

“十三五”高等学校规划教材



Java面向对象 程序设计教程

Java MIANXIANG DUIXIANG CHENGXU SHEJI JIAOCHENG

张志斌 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十三五”高等学校规划教材

Java 面向对象程序设计教程

张志斌 主 编

张 明 祁建宏 郑明秋 副主编

朱小军 岳建斌 参 编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是针对 Java 语言初学者编写的基础教程, 不仅讲解了 Java 程序设计的基础知识, 而且提供了大量实用性很强的编程实例。全书共分 15 章, 内容包括: 初识 Java 语言、Java 语言基础、Java 语言程序结构、数组、字符串、Java 中的方法、类和对象、异常、Java 常用类库、Java 集合框架、Java 文件操作、Java 网络编程、Java 中的线程、AWT 与 Swing、虚拟机中的内存管理。另外, 还提供了 2 个附录, 介绍了正则表达式和 Java 中的反射机制, 便于学生进行 Java 语言的深入学习。

本书内容通俗易懂, 举例恰当生动, 适合作为高等学校相关专业面向对象程序设计课程的教材, 也可作为计算机行业从业人员和编程爱好者的参考用书。

图书在版编目 (CIP) 数据

Java 面向对象程序设计教程 / 张志斌主编. — 北京:
中国铁道出版社, 2017.3

“十三五”高等学校规划教材

ISBN 978-7-113-22845-3

I. ①J… II. ①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 (2017) 第 025813 号

书 名: Java 面向对象程序设计教程

作 者: 张志斌 主编

策 划: 潘晨曦

读者热线: (010) 63550836

责任编辑: 秦绪好 彭立辉

封面设计: 刘 颖

封面制作: 白 雪

责任校对: 张玉华

责任印制: 郭向伟

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: 北京尚品荣华印刷有限公司

版 次: 2017 年 3 月第 1 版 2017 年 3 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 15 字数: 359 千

书 号: ISBN 978-7-113-22845-3

定 价: 35.00 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 是一种跨平台的面向对象的程序设计语言，其前身是 Oak。Java 自面世后就非常流行，且发展迅速，具有卓越的通用性、高效性、健壮性、平台移植性和安全性，广泛应用于 PC、数据中心、游戏控制台、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专业社群。在全球云计算和移动互联网的产业环境下，Java 具备显著的优势和广阔的前景。

Java 语言的风格接近于 C++ 语言，但舍弃了 C++ 中的指针，改以引用取代（按内存地址传递），同时移除了 C++ 的运算符重载功能，移除了多重继承特性，改用接口取代，增加了垃圾回收器功能。

本书通俗易懂，简单明了，重点突出，既考虑了初次接触 Java 的初学者，又为有一定编程经验者提供了相应的指导。此外，用各种事例来阐明比较难懂或者易混淆的概念，学生可以边学边练，逐步加深和完善对核心技术的理解。

本书集合了数位多年教学一线教师的教学实例以及工作、研究经验编写而成，内容涵盖了最新的 Java 应用技术，具有可操作性、实践性和先进性。

本书从面向对象的编程技术着手，涉及图形用户界面、网络通信、网络编程、线程等先进的应用技术，适合作为高等学校相关专业面向对象程序课程的教材，也可作为计算机行业从业人员和编程爱好者的参考用书。

本书由张志斌任主编，张明、祁建宏、郑明秋任副主编，朱小军、岳建斌参与了编写。具体编写分工如下：第 1~6 章、第 14 章由张志斌编写；第 7~9 章由祁建宏编写；第 10、11 章由郑明秋编写；第 12、13 章由张明编写；第 15 章以及附录由岳建斌和朱小军编写。岳建斌和朱小军对书中程序进行了测试并审阅了全书。

由于时间仓促，编者水平有限，书中疏漏和不妥之处在所难免，恳请专家和读者提出宝贵意见。

编者

2016 年 12 月

第 1 章 初识 Java 语言	1
1.1 Java 语言背景.....	1
1.1.1 语言概述	1
1.1.2 平台概述	1
1.1.3 JDK 的概念和下载	2
1.2 Java 的优势和特点	3
1.3 JDK 的配置.....	3
1.4 JRE 的概念.....	5
1.5 程序语言的编译和解释	5
1.6 第一个 Java 程序	6
1.7 编译和运行第一个 Java 程序.....	7
1.8 Java 虚拟机与跨平台性	8
1.9 Java 程序打包.....	8
第 2 章 Java 语言基础.....	10
2.1 标识符	10
2.2 数据类型和变量.....	11
2.2.1 数据类型	11
2.2.2 变量	11
2.2.3 各种类型的变量	12
2.3 字面常量	14
2.4 算术运算和算术运算符	14
2.5 关系运算和关系运算符	18
2.6 逻辑运算和逻辑运算符	18
2.7 位运算和位运算符	19
2.7.1 按位与	19
2.7.2 按位或	19
2.7.3 按位异或	19
2.7.4 按位取反	19
2.8 赋值和赋值运算符.....	20
2.9 类型转换	21
2.9.1 自动类型转换.....	21
2.9.2 强制数据类型转换	22
2.10 运算符优先级.....	24



