ..... | 260 | 13.7.2      | DatagramSocket 类 .....     | 306 |
| 11.10       | 标准输入、输出流 .....                  | 262 | 13.7.3      | 基于 UDP 网络编程的<br>一般流程 ..... | 306 |
| 11.11       | 文件对话框 .....                     | 263 | 13.8        | 基于组播的网络编程 .....            | 312 |
| 11.12       | 用 Desktop 类打开文件 .....           | 266 | 小结 .....    | 315                        |     |
| 小结 .....    | 266                             |     | 习题 13 ..... | 315                        |     |
| 习题 11 ..... | 267                             |     | 第 14 章      | Java 泛型与集合类 .....          | 316 |
| 第 12 章      | Java 多线程编程 .....                | 268 | 14.1        | 泛型 .....                   | 316 |
| 12.1        | 程序、进程与线程 .....                  | 268 | 14.1.1      | 泛型概述 .....                 | 316 |
| 12.2        | Java 多线程机制 .....                | 269 | 14.1.2      | 泛型类 .....                  | 317 |
| 12.3        | Java 多线程实现的方法 .....             | 269 | 14.1.3      | 泛型构造方法 .....               | 318 |
| 12.4        | 线程的生命周期与状态转换 ....               | 271 | 14.1.4      | 泛型方法 .....                 | 320 |
| 12.5        | Java 多线程调度机制 .....              | 272 | 14.1.5      | 泛型通配符 .....                | 320 |
| 12.6        | Thread 类 .....                  | 273 | 14.1.6      | 泛型接口 .....                 | 322 |
| 12.7        | 线程的让步 .....                     | 276 | 14.1.7      | 子类泛型 .....                 | 323 |
| 12.8        | 线程的联合 .....                     | 277 | 14.1.8      | 引入泛型的好处 .....              | 323 |
| 12.9        | 多线程的互斥与同步 .....                 | 278 | 14.2        | 集合类与接口 .....               | 324 |
| 12.9.1      | 线程的互斥 .....                     | 278 | 14.2.1      | Collection 接口 .....        | 324 |
| 12.9.2      | 互斥线程的协调 .....                   | 280 | 14.2.2      | Iterator 接口 .....          | 324 |
|             |                                 |     | 14.2.3      | List 接口 .....              | 325 |

|         |                                  |     |
|---------|----------------------------------|-----|
| 14.2.4  | ArrayList 类 .....                | 325 |
| 14.2.5  | LinkedList 类 .....               | 327 |
| 14.2.6  | Collections 类 .....              | 329 |
| 14.2.7  | Set 接口 .....                     | 331 |
| 14.2.8  | HashSet 类 .....                  | 331 |
| 14.2.9  | TreeSet 类 .....                  | 332 |
| 14.2.10 | EnumSet 类 .....                  | 334 |
| 14.2.11 | Map 接口 .....                     | 334 |
| 14.2.12 | HashMap 类 .....                  | 335 |
| 14.2.13 | SortedMap 接口与<br>TreeMap 类 ..... | 336 |
| 14.2.14 | EnumMap 类 .....                  | 337 |
| 小结      | .....                            | 338 |

|                                          |     |
|------------------------------------------|-----|
| 习题 14 .....                              | 338 |
| 第 15 章 Java Applet 编程简介 .....            | 340 |
| 15.1 Java Applet 简介 .....                | 340 |
| 15.2 Applet 程序的编写方法 .....                | 340 |
| 15.3 Applet 类的主要方法 .....                 | 342 |
| 15.4 <applet>标记的属性及 Applet<br>参数传递 ..... | 344 |
| 15.5 Applet 的组件与事件处理 .....               | 345 |
| 小结 .....                                 | 348 |
| 习题 15 .....                              | 348 |
| 参考文献 .....                               | 349 |

# 第 1 章 | Java 简介与 Java SE 程序开发环境

## 【本章内容提要】

- Java 的发展;
- Java 的 3 种平台;
- Java 的特点;
- Java 程序的运行机制与 Java 虚拟机;
- Java 程序分类;
- JDK 的安装与配置;
- JDK 环境下开发 Java 程序;
- Eclipse 环境下开发 Java 程序。
- 其他几个问题。

