

普通高等教育“计算机类专业”规划教材

Java程序设计 案例教程

李明 吴琼 主编

韩旭明 时兵 王润芳 副主编



清华大学出版社

C14006804

TP312JA
1530

普通高等教育“计算机类专业”规划教材

Java程序设计 案例教程

李明 吴琼 主编

韩旭明 时兵 王润芳 副主编



清华大学出版社



北航

C1693854

TP312JA

1530

内 容 简 介

在计算机人才培养中,实践环节是非常重要的,特别是对那些准备走上 IT 行业的人员来说,实践环节更是重中之重。本书针对市场上一些教材比较侧重理论知识而轻视实践的现象,在书中设计了案例教学环节。在教材的编写中,坚持“理论够用,侧重实践;边学边做,做中学习”的原则;立足于“看得懂、学得会、用得上”的策略;由浅入深、循序渐进地介绍相关知识点。为适应面向应用型人才的培养,该教材以通俗易懂的讲解和丰富的实例为手段,目的在于培养学生解决实际问题的能力。

本书力求通过对核心知识的讲解,使初学者尽快了解 Java 程序设计的核心所在,通过对核心知识的扩展使读者提高编程能力,掌握编程技巧。通过本书的学习,读者可以掌握 Java 面向对象编程的思想和在当前不同 IT 技术中需要了解和掌握的 Java 内容。本书内容全面实用,实例生动简单,易于读者学习和掌握,既可以作为普通高等院校计算机专业及理工类各专业学生学习 Java 语言的教材,也可以作为各种考试的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计案例教程/李明,吴琨主编. —北京:清华大学出版社,2013

普通高等教育“计算机类专业”规划教材

ISBN 978-7-302-33022-6

I. ①J… II. ①李… ②吴… 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(2013)第 145971 号

责任编辑:白立军

封面设计:傅瑞学

责任校对:时翠兰

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:22

字 数:537 千字

版 次:2013 年 10 月第 1 版

印 次:2013 年 10 月第 1 次印刷

印 数:1~2000

定 价:34.50 元

Java 语言是目前 IT 行业的主流技术之一,是面向对象的、支持多线程的编程语言。它具有高度的安全性、可移植性和可重用性。Java 程序设计是中国高校计算机类专业的专业课程之一,培养学生用面向对象的思想解决实际问题是该课程的教学目标。作者作为 Sun 公司的 Java 培训讲师,结合多年在高校的实际教学经验,编写了本书,本书侧重培养学生解决问题的能力。

在计算机人才培养中,实践环节是非常重要的,特别是对那些准备走上 IT 行业的人员来说,实践环节更是重中之重。本书针对市场上一些教材比较侧重理论知识而轻视实践的现象,在书中设计了案例教学环节。在教材的编写中,坚持“理论够用,侧重实践;边学边做,做中学习”的原则;立足于“看得懂、学得会、用得上”的策略;由浅入深、循序渐进地介绍相关知识点。为适应面向应用型人才的培养,该教材以通俗易懂的讲解和丰富的实例为手段,目的在于培养学生解决实际问题的能力。

本书针对 Java 语言的不同应用层面,分为三个篇章来介绍——基础篇、交互篇和高级应用篇。

第一部分是基础篇,由第 1 章至第 7 章组成,以初学者的角度从 Java 语言最基本入门概念开始讲述,对 Java 面向对象程序设计的基本概念、基础知识进行深入浅出的讲解。

第二部分是交互篇,由第 8 章和第 9 章组成,针对人机界面交互进行讲解,包括图形用户界面的管理和布局、多媒体技术等。

第三部分是高级应用篇,由第 10 章至第 12 章组成,针对数据库、网络、多线程方面的编程方法进行了讲解。

每章都安排了大量例题、案例和习题,便于教师教学和学生练习。

本书具有如下特色。

(1) 针对应用型人才的培养,以简单直接的案例方式让学生掌握程序设计的本质。

(2) 针对学习 Java 语言的不同用途,分为基础篇、交互篇和高级应用篇三部分。教学中可根据人才培养方案对内容进行取舍。如果 Java 只作为后台编程语言,可以略过交互篇的学习。

