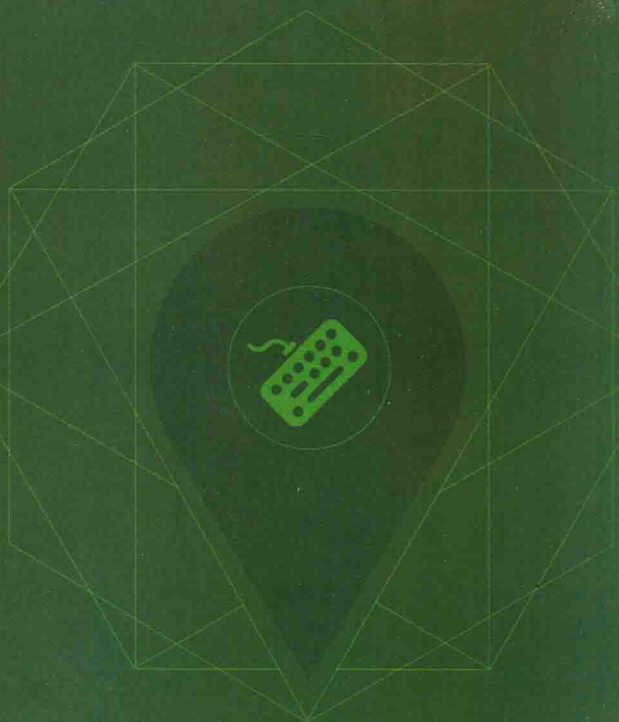




大学计算机规划教材

Java

大学实用教程 (第4版)

◆ 耿祥义 张跃平 编著

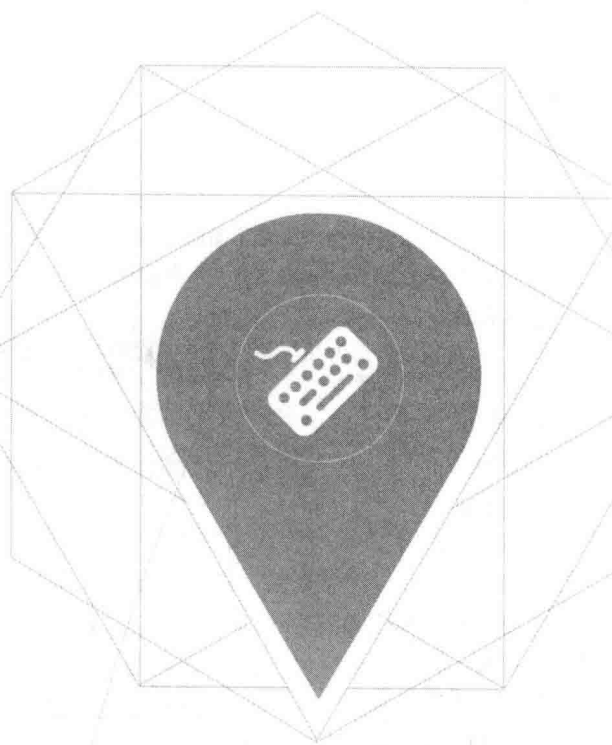
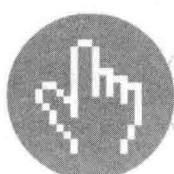
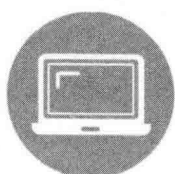
Java University Practical Guide
Fourth Edition



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



大学计算机规划教材

Java

大学实用教程 (第4版)

◆ 耿祥义 张跃平 编著

Java University Practical Guide
Fourth Edition

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书共 13 章, 重点讲解类与对象、类的继承、接口、泛型、字符串与模式匹配、实用类及数据结构、多线程、输入/输出流、图形用户界面设计、网络编程、数据库操作、Java Applet 程序设计等内容。本书注重可读性和实用性, 加大了面向对象的知识容量, 对部分例题的内容做了调整, 特别将第 12 章关于数据库的讲解更改为 Derby 数据库。本书提供配套的教学资源, 有配套的辅导书《Java 大学实用教程学习指导(第 4 版)》。