## 1.1 Java 的发展

1990 年 12 月, Sun 公司就由 James Gosling 等人发起一个叫做 Green 的开发小组, 主要目标是家用消费类电子产品开发一个分布式代码系统, 起初的名字叫做 Oak, 但是有一家公司已经用了这个名字, 于是就以一个盛产咖啡的小岛名 Java 命名了。1996 年, Sun 公司发布了 JDK (Java Development Kit) 1.0, 其中包括了 JRE (Java Runtime Environment) 及 JDK 两部分。1997 年推出的 JDK1.1, 其中增加了 JIT (即时编译) 功能, 有效提高了 Java 程序运行效率。1998 年底, Sun 公司发布了 JDK1.2, 这是 Java 发展史上的一个里程碑。1999 年 6 月, Sun 公司将 Java 分为三个版本 J2SE (标准版), J2EE (企业版) 及 J2ME (微型版)。从 2000 年到 2004 年, JDK1.3、JDK1.4、J2EE1.3、J2SE1.4 等陆续被发布。2004 年 9 月, J2SE1.5 被发布, 成为 Java 语言发展史上的又一个里程碑, 为了表示该版本的重要性, J2SE1.5 更名为 Java SE 5.0。2005 年 6 月, JavaOne 大会召开, Sun 公司公开 Java SE 6。此时, 将 Java 的各版本更名, 以取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。2006 年 12 月, JDK 6.0 正式版发布。2009 年, Sun 公司被 Oracle 公司收购, 2011 年, Oracle 发布了 Java 7.0 的正式版, 2014 年又发布了 Java 8.0 的正式版。

目前, Java 的开发与应用涉及很多方面, 包括桌面级的开发、网络开发和嵌入式开发等。在动态网站和企业级开发中, Java 作为一种主流编程语言占到了很大份额; 在嵌入式方面的发展更是迅速, 现在流行的手机游戏, 几乎都是应用 Java 语言开发的。

## 1.2 Java 的 3 种平台

Java 开发平台有 Java SE (Java Platform Standard Edition, Java 平台标准版)、Java EE (Java Platform Enterprise Edition, Java 平台企业版) 及 Java ME (Java Platform Micro Edition, Java 平台微型版) 3 种。

**Java SE:** 它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 包含了构成 Java 语言的核心类库和基础开发环境, 并为 Java EE 和 Java ME 提供开发基础支持。

**Java EE:** 企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的, 它提供 Web 服务、组件模型、管理和通信 API, 可以用来构建 Web 应用程序以及面向服务体系结构的企业级应用。

**Java ME:** 为在移动设备和嵌入式设备 (如手机、PDA、电视机顶盒等) 上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的联网和离线应用程序的丰富支持。

## 1.3 Java 的特点

Java 具有入门简单、面向对象, 跨平台、安全性、并发机制等一系列特点。

- **跨平台性:** 依赖于 Java 独有的虚拟机机制, 使得 Java 源程序经编译后形成的字节码不依赖于任何具体的平台, 只要具有 JRE (Java Runtime Environment), 就可以经解释后执行。因此, Java 程序具备“一次编译, 到处运行”的跨平台特性。
- **简单性:** Java 语言语法借鉴了 C、C++ 的语法。但没有显式地运用指针、运算符重载、多重继承等难以理解的概念, 并通过实现垃圾自动收集机制简化了程序设计者的内存管理工作。
- **面向对象:** Java 语言是纯面向对象的编程语言, 它实现了面向对象编程思想中的抽象、封装、继承及多态等基本特性。同时, 它本身提供了强大丰富的系统类与接口包。
- **解释执行:** Java 的设计者设计 Java 的主要目的, 就是希望可以做到, “编写一次, 到处运行”, Java 源程序被编译成字节码, 这种字节码在任何 Java 运行环境中由 Java 虚拟机解释执行, 这种方式保证了 Java 的与平台无关性和可移植性。解释执行与即时编译技术的完美结合, 提供了相当高的运算性能。
- **安全性:** Java 安全性包括语言级安全性、编译时安全性、运行时安全性、可执行代码安全性。语言级安全性指 Java 的数据结构是完整的对象, 这些封装过的数据类型具有安全性; 编译时要进行 Java 语言和语义的检查, 保证每个变量对应一个相应的值及编译后生成 Java 类; 当 Java 源程序经过编译后形成的字节码被加载到内存后, Java 解释器中的代码校验器对字节码的合法性进行检查, 若发现有破坏 Java 安全性的操作, 则给出错误信息并拒绝执行该程序; Java 类在网络上使用时, 对它的权限进行了设置, 保证了被访问用户的安全性。
- **健壮性:** Java 致力于检查程序在编译和运行时的错误, 其类型检查可以检查出许多开发早期出现的错误; Java 支持自动内存管理, 减少了内存出错的可能性; Java 采用对象数组并能检测数组边界, 避免了覆盖数据的可能; 基于异常 (Exception) 处理机制, 在编译时, 能揭示出可能出现但未被处理的异常, 帮助程序员正确地进行选择, 以防止系统的崩溃。