(3) 本书在内容的组织上注重不同应用方向的不同需求;在思想上注重强调面向对象的特征;在语言组织上注重简单易懂;在实例上注重实用性。

本书力求通过对核心知识的讲解,使初学者尽快了解 Java 程序设计的核心所在,通过对核心知识的扩展使读者提高编程能力,掌握编程技巧。通过本书的学习,读者可以掌握 Java 面向对象编程的思想和在当前不同 IT 技术中需要了解和掌握的 Java 内容。本书内容全面实用,实例生动简单,易于读者学习和掌握。既可以作为普通高等院校计算机专业及理工类各专业学生学习 Java 语言的教材,也可以作为各种考试的参考书。

本书由李明、吴琼主编,韩旭明、时兵、王润芳副主编。

在本书的编写过程中,参考了网络上很多个人的博客和留言,在此表示感谢。由于作者水平有限,书中难免存在不妥和疏漏之处,敬请广大读者批评指正。

如果读者需要书中的源程序代码和教学用的 PPT 文件,可以从清华大学出版社网站(www.tup.com.cn)下载。

本书共分 11 章,主要介绍 Java 语言的基本语法、面向对象编程、集合、多线程、网络编程、数据库编程、XML 编程、JSP 编程、Struts 框架、Spring 框架、Hibernate 框架等。本书可作为高等院校计算机专业及相关专业的教材,也可供从事 Java 开发的工程技术人员参考。

本书由清华大学出版社组织编写,由清华大学出版社负责出版。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。本书的编写得到了清华大学出版社的大力支持,在此表示衷心的感谢。

第 1 章	Java 简介	1
1.1	Java 技术	1
1.2	Java 的主要应用	3
1.3	基本的 Java 应用程序	4
1.3.1	程序说明	4
1.3.2	运行环境说明	5
1.3.3	错误诊断	15
1.4	本章小结	16
1.5	习题	16
1.5.1	选择题	16
1.5.2	简答题	17
第 2 章	Java 语言基础	18
2.1	基础知识	18
2.1.1	标识符	18
2.1.2	关键字	18
2.1.3	数据类型	19
2.1.4	常量和变量	21
2.1.5	注释	23
2.2	运算符和表达式	25
2.2.1	运算符	25
2.2.2	表达式	29
2.2.3	基本数据类型间的转换	30
2.2.4	案例描述	31
2.3	流程控制	32
2.3.1	顺序结构	32
2.3.2	选择结构	32
2.3.3	循环结构	36
2.3.4	案例描述	40
2.4	本章小结	41
2.5	习题	42
2.5.1	选择题	42
2.5.2	编程题	44
2.5.3	简答题	44

第 3 章 类和对象	45
3.1 面向对象概述	45
3.2 类	45
3.2.1 类的定义	45
3.2.2 类的定义格式	46
3.2.3 案例描述	49
3.3 对象	50
3.3.1 创建对象	50
3.3.2 使用对象	51
3.3.3 构造方法	53
3.3.4 对象初始化过程	54
3.3.5 案例描述	55
3.4 源文件布局与访问权限	57
3.4.1 包	57
3.4.2 引入类	58
3.4.3 Java 的源文件布局	58
3.4.4 访问权限	59
3.4.5 案例描述	61
3.5 方法的调用	63
3.5.1 参数传递	64
3.5.2 递归调用	67
3.5.3 案例描述	69
3.6 类的继承	70
3.6.1 子类的定义	71
3.6.2 类的继承规则	72
3.6.3 类的多态	76
3.6.4 案例描述	80
3.7 本章小结	82
3.8 习题	82
3.8.1 选择题	82
3.8.2 读程序	84
3.8.3 编程题	85

第 4 章	数组和字符串	86
4.1	数组	86
4.1.1	基本概念	86
4.1.2	数组的复制	88
4.1.3	案例描述	89
4.2	多维数组	90
4.2.1	基本知识	91
4.2.2	异类集合	93
4.2.3	案例描述	93
4.3	String 类	95
4.3.1	String 类的对象	95
4.3.2	String 类常用方法	97
4.3.3	字符串类型与基本数据类型的相互转换	99
4.3.4	案例描述	100
4.4	可变字符串	101
4.4.1	StringBuffer 类	102
4.4.2	StringBuilder 类	104
4.4.3	案例描述	104
4.5	本章小结	105
4.6	习题	106
4.6.1	选择题	106
4.6.2	读程序	107
4.6.3	编程题	108
第 5 章	类的高级特性	109
5.1	static 关键字	109
5.1.1	静态变量	109
5.1.2	静态方法	110
5.1.3	静态代码块	111
5.1.4	static 修饰类	112
5.1.5	案例描述	112
5.2	final 和 abstract 关键字	114
5.2.1	final 关键字	114