第3章	Java 语言程序结构	25
3.1	程序结构概述	25
3.1.1	顺序结构	25
3.1.2	分支结构	25
3.1.3	循环结构	25
3.2	Java 分支结构	26
3.2.1	if 语句	26
3.2.2	switch 语句	28
3.3	三元表达式	30
3.4	Java 循环结构	31
3.4.1	while 循环	31
3.4.2	do...while 循环	33
3.4.3	for 循环	34
3.5	循环控制	36
3.5.1	break 语句	36
3.5.2	continue 语句	36
3.6	分支与循环的嵌套	36
3.6.1	分支结构嵌套	37
3.6.2	循环结构嵌套	37
第4章	数组	39
4.1	数组的定义	39
4.2	一维数组	39
4.2.1	一维数组声明	40
4.2.2	一维数组的初始化	40
4.2.3	使用数组	43
4.3	多维数组	44
4.3.1	二维数组初始化	44
4.3.2	关于多维数组	47
4.4	数组应用	47
4.4.1	获取数组中所有元素的最大值	47
4.4.2	对数组进行从大到小排序	48
第5章	字符串	51
5.1	字符串概述	51
5.2	字符串的定义与基本操作	51
5.3	字符串的常用操作	52
5.4	数据类型小结	55
5.4.1	基本类型与引用类型	55
5.4.2	两种类型数据存储方式	55

5.4.3	基本类型的包装类	58
5.5	字符串类型与基本类型的转换	58
5.6	字符串的格式化输出	60
5.7	字符串的正则表达式概述	61
第 6 章	Java 中的方法	62
6.1	方法的定义	62
6.2	方法的使用	63
6.3	方法的返回值	65
6.4	方法的参数	66
第 7 章	类和对象	69
7.1	类和对象概述	69
7.2	类的定义	70
7.3	类的使用	71
7.4	局部变量与成员变量	73
7.5	静态成员和对象成员	74
7.5.1	静态属性	75
7.5.2	静态方法	76
7.6	构造方法	77
7.7	方法重载	78
7.8	Java 中的程序包	80
7.8.1	包的定义	80
7.8.2	系统包	80
7.8.3	import 关键字	81
7.9	面向对象中的封装	81
7.9.1	面向对象概述	81
7.9.2	封装	82
7.10	面向对象中的继承	83
7.10.1	extends 关键字	84
7.10.2	方法重写与 super 关键字	85
7.10.3	子类对象实例化过程	86
7.10.4	Object 类	87
7.11	面向对象中的多态	87
7.12	抽象类和接口	90
7.12.1	抽象类	90
7.12.2	接口	92
7.12.3	抽象类与接口的比较	92
7.13	final 修饰符	93
7.14	Java 中的枚举	93



第 8 章	异常	95
8.1	异常的概念	95
8.2	异常的类型	95
8.3	try...catch 关键字	96
8.3.1	基本语法	96
8.3.2	多重 catch 块	97
8.4	finally 关键字	98
8.5	运行时异常与非运行时异常	99
8.6	throws 关键字	99
8.7	throw 关键字	100
8.8	自定义异常	100
第 9 章	Java 常用类库	102
9.1	StringBuffer 类和 StringBuilder 类	102
9.2	Math 类	103
9.3	基本数据类型包装类	104
9.4	Class 类	105
9.4.1	类加载	106
9.4.2	Static 块	107
9.5	日期和时间	107
9.5.1	Date 类	107
9.5.2	Calendar 类	108
9.5.3	日期格式化	110
9.6	数字格式化	114
第 10 章	Java 集合框架	116
10.1	Collection 接口及其子接口	116
10.1.1	List 接口	117
10.1.2	Vector 类	117
10.1.3	ArrayList 类	118
10.1.4	LinkedList 类	121
10.1.5	Set 接口	122
10.1.6	HashSet 类	123
10.1.7	迭代器	125
10.2	泛型	126
10.3	foreach 循环	130
10.4	Map 接口及实现类	131
10.5	集合工具类	132
10.5.1	Arrays 类	132
10.5.2	Collections 类	133