- 多线程：Java 支持用户编写包含多条执行线索的程序，每条执行线索对应一个线程，每个线程具有独立的任务，通过 Java 虚拟机的调度并发地完成多个任务。
- 动态特性：Java 的动态特性表现在两个方面：其一是在 Java 语言中，可以简单、直观地查询运行时的信息；其二是可以将新代码加入到一个正在运行的程序中。
- 分布性：Java 拥有广泛的能轻易地处理 TCP/IP 协议的运行库，这使得在 Java 中比在 C 或 C++ 中更容易建立网络连接。Java 应用程序可以借助 URL 通过网络开启和存取对象，就如同存取一个本地文件系统一样简单。

## 1.4 Java 程序的运行机制与 Java 虚拟机

### 1.4.1 Java 程序的运行机制

各种计算机程序在运行机制上，大体可被分为编译型和解释型两种。而 Java 程序需要经过编译及解释执行两个阶段。

一个 Java 程序要经过编辑、编译、解释执行等几个阶段。首先，运用 Java 语言编写的程序以“主类名.java”的形式保存，被称为 Java 源程序；其次，Java 源程序被 JDK 提供的编译命令 `javac` 编译后，形成的“主类名.class”文件，该文件被称为字节码文件；最后，Java 字节码文件被 Java 虚拟机 JVM（Java Virtual Machine）解释为可在不同平台下执行的指令并执行，如图 1.1 所示。

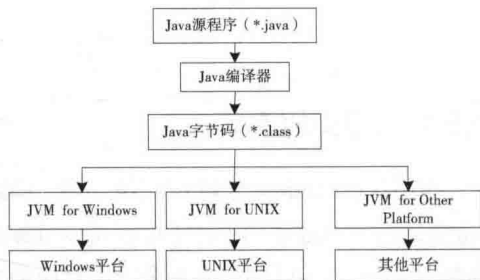


图 1.1 Java 程序运行过程

Java 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，是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件，它必须由 JVM 来解释执行，因此说 Java 是一种在编译基础上进行解释执行的语言。

### 1.4.2 Java 虚拟机

Java 程序是如何做到“一次编译，到处运行”呢？这正是通过 JVM 来实现的。JVM 可以被理解成一个用软件模拟的、以字节码为机器指令的虚拟计算机，它屏蔽了 Java 程序底层运行平台（操作系统和硬件）的差别，在执行字节码时，把字节码解释成适合于不同具体平台上的机器指令并将机器指令交给本地的操作系统来运行。

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，并且 Java 字节码也只能运行于 JVM 之上。只要在不同的计算机上安装了针对特定具体平台的 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 把这种不同软硬件平台的具体差别隐藏起来，从而实现了真正的二进制代码级的跨平台移植，Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。

Java 程序通过 JVM 可以实现跨平台特性，但 JVM 是不跨平台的。也就是说，不同操作系统之上的 JVM 是不同的，如 Windows 平台之上的 JVM 与 UNIX 平台上的 JVM 是不同的。