88	5.2.2 abstract 关键字	115
88	5.2.3 案例描述	117
88	5.3 接口	119
88	5.3.1 接口的定义	119
88	5.3.2 接口的实现	120
88	5.3.3 接口回调	121
100	5.3.4 接口与抽象类的异同	122
88	5.3.5 案例描述	122
88	5.4 嵌套类和内部类	125
88	5.4.1 内部类	125
88	5.4.2 嵌套类	127
88	5.4.3 案例描述	129
88	5.5 封装类和集合	131
88	5.5.1 Wrapper 类	131
88	5.5.2 集合	131
88	5.5.3 案例描述	135
88	5.6 本章小结	136
88	5.7 习题	137
88	5.7.1 选择题	137
88	5.7.2 读程序	138
88	5.7.3 编程题	138
88	5.7.4 简答题	139
88	第 6 章 异常处理	140
88	6.1 异常的概念	140
88	6.2 异常处理机制	143
88	6.2.1 基本知识	143
88	6.2.2 案例描述	145
88	6.3 throw 语句和 throws 语句	147
88	6.3.1 throw 语句	148
88	6.3.2 throws 语句	148
88	6.3.3 案例描述	151
88	6.4 自定义异常类	152
88	6.4.1 基本知识	152

6.4.2	案例描述	153
6.5	本章小结	155
6.6	习题	156
6.6.1	选择题	156
6.6.2	编程题	157
第 7 章	文件和流	158
7.1	流的简介	158
7.2	File 类	161
7.2.1	基本知识	161
7.2.2	案例描述	164
7.3	字节流	166
7.3.1	字节流父类	167
7.3.2	文件字节流	168
7.3.3	其他字节流	170
7.3.4	案例描述	175
7.4	字符流	178
7.4.1	字符流父类	179
7.4.2	文件字符流	180
7.4.3	其他字符流	181
7.4.4	案例描述	185
7.5	文件的随机读写	187
7.5.1	基本知识	187
7.5.2	案例描述	189
7.6	本章小结	190
7.7	习题	191
7.7.1	选择题	191
7.7.2	编程题	191
第 8 章	图形用户界面	192
8.1	图形用户界面简介	192
8.2	使用 AWT 设计图形用户界面	193
8.2.1	组件	193
8.2.2	容器	194
8.2.3	布局管理器	197

8.2.4	事件处理	203
8.2.5	案例描述	212
8.3	Swing 组件	215
8.3.1	Swing 组件简介	215
8.3.2	Swing 中的容器	215
8.3.3	Swing 中的组件	223
8.3.4	案例描述	236
8.4	本章小结	240
8.5	习题	240
8.5.1	选择题	240
8.5.2	编程题	241
第 9 章	Java Applet 及多媒体编程	242
9.1	Applet 小程序介绍	242
9.1.1	一个 Applet 程序	242
9.1.2	Applet 的安全机制	244
9.2	Applet 小程序的生命周期	244
9.2.1	基本知识	244
9.2.2	案例描述	246
9.3	多媒体编程	249
9.3.1	基本知识	249
9.3.2	案例描述	255
9.4	本章小结	257
9.5	习题	257
9.5.1	选择题	257
9.5.2	编程题	258
第 10 章	数据库编程	259
10.1	JDBC 的概述	259
10.1.1	JDBC 的介绍	259
10.1.2	数据库驱动程序的分类型	260
10.2	数据库连接过程	261
10.2.1	JDBC-ODBC 桥驱动	261
10.2.2	本地 API 驱动程序连接	267
10.2.3	案例描述	272