本书适合作为高等学校相关课程的教材, 也可以作为自学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

Java 大学实用教程 / 耿祥义, 张跃平编著. —4 版. —北京: 电子工业出版社, 2017.3

ISBN 978-7-121-31032-4

I. ① J… II. ① 耿… ② 张… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2017)第 043425 号

策划编辑: 章海涛

责任编辑: 章海涛

特约编辑: 何 雄

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 20.75 字数: 590 千字

版 次: 2005 年 3 月第 1 版

2017 年 3 月第 4 版

印 次: 2017 年 3 月第 1 次印刷

定 价: 46.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 192910558 (QQ 群)。

第 4 版前言

本书全面地讲解了 Java 的基础内容和编程方法,在内容的深度和广度方面都进行了仔细考虑,在类、对象、继承、接口等重要的基础知识的讲解上侧重深度,而在实用类的讲解上侧重广度。本书继续保留第 3 版的特点——注重教材的可读性和实用性。特别将第 12 章关于数据库的讲解更改为 Derby 数据库。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 在网络编程中的一些重要技术。

全书共 13 章。第 1 章主要介绍 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、3 章主要介绍 Java 的基本数据类型、运算符和控制语句。第 4、5 章是本书的重点内容,讲述类、对象、继承、接口以及 Java 语言新增的泛型等重要知识,特别讲述面向抽象的程序设计思想。第 6 章讲述常用的字符串和相关的模式匹配的知识,模式匹配问题是很多信息技术经常需要处理的问题之一,重点讲解使用 Scanner 类解析字符串的实用技术。第 7 章讲述常用的实用类,包括处理日期、数学计算、数字格式化以及数据结构等实用类。第 8 章讲述多线程技术,也是很难理解的一部分内容,通过许多有启发的例子来帮助读者理解多线程编程。第 9 章讲解 Java 中的输入/输出流技术,特别介绍怎样使用输入/输出流来克隆对象以及 Java 的文件锁技术。第 10 章是基于 SWING 的 GUI 设计,讲解常用的组件和容器,对于比较复杂的组件都给出了实用的例子。第 11 章讲解 Java 在网络编程中的一些重要技术,涉及 URL、Socket、InetAddresses、DatagramPacket、Broadcast 以及 Java 远程调用等重要的网络编程技术。第 12 章主要讲解 Java 怎样操作数据库,介绍预处理、事务处理、批处理等重要技术。第 13 章主要讲解 Java Applet 的运行原理以及在网络中的角色。

本书的例题全部在 JDK 1.6 环境下编译通过。每章都有问答题和作业题,通过回答问题可以使读者加深知识的理解。理解该章内容后,读者完全有能力独立地完成作业题。

本书配有配套的电子课件,请教师登录华信教育资源网站 <http://www.hxedu.com.cn> 下载(如果是第一次登录该网站,请先注册)。

本书配有上机实验指导,除了按照主教材的章节配备实验指导外,还配备综合实验。每章的实验指导由实验内容和知识扩展两部分内容组成,学生可按照实验的要求上机编写程序,每个实验都提供了程序模板,学生完成实验后需填写实验报告。知识扩展是对实验内容的补充,结合实例讲解主教材未能涉及的一些知识或已学知识的深入讨论。综合实验的目的是让读者综合运用所学知识来设计一个完整的软件。

作 者

作者简介

耿祥义,1995 年中国科学技术大学博士毕业,获理学博士学位,1997 年从中山大学博士后流动站出站。现任大连交通大学教授,具有多年从事 Java 语言教学经验,编写多部教材。