# 1.5 Java 程序分类

针对 Java SE、Java EE 以及 Java ME 等不同平台，Java 程序可以被分成如下几类。

- Java 应用程序 (Java Application)：是必须包括一个 main 方法的独立完整的桌面应用程序，main 方法作为程序的入口。
- Java 小应用程序 (Java Applet)：将其编译形成的字节码嵌入在网页 (.html) 文件中并被客户端浏览器加载执行的一种 Java 程序。
- Java Servlet：在 Java EE 平台下开发的，运行在 Web 服务器端的 Java 程序，为了能够支持 Servlet 的运行，Web 服务器端必须安装包含 Java 虚拟机的服务器软件。
- 手机上运行的 Java ME 应用程序 MIDlet (Mobile Information Devices let)，即移动信息设备小程序。

Java Application 与 Java Applet 是 Java SE 平台下两种基本的程序类型。但是目前，Java Applet 的应用较少，所以，本书中绝大多数程序是 Java Application 程序。

# 1.6 JDK 的安装与配置

## 1.6.1 JDK 的安装

本书以 Windows 8 操作系统环境，JDK1.7 (JDK7) 为例，在 JDK1.7 的官方下载地址 <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> 下载 JDK1.7。根据操作系统是 32 位或 64 位的，分别要下载对应的 jdk-7u79-windows-i586.exe (32 位) 或者 jdk-7u79-windows-x64.exe (64 位) 开发包 (见图 1.2)。

|             |           |                           |
|-------------|-----------|---------------------------|
| Windows x86 | 138.31 MB | jdk-7u79-windows-i586.exe |
| Windows x64 | 140.06 MB | jdk-7u79-windows-x64.exe  |

图 1.2 JDK1.7 开发包

下面以 jdk-7u79-windows-x64 的安装与配置为例，介绍 JDK1.7 的安装与配置方法。

双击安装程序，按照安装向导 (见图 1.3) 提示一步步完成即可。安装过程中可以更改安装路径 (见图 1.4)，然后单击“下一步”按钮继续完成安装，当弹出图 1.5 所示的向导页面时，单击其中的“后续步骤”按钮后可浏览和下载 JDK1.7 的开发文档等资源 (见图 1.6)，同时，可进一步了解 JDK1.7 的其他特性等。