10.3	常用接口的使用方法	276
10.3.1	Connection 接口	276
10.3.2	Statement 接口	278
10.3.3	ResultSet 接口	280
10.3.4	PreparedStatement 接口	283
10.3.5	案例描述	285
10.4	事务回滚	290
10.4.1	基础知识	290
10.4.2	案例描述	292
10.5	本章小结	298
10.6	习题	298
10.6.1	选择题	298
10.6.2	编程题	299
第 11 章	多线程编程	300
11.1	线程的概念	300
11.1.1	基本概念	300
11.1.2	多线程的优点	301
11.2	线程的生命周期	301
11.2.1	线程的状态	301
11.2.2	线程的方法	302
11.2.3	线程的优先级别	302
11.3	线程的创建	303
11.3.1	继承 Thread 类创建线程	303
11.3.2	实现 Runnable 接口 创建线程	304
11.3.3	案例描述	305
11.4	线程的同步	306
11.4.1	问题提出	306
11.4.2	同步的解决机制	307
11.4.3	案例描述	312
11.5	本章小结	314
11.6	习题	314
11.6.1	选择题	314

11.6.2	简答题	315
11.6.3	编程题	315
第 12 章 网络编程		316
12.1	网络连接方式	316
12.1.1	网络基础	~
12.1.2	不同协议层对应的网络 连接方法	317
12.2	URL 连接方式	317
12.2.1	概念	317
12.2.2	URL 对象连接模型	319
12.2.3	URLConnection 对象 连接模型	319
12.2.4	案例描述	321
12.3	Socket 连接方式	322
12.3.1	概念	322
12.3.2	连接模型	324
12.3.3	案例描述	326
12.4	Datagram 连接方式	329
12.4.1	概念	329
12.4.2	连接模型	330
12.4.3	案例描述	332
12.5	本章小结	337
12.6	习题	337
12.6.1	选择题	337
12.6.2	简答题	338
12.6.3	编程题	338
参考文献		339

第 1 章 Java 简介

Java 技术是当前使用最广泛的 IT 技术,它在信息技术领域掀起了一场巨大的革命。本质上说,Java 不仅仅是一种编程语言,还是一种开发环境、一种应用程序环境、一种部署环境。本章主要介绍 Java 技术的基本知识,并重点探讨 Java 技术在不同方面的应用。

【本章要求】

- (1) 了解 Java 的特性。
- (2) 熟悉 Java 程序设计的基本结构。
- (3) 掌握 Java 开发环境的选择及安装。

1.1 Java 技术

Java 是由 SUN Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。

Java 语言是一种可以撰写跨平台应用程序的面向对象的程序设计语言,它最开始起源于 Sun 公司在 1990 年 12 月开始研究的 Green 内部项目。团队最初考虑使用 C 语言,但是很多成员发现 C 太复杂,经常出现使用错误。于是决定创造出一种全新的语言,命名为 Oak(橡树)。但是 Oak 已被一家显卡制造商注册,因此将 Oak 改名为 Java。

Java 是印度尼西亚爪哇岛的英文名称,因盛产咖啡而闻名。Java 语言中的许多类名和工具,都与咖啡有关。如 JavaBeans、NetBeans 以及 ObjectBeans 等。Java 的标识也是一杯正冒着热气的咖啡。

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 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,同时拥有全球最大的开发者使用群。

1. Java 的含义

Java 最初是 Sun 公司推出的新一代面向对象程序设计语言。但随着众多软件开发商的支持和行业的认可,Java 技术已经不仅仅是一种语言,还是一种开发环境、一种应用程序环境、一种部署环境。

Java 作为一种语言,语法与 C++ 的语法相似,任何传统的编程语言所能创建的应用程序,都可以用 Java 编程语言来创建。

Java 作为一种开发环境,提供了一系列工具,如编译器、解释器、文档生成器、类文件、打包工具等。

Java 技术建立的应用程序是独立的程序,可运行于任何安装了 Java 运行环境(JRE)的计算机中。

Java 作为部署环境,主要有两种。一种是 JRE,它包含一套完整的 Java 技术所有的类文件,其中包括语言类、图形界面类和高级的集合 API 等。另一种是浏览器,大多数商业浏览器都提供 Java 技术的解释器和运行时环境。

2. Java 的特点

(1) 简单。Java 语言的语法与 C 语言和 C++ 语言很接近。而且 Java 丢弃了 C++ 中操作符重载、多继承、自动强制类型转换等很难理解的特性。特别是 Java 语言不使用指针, 自动进行垃圾回收, 只支持单继承的特性使 Java 语言变得容易掌握。