张跃平,现任大连交通大学讲师,具有多年从事 Java 语言教学经验。

目 录

第 1 章 Java 语言概述	1
1.1 Java 语言的诞生	1
1.2 学习 Java 的必要性	2
1.3 Java 的特点	3
1.4 Java 与 C/C++ 之关系	5
1.5 Java 运行平台	5
1.6 Java 程序开发	7
1.7 JDK 1.6 编译器的兼容性	10
问答题	10
作业题	10
第 2 章 基本数据类型和数组	11
2.1 标识符和关键字	11
2.2 基本数据类型	12
2.3 基本数据类型的转换	14
2.4 数据的输入和输出	15
2.5 数组	17
问答题	19
作业题	20
第 3 章 运算符、表达式和语句	21
3.1 算术运算符和算术表达式	21
3.2 关系运算符和关系表达式	22
3.3 逻辑运算符和逻辑表达式	23
3.4 赋值运算符和赋值表达式	23
3.5 移位运算符	23
3.6 位运算符	25
3.7 条件运算符	26
3.8 instanceof 运算符	26
3.9 一般表达式	26
3.10 语句概述	26
3.11 分支语句	27
3.12 循环语句	30
3.13 跳转语句	32
问答题	33

作业题.....	35
第4章 类和对象	36
4.1 面向对象编程.....	36
4.2 类声明和类体.....	39
4.3 类体的构成	40
4.4 构造方法和对象的创建	41
4.5 对象的引用和实体.....	45
4.6 成员变量	46
4.7 方法.....	48
4.8 方法重载	53
4.9 关键字 this.....	53
4.10 包.....	55
4.11 import 语句	56
4.12 访问权限	60
4.13 对象的组合	63
4.14 基本类型数据的类包装	66
4.15 对象数组	68
4.16 反编译和文档生成器	68
4.17 JAR 文件	69
问答题.....	70
作业题.....	71
第5章 继承、接口和泛型.....	72
5.1 子类和父类	72
5.2 子类的继承性.....	73
5.3 子类对象的构造过程.....	74
5.4 成员变量隐藏和方法重写	75
5.5 关键字 super.....	78
5.6 final 类和 final 方法	81
5.7 对象的上转型对象	81
5.8 继承和多态	83
5.9 抽象类	83
5.10 面向抽象	84
5.11 接口	87
5.12 接口回调	89
5.13 面向接口	90
5.14 抽象类与接口的比较	92
5.15 内部类	92

5.16 匿名类	93
5.17 异常类	95
5.18 泛型类	97
问答题	100
作业题	102
第 6 章 字符串和正则表达式	103
6.1 String 类	103
6.2 StringBuffer 类	109
6.3 StringTokenizer 类	110
6.4 正则表达式及字符串的替换和分解	111
6.5 Scanner 类	114
6.6 模式匹配	116
问答题	118
作业题	119
第 7 章 常用实用类	120
7.1 Date 类	120
7.2 Calendar 类	122
7.3 Math 类和 BigInteger 类	123
7.4 数字格式化	126
7.5 LinkedList<E>泛型类	129
7.6 HashSet<E>泛型类	133
7.7 HashMap<K,V>泛型类	135
7.8 TreeSet<E>泛型类	137
7.9 TreeMap<K, V>泛型类	139
7.10 Stack<E>泛型类	140
问答题	142
作业题	142
第 8 章 线程	143
8.1 Java 中的线程	144
8.2 线程的生命周期	144
8.3 线程的优先级和调度管理	148
8.4 Thread 的子类创建线程	149
8.5 Runnable 接口	150
8.6 线程的常用方法	155
8.7 线程同步	158
8.8 使用 wait()、notify()和 notifyAll()协调同步线程	159

8.9 挂起、恢复和终止线程	161
8.10 线程联合	163
8.11 守护线程	164
问答题	165
作业题	166
第 9 章 输入流和输出流	167
9.1 文件	168
9.2 文件字节流	170
9.3 文件字符流	172
9.4 缓冲流	173
9.5 数组流	176
9.6 字符串流	177
9.7 数据流	178
9.8 对象流	179
9.9 序列化和对象克隆	181
9.10 随机读写流	182
9.11 使用 Scanner 解析文件	185
9.12 文件锁	188
问答题	189
作业题	189
第 10 章 图形用户界面设计	190
10.1 AWT 组件与 SWING 组件概述	190
10.2 JFrame 窗体	191
10.3 菜单组件	192
10.4 布局设计	194
10.5 中间容器	199
10.6 文本组件	201
10.7 按钮与标签组件	208
10.8 复选框与单选按钮组件	210
10.9 列表组件	212
10.10 表格组件	213
10.11 树组件	215
10.12 进度条组件	217
10.13 组件常用方法	220
10.14 窗口事件	223
10.15 鼠标事件	224
10.16 焦点事件	229