图 1.3 JDK1.7 安装向导

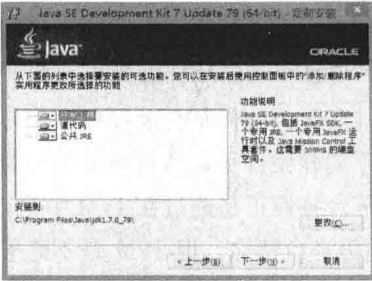


图 1.4 JDK1.7 安装路径选择

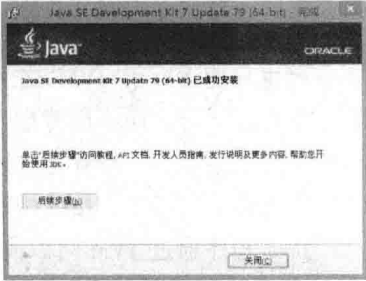


图 1.5 JDK1.7 安装完成

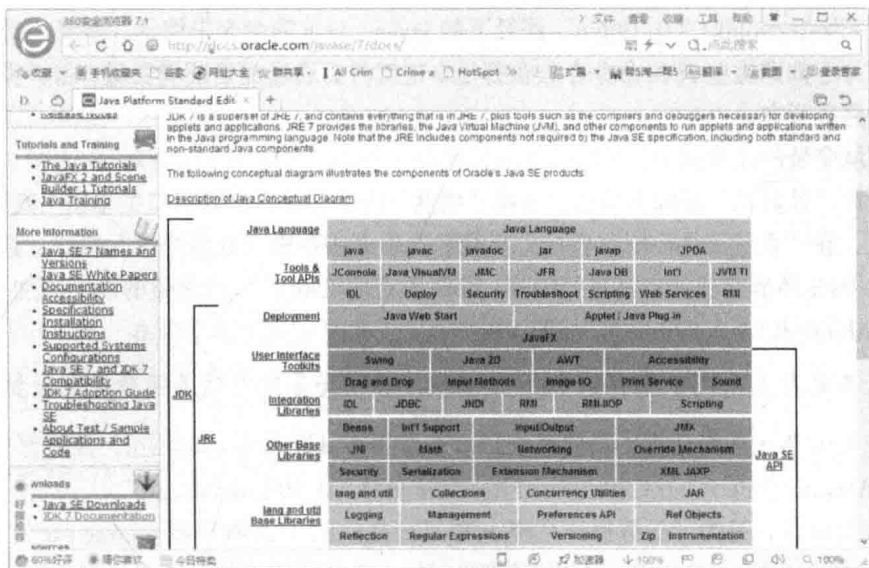


图 1.6 JDK 文档页面

JDK1.7 安装完成后，其安装路径下会包含图 1.7 所示的 jdk1.7.0\_79 以及 jre7 两个文件夹。其中，JRE7(Java SE Runtime Environment)提供了运行 Java 程序不可或缺的环境，jdk1.7.0\_79 中产生以下几个主要子目录，它们各自的主要作用如表 1.1 所示。

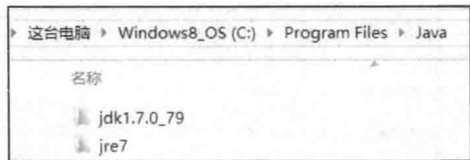


图 1.7 JDK 安装路径下的文件夹

表 1.1 jdk1.7.0\_79 中主要子目录及其各自的主要作用

| 名 称     | 作 用              | 名 称 | 作 用         |
|---------|------------------|-----|-------------|
| bin     | 实用工具程序及 JDK 命令集合 | jre | Java 运行环境文件 |
| include | 系统 C 语言的头文件等     | lib | Java 类库文件   |

安装好 JDK 后，是否就意味着可以用 JDK 的命令来编译、执行 Java 程序了呢？为了验证这一点，打开 Windows 命令提示符窗口，在其中键入 JDK 命令 javac 并回车。系统会提示“‘javac’不是内部或外部命令，也不是可执行程序或批处理文件”，如图 1.8 所示。这就说明，系统还不认识 JDK 命令。因此，还需要配置环境变量。

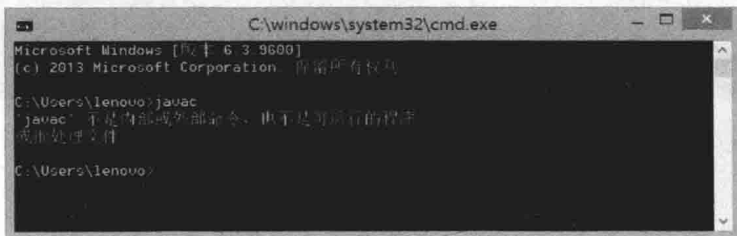


图 1.8 在命令提示符下键入 JDK 命令

## 1.6.2 JDK 环境变量的配置

JDK 安装完成后，需要配置环境变量。为什么要配置环境变量？这是因为我们需要使用

“C:\Program Files\Java\jdk1.7.0\_79\bin” 路径下的 javac、java 等命令来编译、解释执行 Java 程序。但操作系统并不知道这些 JDK 命令存放在哪里，这就需要通过配置环境变量，告诉操作系统这些命令的存放路径所在。

配置环境变量的步骤如下：

- 在桌面“计算机”图标上右击，选择“属性”命令，在弹出的窗口中单击“高级系统设置”按钮，在“系统属性”对话框的“高级”选项卡中单击“环境变量”按钮，选择系统中原有的 Path 环境变量并对其进行编辑，即在原有的 Path 环境变量值的前面添加“;C:\Program Files\Java\jdk1.7.0\_79\bin;”，如图 1.9 所示。

注意：不要覆盖系统中原有的 Path 变量值，否则会导致有些系统命令无法执行。

- 在图 1.10 所示的对话框中新建系统环境变量并将其命名为 classpath，其值为“;C:\Program Files\Java\jre7\lib\rt.jar; C:\Program Files\Java\jdk1.7.0\_79\lib\tools.jar;”。