(2) 分布式。Java 语言支持 Internet 应用的开发, Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。

(3) 健壮。Java 的强类型机制、异常处理、垃圾自动回收机制等是 Java 程序健壮性的重要保证。Java 的安全检查机制使得 Java 更具健壮性。

(4) 安全。Java 对通过网络下载类具有一个安全防范机制(类加载器), 以防恶意代码的攻击。除此以外, 分配不同的名字空间以防替代本地的同名类、字节代码检查、安全管理机制让 Java 应用设置更加安全。

(5) 可移植。Java 程序在 Java 平台上被编译为字节码格式, 字节码格式的文件可在 Java 平台的虚拟机上执行, 因此, Java 程序可以在任何装了 Java 虚拟机的平台环境中运行。

(6) 多线程。在 Java 语言中, 线程是一种特殊的对象, 线程的活动由一组方法来控制。Java 语言支持多个线程的同时执行, 并提供多线程之间的同步机制。

(7) 动态。Java 程序需要使用的类可以动态地被载入到运行环境, 也可以通过网络载入所需要的类。这也有利于软件的升级。

3. Java 的特性

Java 技术体系结构通过三个特性实现了上述特点: Java 虚拟机(JVM)、垃圾回收机制、代码安全性检查。下面分别介绍三者 in Java 技术中的作用。

1) Java 虚拟机

Java 虚拟机简称为 JVM, 就是在真实的机器中用软件模拟实现的一种虚拟机器, 为软件的运行提供了一个平台。Java 的可执行程序在 Java 虚拟机上运行。

Java 应用程序的编译、解释、运行工作过程如图 1-1 所示。程序编写后被保存为 *. java 文件, 编译器将其编译成 *. class 文件。程序执行时, *. class 文件被加载到类加载器上, 通过字节码检验器的校验, 解释器进行解释执行, 得出结果。图中由虚线围起来的部分就是 Java 虚拟机。因为 Java 虚拟机将程序和硬件平台隔离开来, 程序只对 Java 虚拟机运行, 所以 Java 程序的可移植性非常好。

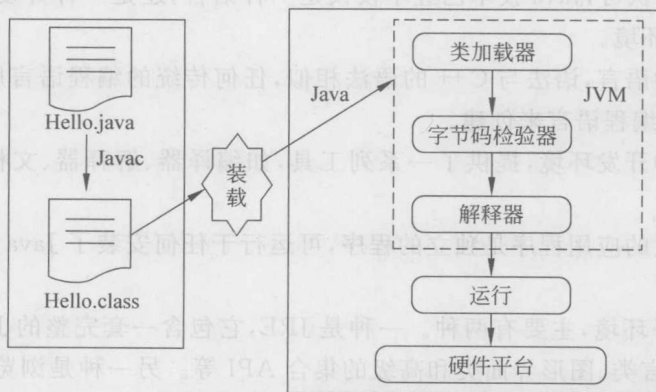


图 1-1 Java 程序运行流程

2) 垃圾回收机制

垃圾回收机制是 Java 语言区别于其他程序设计语言的显著特性。它把程序员从手工回收内存空间的繁重工作中解脱出来。在 C 或 C++ 等语言中,程序员开辟内存空间,使用结束后要负责释放该内存空间。在 Java 中,当一个内存空间没有引用指向它时,该内存空间会被系统认定为垃圾。Java 虚拟机的一个系统级线程会自动释放该内存空间。垃圾回收意味着程序不再需要的对象就是“无用信息”,这些信息将被丢弃。垃圾回收机制能自动释放内存空间,减轻程序员的负担,提高编程效率。

Java 虚拟机的垃圾回收在两种情况下进行:一种情况是 Java 虚拟机空闲时,检查内存是否为垃圾,进行垃圾回收,释放空间。另一种情况是在系统没有可用空间时,进行垃圾回收,释放空间。Java 的垃圾回收机制是自动进行的,不需要程序员干预。在实现的过程中,可能存在很大的差异。