第 11 章	Java 文件操作	134
11.1	文件	134
11.2	流	136
11.2.1	字节流	137
11.2.2	字符流	142
11.3	Properties 类	144
11.4	序列化与反序列化	145
第 12 章	Java 网络编程	147
12.1	网络编程基础	147
12.1.1	IP 协议	147
12.1.2	TCP 协议	148
12.1.3	IP 地址与端口号	148
12.1.4	UDP	148
12.1.5	InetAddress 对象定位主机	149
12.2	基于 TCP 协议的 Socket 编程	149
12.2.1	使用 ServerSocket 进行服务器端开发	150
12.2.2	使用 Socket 进行客户端开发	153
12.3	基于 UDP 的 Socket 编程	156
12.4	URL 与 URI	157
第 13 章	Java 中的线程	159
13.1	线程概述	159
13.1.1	基本概念	159
13.1.2	线程状态	160
13.1.3	守护线程	160
13.2	线程实现	161
13.2.1	通过继承 Thread 类创建线程类	161
13.2.2	实现 Runnable 接口	163
13.2.3	两种方式的比较	165
13.3	守护线程实现	165
13.4	线程安全	166
13.4.1	同步代码块	167
13.4.2	同步方法	169
13.4.3	线程间通信	169
13.5	定时器	174
第 14 章	AWT 与 Swing	175
14.1	AWT 概述	175
14.1.1	容器	176
14.1.2	组件	177

14.1.3	布局管理器.....	179
14.1.4	事件处理.....	184
14.1.5	AWT 常用组件.....	190
14.1.6	字体颜色.....	191
14.1.7	图形和绘制方法	191
14.2	Swing 概述	193
14.2.1	容器	194
14.2.2	组件	195
14.2.3	优点	195
14.3	Swing 中的容器组件.....	196
14.3.1	JFrame	196
14.3.2	JPanel	196
14.3.3	JScrollPane	198
14.4	SwingGUI 组件.....	199
14.4.1	JTable.....	199
14.4.2	JTree.....	201
14.5	菜单	204
14.6	对话框	205
14.6.1	JOptionPane.....	205
14.6.2	JDialog.....	207
14.7	Java 2D 绘制图形	207
第 15 章	虚拟机中的内存管理.....	209
15.1	Java 程序内存分配概述	209
15.2	堆和栈	210
15.3	方法的值传递和引用传递	213
15.4	垃圾回收机制.....	214
15.5	Runtime 类的使用	216
15.6	字符串和字符串池	217
附录 A	正则表达式	221
附录 B	Java 中的反射机制	225

第 1 章

初识 Java 语言

Java 语言与 C++ 语言一样,都属于面向对象语言。每种计算机语言都有它的优点和适用的领域,要想更好地了解某种语言,就需要了解它产生的原因、发展的历史,以及推动它发展的动力。本章将讲述 Java 语言的发展历史、开发环境的搭建,以及它的优势,并动手编写第一个 Java 程序。

1.1 Java 语言背景

1.1.1 语言概述

1991 年 4 月, Sun 公司(已于 2009 年被 Oracle 公司收购)的 James Gosling 领导的绿色计划(Green Project)开始着力发展一种分布式系统结构,使用这种技术可以把 E-mail 发送给电冰箱、电视机等家用电器,并对它们进行控制或与它们进行信息交流。由于 Green 项目组的成员都具有 C++ 背景,一开始他们使用 C++ 语言来完成这个项目,所以首先把目光锁定了 C++ 编译器,但很快他们就感到 C++ 的很多不足,于是他们使用 C++ 开发了一种新的语言 Oak(Java 的前身)。Internet 的出现,给 Java 语言的发展提供了契机,当时,Mark Andreessen 开发的 Mosaic 和 NetScape 启发了 Oak 项目组成员,他们用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席运行官 Scott McNealy 的支持,并推动 Java 进军 Internet。关于 Java 的取名有一个趣闻,有一天,几位 Java 成员组的会员正在讨论这个新的语言取什么名字,当时他们正在咖啡馆喝着名为 Java(爪哇)的咖啡,有人就建议叫作 Java,这个提议得到了其他的高度赞赏,于是 Java 这个名字就这样被传开了。现在人们看到 Java 的 Logo 就是一杯冒着热气的咖啡,如图 1-1 所示。



图 1-1 Java Logo

Java 总是和 C++ 联系在一起,而 C++ 则是从 C 语言派生而来的,所以 Java 语言继承了这两种语言的大部分特性。Java 的语法是从 C 继承的,Java 许多面向对象的特性受到 C++ 的影响,但是学习 Java 语言之前,完全不必先去了解 C 或者 C++。