对 Path 的设置，是为了告诉操作系统 JDK 平台提供的 Java 编译器 (javac.exe)、Java 解释器 (java.exe) 等命令在 Java 安装目录的 bin 文件夹中。对 classpath 的设置，是为了使编译程序能够找到用户定义的和系统类所在的包。

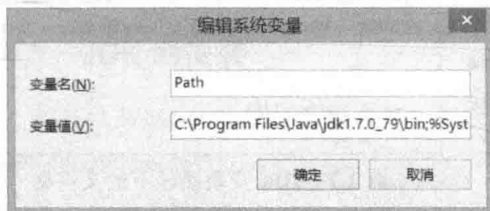


图 1.9 设置 Path

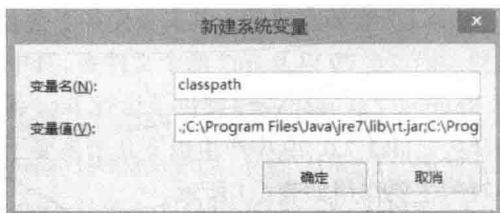


图 1.10 设置 classpath

当配置好环境变量后，重新打开 Windows 命令提示符窗口，在其中键入 javac 命令并回车（如图 1.11 所示），发现 javac 的命令参数选项逐一列出，就表明 JDK 环境变量配置好了。



图 1.11 配置环境变量后测试 JDK 命令

## 1.7 JDK 环境下开发 Java 程序

### 1.7.1 Java Application 的开发

Java 源程序是以 .java 为扩展名，可以用各种 Java 集成开发环境中的源代码编辑器来编写，也可以用其他文本编辑工具如 Windows 中的记事本、JCreator 等编辑 Java 源程序。

Java Application 程序的开发可以分为以下三个步骤：

### (1) 编辑源程序

利用记事本等编辑工具完成源程序编辑,形成“主类名.java”的 Java 源文件。

### (2) 编译源程序

使用 JDK 的 javac.exe 命令将源文件编译生成字节码文件“主类名.class”。

### (3) 运行程序

对于 Java Application,用 JDK 的 java 命令对字节码文件进行解释执行。

#### 【例题 1\_1】第一个 Java Application 程序。

##### (1) 打开记事本,在其中编辑如下程序:

```
public class MyFirstApplication {  
    public static void main(String[] args) {  
        System.out.println("这是我的第一个 Java Application 程序!");  
        System.out.println("This is my first Java Application.");  
    }  
}
```

(2) 将以上程序保存为 MyFirstApplication.java,保存时一定要在“另存为”对话框的“保存类型”列表框中选择“所有文件”选项,如图 1.12 所示,否则会默认保存为.txt 文件。假定保存路径选择 E:\Java Pro。

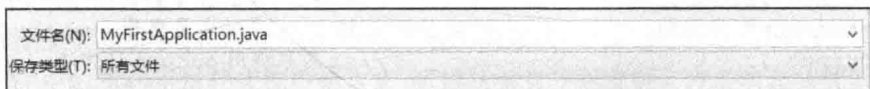


图 1.12 保存 Java 源程序

(3) 打开 Windows 命令提示符窗口,将路径转到当前保存源程序文件的 E:\Java Pro 下。在当前命令提示符下键入 E:并回车,将根目录转到 E:\目录下。

(4) 用 cd 命令进入 Java Pro 子目录下(见图 1.13),继而键入 javac MyFristApplication.java 并回车对源程序进行编译,编译过程中若有语法错误,则会报错。

(5) 若没有语法错误,编译后在 Java Pro 子目录下会生成对应的字节码文件 MyFirstApplication.class;再输入 java MyFristApplication 并回车来解释执行程序,得到正确的输出结果(见图 1.14)。



图 1.13 用 javac 命令编译 Java 源程序



图 1.14 用 java 命令执行 Java 程序

程序说明:

- 类是组成 Java 源程序的基本单位,一个 Java 源程序是由若干个类组成的。本例只有 MyFirstApplication 一个类。
- “public class MyFirstApplication”被称为 Java 类的声明部分,其中,public 及 class 都是 Java 的关键字,public 表明该类的访问权限为公有,class 是定义类的关键字 MyFirstApplication 是类名标识符。