10.17	键盘事件	229
10.18	AWT 线程	232
10.19	计时器	234
10.20	MVC 设计模式	235
10.21	播放音频	237
10.22	按钮绑定到键盘	239
10.23	对话框	241
10.24	多文档界面	247
10.25	发布应用程序	250
	问答题	251
	作业题	251
第 11 章	Java 中的网络编程	253
11.1	URL 类	254
11.2	读取 URL 中的资源	254
11.3	显示 URL 资源中的 HTML 文件	256
11.4	处理超链接	257
11.5	InetAddress 类	259
11.6	套接字 Socket	260
11.7	使用多线程处理套接字连接	263
11.8	UDP 数据报	266
11.9	广播数据报	270
11.10	Java 远程调用	273
	问答题	277
	作业题	277
第 12 章	数据库操作	278
12.1	JDBC 简介	278
12.2	Derby 数据库	279
12.3	在命令行连接 Derby 数据库	279
12.4	Java 程序连接 Derby 数据库	283
12.5	查询操作	284
12.6	更新、插入和删除操作	288
12.7	用结果集更新数据库中的表	289
12.8	CachedRowSetImpl 类	291
12.9	预处理语句	292
12.10	事务	295
12.11	批处理	296
12.12	使用 JTable 组件操作表	297

12.13 SQL Server 2000 数据库	302
12.14 使用纯 Java 数据库驱动	303
问答题	305
作业题	305
第 13 章 Java Applet	306
13.1 Java Applet 的运行原理	306
13.2 在 Java Applet 中播放音频	309
13.3 网页向 Java Applet 传值	310
13.4 在 Java Applet 中使用组件	310
13.5 在 Java Applet 中绘制图形	315
13.6 在 Java Applet 中绘制图像	316
13.7 在 Java Applet 中播放幻灯片	317
13.8 Java Applet 网络聊天室	318
问答题	322
作业题	322

第 1 章 Java 语言概述

本章导读

- ✧ Java 语言的诞生
- ✧ 学习 Java 的必要性
- ✧ Java 的特点及与 C/C++ 之关系
- ✧ Java 程序开发
- ✧ JDK 1.6 编译器的新规定

在学习 Java 语言之前,读者应当学习过 C 语言,熟悉计算机的一些基础知识。读者学习过 Java 语言之后,可以继续学习与 Java 相关的一些重要内容。比如,如果希望从事编写和数据库相关的软件,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 Java Micro Edition (Java ME);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和设计,可以学习 Java Enterprise Edition (Java EE),如图 1-1 所示。

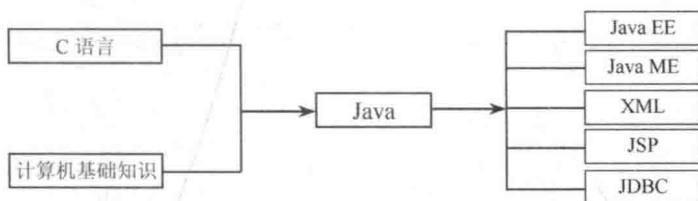


图 1-1 Java 的先导知识和后继技术

1.1 Java 语言的诞生

Java 诞生于 1995 年,是 Sun 公司组织开发的一门编程语言,主要贡献者是 James Gosling。开发 Java 语言的动力源于对独立于平台的需要,即这种语言编写的程序不会因为芯片的变化而发生无法运行或运行错误。当时,C 语言已无法满足人们的这一愿望,因为 C 语言总是针对特定的芯片将源程序编译为机器码,该机器码的运行与特定的芯片指令有关,在其他不同类型的芯片上可能无法运行或出现运行错误。芯片制造者、芯片使用者和软件编写者往往隶属于不同的公司。芯片制造者会不断地推出性能更好且价格更便宜的新型芯片,当有一种性价比更高的芯片出现时,芯片使用者就可能立即使用新的芯片,这些芯片可能安装在各种计算机或电子设备上,如果不能保证程序在新的芯片上正确运行,就可能出现难以发现的错误,