3) 代码安全性检查

Java 运行时主要执行三个任务:加载代码、校验代码、执行代码。其中,加载代码由类加载器完成,校验代码由字节码校验器完成,执行代码由解释器完成。

首先由类加载器将类文件(也就是编译后产生的 *.class 文件)加载到虚拟机中,为了增加安全性,将本地文件系统的类和网络系统导入的类区分开来。本机存储的类总是先被加载,这就可以限制任何“特洛伊木马”程序,一旦程序所需的所有类加载完毕,执行文件的内存就固定了。

其次,字节码校验器进行校验。所有通过网络导入的类文件都要经过字节码校验器校验,但是字节码校验器不检查那些可信任的编译器所生成的类文件,而是对那些有意违背命名空间规定和 Java 语言规则的类文件进行检查,并且关闭具有安全性漏洞的类文件。字节码校验器对程序代码进行四次校验,主要执行下面的检查。

(1) 加载的类是否符合 JVM 规范的类文件格式。

(2) 是否违反访问限制。

(3) 代码上是否造成上溢或者下溢。

(4) 所有操作代码的参数类型是否是正确的。

(5) 是否有非法的数据类型转换发生。

(6) 该类文件是否存在某些非法操作。

最后,如果程序代码通过了字节码校验器检验,Java 解释器负责把类文件解释成为机器码进行执行。

1.2 Java 的主要应用

目前 Java 有三个版本:Java SE(标准版)、Java EE(企业版)、Java ME(微型版)。这也就是 SunONE(Open Net Environment)体系。

人们比较习惯说 J2SE、J2EE 和 J2ME。其中的 2 代表 1.2 版本。但是现在 Java 语言在不断升级,2 已经没有意义了。从 JDK 1.5 以后,这种说法被 Java SE、Java EE、Java ME 所取代。

Java SE 就是 Java 的标准版,主要用于桌面应用程序的编程;Java EE 是 Java 的企业版,主要用于分布式的网络程序的开发,如电子商务网站和 ERP 系统;Java ME 是 Java 的微型版,主要应用于嵌入式系统开发,如手机和 PDA 的编程。

在学习 Java 技术时,针对不同的学习目的,学习的侧重点也有所不同。下面介绍不同的应用领域涉及的理论知识和应该掌握的内容。

1. Java SE 应用程序

Java SE 就是 Java Standard Edition(标准版),包含那些构成 Java 语言核心的类。比如:数据库连接、接口定义、输入输出、网络编程等。Java SE 定位在客户端程序的应用上。

如果开发 Java SE 的程序,就需要用 Java 语言编写完整的工程项目。前台有用户交互的界面,后台有数据库连接。这样就需要掌握 Java 语言基础、用户界面、高级编程,即本书的三个篇章都需要学习。

2. Java EE 应用基础

Java EE 就是 Java Enterprise Edition(企业版),包含 Java SE 中的类,并且还包含用于开发企业级应用的类,如 EJB、Servlet、JSP、XML、事务控制等。企业版在 Java SE 的基础上增加了企业内部扩展类函数库的支持,Java EE 定位在服务器端程序的应用上。

如果开发 Java EE 的程序,Java 语言在工程项目中的作用仅仅是搭建后台的控制层。而前台的用户交互界面,则由 MVC 结构中的视图层搭建,就是 HTML 和 JavaScript 等语言负责。所以只需要掌握 Java 语言基础、高级编程,即本书的基础篇和高级应用篇。

3. Java ME 应用基础

Java ME 就是 Java Micro Edition(微型版),用于消费类电子产品的软件开发。比如:智能卡、手机、PDA、机顶盒。Java ME 并不支持所有的 Java 基本类,只支持 Java 标准规范中所定义的核心类函数库的子集,并加入了一些支持嵌入式系统的扩展类函数库。Java ME 定位于嵌入式系统的应用上。

如果开发 Java ME 的程序,前台的用户交互界面和后台的数据库连接都由嵌入式平台提供。所以只需要掌握 Java 语言基础、多线程编程和网络编程,即本书的基础篇和高级应用篇的部分章节。

1.3 基本的 Java 应用程序

下面通过一个基本的应用程序的例子,说明 Java 编程语言的编写、执行、调试的过程。

1.3.1 程序说明

1. 编写 Java 程序

【例 1-1】 在记事本中编写应用程序,并保存程序名为 Example1_1.java 的文件。

```
public class Example1_1
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```