最终导致严重的后果，可能引起设备的毁坏等灾难性后果。所以，软件编写者必须针对新的芯片重新编译源程序，甚至需要对源程序进行必要的修改，这是令软件开发者最头痛的工作，还可能由于各种原因（如工作量的巨大等）而导致工作无法完成。

1990年，Sun公司成立了由James Gosling领导的开发小组，开始致力于开发一种可移植的、跨平台的语言，该语言能生成正确运行于各种操作系统、适应各种CPU芯片的代码。他们的精心钻研和努力促成了Java语言的诞生。Java的快速发展得益于Internet和Web的出现，Internet上有各种计算机，它们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。

James Gosling的办公室外面有一棵大橡树，他最初将Java语言命名为oak，后来发现已经有一种计算机语言的名字叫oak，最后决定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。“Java”是印度尼西亚一个盛产咖啡的岛屿，中文译名是“爪哇”。

1.2 学习Java的必要性

Java不仅可以用来开发大型的桌面应用程序，还特别适合Internet的应用开发。目前，Java语言不只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。Java语言面向对象编程，并涉及网络、多线程等重要的基础知识，而且很多新的技术领域都涉及Java语言。因此，学习和掌握Java已成为共识，国内外许多大学将Java语言列入了本科教学计划。IT行业对Java人才的需求正在不断增长，一些软件公司对其开发人员周期地进行Java的基础培训工作。在IT行业发达的北美洲，有将近60%的软件开发人员使用Java完成他们的工作，Evans Data公司在2002年做的一项调查中发现：在北美洲，Java的使用率已经接近C/C++。

2003年，James Gosling曾来北京，与中国的IT人士进行了交流，以下是对话的节选。

问：在近几年的发展过程中，很多编程语言都逐渐消失，Java语言却越来越火，请问其中的原因是什么？

James Gosling: 我认为，很多编程语言在发展中并不是消失，而是转移到了其他领域中，而Java的经久不衰取决于Java的技术基础。如果你问程序员，为什么会选择Java，他会告诉你，Java提供了多种功能，提供了方便的平台，是个足以吸引人的工具。我认为，推动Java最主要的因素是网络，Java是以网络应用为基础的开发工具，这是它的强项。

问：在传统计算机领域中，Java并不是十分大的平台，如PC。而在其他领域，如移动领域，Java发展迅速，Java的未来发展方向是什么？

James Gosling: 在PC领域，我并不认为Java不够强大。在PC领域，Java有很多应用，这是表面上看不到的，主要因为微软花了大力气避免用户看到，实际上Java应用很广泛，如人工智能游戏。在其他领域，Java更是应用广泛，如汽车、铁路机车上的即时控制系统，以及军用方面。

问：大家尊称您为Java之父，您能不能跟大家分享一下您在Java事业中最深的感受是什么？

James Gosling: 当看到Java的客户通过Java完成了很多神奇的工作，如看到夏威夷火山上的观测台使用Java控制望远镜，看到荷兰健康医疗组织使用Java解决了保护隐私问题等，那真是一种奇妙的感觉。

1.3 Java 的特点

1. 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由操作系统（OS）和处理器（CPU）构成。与平台无关是指软件的运行不因操作系统、处理器的变化导致程序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。注意，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，某平台可能用 8 位序列代码 1000 1111 表示一次加法操作，用 1010 0000 表示一次减法操作，另一平台可能用 8 位序列代码 1010 1010 表示一次加法操作，用 1001 0011 表示一次减法操作。

(2) C/C++程序依赖平台

现在来分析为何 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、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）。这样无法保证 C/C++ 编译器所产生的可执行文件在所有平台上都能正确地被运行，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。

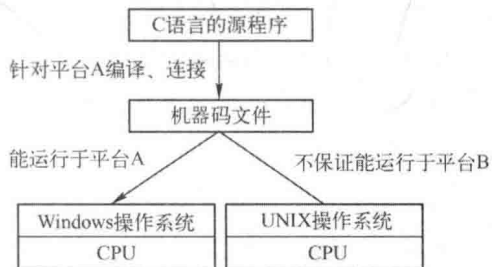


图 1-2 C/C++生成的机器码文件依赖平台

(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，这也是 Java 风靡全球的主要原因。Java 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）、类库及一些核心文件组成。Java 虚拟机的核心是字节码指令，即可以被 Java 虚拟机直接识别、执行的一种由 0、1 组成的序列代码。字节码并不是机器指令，因为它不与特定的平台相关，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，如所有的虚拟机都将 1111 0000 识别、执行为加法操作。

与 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 源程序编译为称为字节码的一种“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。

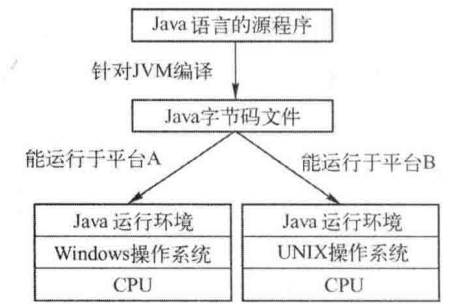


图 1-3 Java 生成的字节码文件不依赖平台

2. 面向对象

面向对象编程是一种先进的编程思想，更加容易解决复杂的问题。面向对象编程主要体现在下列三个特性。

① 封装。面向对象编程的核心思想之一就是数据和对数据的操作封装在一起。抽象即从具体的实例中抽取共同的性质形成一般的概念，如类的概念。人们经常谈到的机动车类就是从具体的实例中抽取共同的属性和功能形成的一个概念，那么一个具体的轿车就是机动车类的一个实例，即对象。一个对象将自己的数据和对这些数据的操作合理、有效地封装在一起，如每辆轿车调用“加大油门”改变的都是自己的运行速度。

② 继承。继承体现了一种先进的编程模式。子类可以继承父类的属性和功能，即继承了父类的数据和数据上的操作，又可以增加子类独有的数据和数据上的操作。比如，“人类”自然继承了“哺乳类”的属性和功能，同时增加了人类独有的属性和功能。

③ 多态。多态性是面向对象编程的又一重要特征。多态有两种。一种是操作名称的多态，即有多个操作具有相同的名字，但这些操作接收的消息类型必须不同。另一种是与继承有关的多态，是指同一个操作被不同类型调用时可能产生不同的行为。

Java 是面向对象的编程语言，本书将在第 4 章详细、准确地讨论类、对象、继承、多态、接口等重要概念。

3. 多线程

Java 的特点之一就是内置对多线程的支持。多线程允许同时完成多个任务，使人产生多个任务在同时执行的错觉。目前，计算机的处理器在同一时刻只能执行一个线程，但处理器可以在不同的线程之间快速切换，由于处理器速度非常快，远远超过了人们接收信息的速度，所以感觉好像多个任务在同时执行。C++没有内置的多线程机制，因此必须调用操作系统的多线程功能来进行多线程程序的设计。

4. 安全

当用户准备从网络上下载一个程序时，最大的担心是程序中含有恶意的代码，如试图读取或删除本地机上的一些重要文件，甚至该程序是一个病毒程序等。当用户使用支持 Java 的浏览器时，可以放心地运行 Java Applet（Java 小应用程序），不必担心病毒的感染和恶意的企图，Java Applet 将限制在 Java 运行环境中，不允许它访问计算机的其他部分。本书将在第 12 章详细讲述 Java Applet。

5. 动态

在学习了第 4 章后就会知道，Java 程序的基本组成单元就是类，有些类是用户自己编写的，有些是从类库中引入的。而类是运行时动态装载的，这就使得 Java 可以在分布环境中动态地维护程序及类库，而不像 C++那样，每当其类库升级之后，如果想让程序具有新类库提供的功能，就必须重新修改、编译程序。

1.4 Java 与 C/C++之关系

如果学习过 C++ 语言,读者会感觉 Java 很眼熟,因为 Java 中许多基本语句的语法与 C++ 类似,像常用的循环语句、控制语句等与 C++ 几乎一样,但不要误解为 Java 是 C++ 的增强版。Java 和 C++ 是两种完全不同的语言,它们各有各的优势,Java 语言和 C++ 语言已成为软件开发者应当掌握的语言。如果从语言的简单性方面看,Java 要比 C++ 语言简单,C++ 语言中许多容易混淆的概念或者被 Java 语言弃之不用,或者以一种更清楚、更容易理解的方式实现,如 Java 语言不再有指针的概念。Java 语言既易学又好用,但不要误解为这门语言很干瘪。读者可能很赞同这样的观点:英语要比阿拉伯语言容易学,但这并不意味着英语就不能表达丰富的内容和深刻的思想。

1.5 Java 运行平台

1. 三种平台简介

Sun 公司要实现“编写一次,到处运行 (write once, run anywhere)”的目标,就必须提供相应的 Java 运行平台。目前,Java 运行平台主要分为下列 3 个版本。

- ❖ Java SE (曾称为 J2SE)——Java 标准版或 Java 标准平台。Java SE 提供了标准的 JDK 开发平台,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。
- ❖ Java EE (曾称为 J2EE)——Java 企业版或 Java 企业平台,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。
- ❖ Java ME (曾称为 J2ME)——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。

登录 Sun 公司的网站 <http://java.sun.com>,就能看到有关 Java SE、Java EE 和 Java ME 的介绍。上述 Java 运行平台都包括了相应的 JVM, JVM 负责将字节码文件(包括程序使用的类库中的字节码)加载到内存中,然后采用解释方式来执行字节码文件,即根据相应硬件的机器指令翻译一句,执行一句。

2. 安装 Java SE 平台

学习 Java 应当从 Java SE 开始,因此本书基于 Java SE 来学习 Java。目前, Sun 公司已发布了 JDK 1.6,可以登录到其官网免费下载(如 `jdk-6u3-windows-i586-p.exe`),在“Popular Downloads”页面中选择“Java SE”→“JDK 6 Update”,单击“下载”即可。双击 `jdk-6u3-windows-i586-p.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。这里将默认的安装路径“`C:\program Files\Java\Jdk1.6.0_3`”修改为“`E:\jdk1.6`”,如图 1-4 所示。

注:在安装过程中会出现安装支持欧洲语言的 JRE (Java Runtime Environment) 的界面,在该界面上不必更改默认的安装路径,使用默认的安装路径即可。JRE 的安装路径不可以与 JDK 的安装路径相同。

将 JDK 安装到 E:\jdk1.6 目录下会生成如图 1-5 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(JRE)。

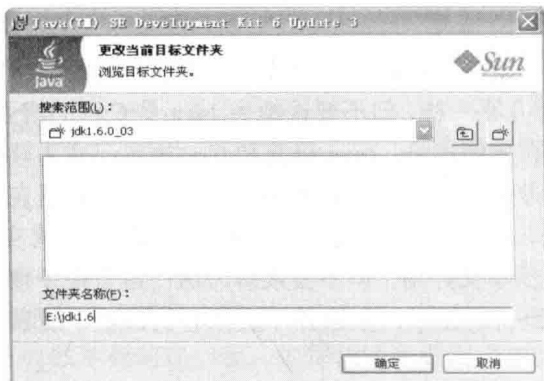


图 1-4 JDK 的安装路径

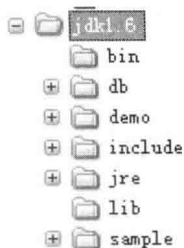


图 1-5 JDK 的目录结构

JDK 主要目录内容如下。

- ① 开发工具：位于 bin 子目录中，包括工具和实用程序，可以开发、执行、调试和保存用 Java 编程语言编写的程序。
- ② Java 运行环境：位于 jre 子目录中，由 JDK 使用的 JRE 实现。JRE 包括 Java 虚拟机 (JVM)、类库及其他支持执行用 Java 语言编写的程序的文件。
- ③ 附加库：位于 lib 子目录中，包括开发工具所需的其他类库和支持文件。
- ④ 演示 Applet 和应用程序：位于 demo 子目录中，Java 平台的编程示例（带源代码）。这些示例包括使用 SWING 和其他 Java 基类以及 Java 平台调试器体系结构的示例。
- ⑤ 样例代码：位于 sample 子目录中，某些 Java API 的编程样例（带源代码）。
- ⑥ C 头文件：位于 include 子目录中，支持使用 Java 本机界面、JVM 工具界面及 Java 平台的其他功能进行本机代码编程的头文件。
- ⑦ 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）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 子目录中，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 XP，右键单击“我的电脑”，在弹出的快捷菜单中选择“属性”，弹出“系统特性”对话框，再单击“高级选项”，然后单击“环境变量”按钮，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，添加需要的值，如图 1-6 所示。也可以在命令窗口（如 MS-DOS 窗口）中输入“path = E:\jdk1.6\bin;”。

（2）系统环境 classpath 的设置

JDK 的安装目录的 jre 子目录中包含 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。安装这些产品后，classpath 的值可能会被修改。那么运行 Java 应用程序时，读者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被找到，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 2000/2003/XP，右键单击“我的电脑”，在弹出的快捷菜单中选择“属性”，弹出“系统特性”对话框，再单击该对话框中的“高级选项”，然后单击“环境变量”按钮，添加如图 1-7 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，添加需要的值。

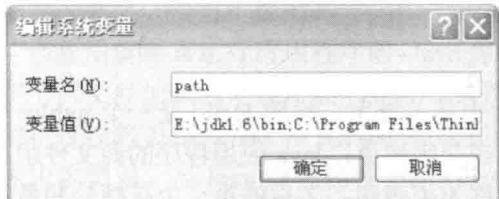


图 1-6 设置环境变量 path

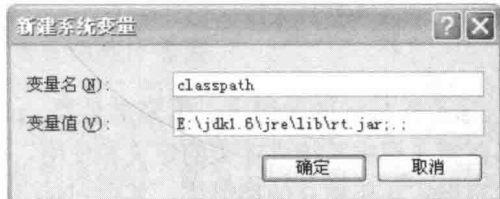


图 1-7 设置环境变量 classpath

对于 Windows 9x，用记事本编辑 `autoexec.bat` 文件，加入设置语句“`set classpth=E:\jdk1.6\jre\lib\rt.jar;.`”即可，也可在命令行窗口（如 MS-DOS）中输入“`set classpth=E:\jdk1.6\jre\lib\rt.jar;.`”。

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。

（3）仅仅安装 JRE

如果读者只想运行别人的 Java 程序，可以只安装 Java 运行环境 JRE。JRE 由 JVM、Java 的核心类及一些支持文件组成。读者可以登录 Sun 公司的官网免费下载 JRE。

（4）一些 IDE 开发工具

还有一些其他很好的 Java 程序 IDE 开发环境可用，包括来自 Sun、Borland、Symantec 公司的产品，如 Sun One、JBuilder 和 Eclipse 等，目前以 Eclipse 最流行。这些 IDE 产品都集成 JDK 作为主要部分。初学者应当使用 JDK 来开发 Java 程序，这不仅是学习 Java 最好的方式，还是掌握使用 IDE 开发工具的必要条件。IDE 开发环境适合于设计开发大型项目时使用，不适合于学习 Java 语言。

建议下载 Sun 公司的 Java 类库帮助文档，如 `jdk-6-doc.zip`。

1.6 Java 程序开发

用 Java 标准平台编译得到的字节码文件，可以在任何具有 Java 标准平台的计算机上正确地运行。开发一个 Java 应用程序需要经过三个步骤：编写源文件 → 编译源文件，生成字节码 → 加载运行字节码。

1. 编写源文件

（1）源文件的结构

应使用文本编辑器（如 Edit 或记事本）来编写源文件，不可使用 Word 编辑器，因它包含不可见字符。Java 是面向对象编程，Java 应用程序的源文件由若干个书写形式互相独立的类组成。

【例 1-1】 编写 3 个类：A，B 和 Hello。

```
class A {
    void f() {
        System.out.println("I am A");
    }
}
```