1.1.2 平台概述

Sun 公司将 Java 语言设计为可以针对移动平台、桌面系统、企业级应用进行开发的综合平台,极大地提高了 Java 语言的生产力。当掌握了 Java 语言的基本语言特性后,再通过学习特定的开发包,就可以开发移动应用程序(如手机游戏)、桌面应用程序(如人们熟知的 QQ、MP3 播放器都



属于桌面应用程序)和企业级的高级应用程序。现在,Java 语言在这 3 种平台的应用开发中,都占据了举足轻重的地位,Sun 公司将 3 种平台下的开发分别命名为 Java ME、Java SE 和 Java EE,它们是 Java 语言开发的 3 个应用领域。

(1) Java SE (Java Standard Edition): Java 标准版本,对应于桌面开发,可以开发基于控制台或图形界面的应用程序。Java SE 中包括了 Java 的基础类库,也是进一步学习其他两个分支的基础。本书主要学习的内容就是 Java SE 中基于控制台的应用程序开发。

(2) Java ME (Java Mobile Edition): Java 移动版本,对应于移动平台(如手机、PDA 等设备)的开发,因为这类设备的硬件差异很大,而 Java 恰恰具有平台无关的特性,同样的 Java 代码可以在不同的设备上运行,所以在移动平台开发中,Java ME 非常流行。从技术角度上可以认为 Java ME 是经过改变的 Java SE 的精简版。

(3) Java EE (Java Enterprise Edition): Java 企业版本,对应于企业级开发,包括 B/S 架构开发、分布式开发、Web 服务等非常丰富的应用内容,在软件开发企业中被大量应用。

1.1.3 JDK 的概念和下载

Sun 公司提供了自己的一套 Java 开发环境,称为 JDK (Java Development Kit)。Sun 提供了多种操作系统下的 JDK,版本不断升级,如:JDK1.2、JDK1.3、JDK1.4、JDK1.5、JDK1.6 等。

JDK 是整个 Java 的核心,包括了 Java 运行环境 (Java Runtime Environment)、各种 Java 工具和 Java 基础的类库 (rt.jar)。

提示: 在 2004 年 10 月, Sun 发布了 JDK1.5 版本,它加入了泛型、枚举、annotation 等特性,使得 Java 编程更加方便,也许是为了纪念这次重大的革新, Sun 公司使用了 Java 5 这个名字。

用户可以到 Sun 的网站下载 JDK,地址为 <http://java.sun.com> (因为 2009 年 4 月 Oracle 宣布收购了 Sun 公司,当输入了这个地址之后,会发现跳转到了 Oracle 公司的网站上)。在下载之前需要确定在哪个操作系统中使用,比如在 32 位 Windows 操作系统中使用,就应该下载与操作系统对应的 JDK 版本,这里下载的是 1.6 版本。选择平台界面如图 1-2 所示, JDK 下载界面如图 1-3 所示。

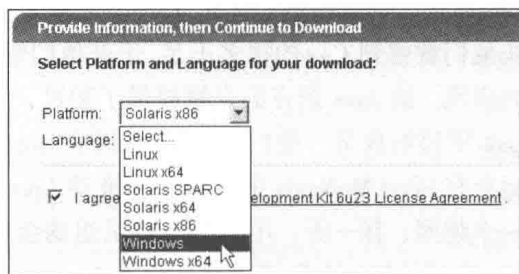


图 1-2 选择使用平台

双击下载后的 exe 文件就可以进行安装。JDK 安装完毕之后,得到如图 1-4 所示的目录结构。

关于 JDK 如何配置,如何使用 JDK 进行开发,请参考本章后续内容。JDK 中主要的目录内容如下:

(1) bin 目录:存放各种命令文件,在编译 Java 程序、运行 Java 程序时需要用到该目录中的文件。比如,编译 Java 程序就需要用到 javac 命令,运行 Java 程序就需要用到 java 命令。

(2) demo 目录:使用 Java 语言编写的一些示例程序。

(3) jre: Java 运行环境。Java 程序运行必需的运行环境,参见 1.4 节的内容。

(4) lib: Java 程序中用到的一些类库。初学阶段不必知道什么是类库,后续课程中会讲到。

(5) src.zip: 类库的源代码。

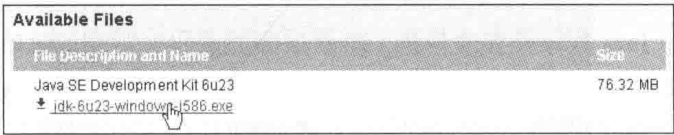


图 1-3 下载 Windows 使用的 JDK

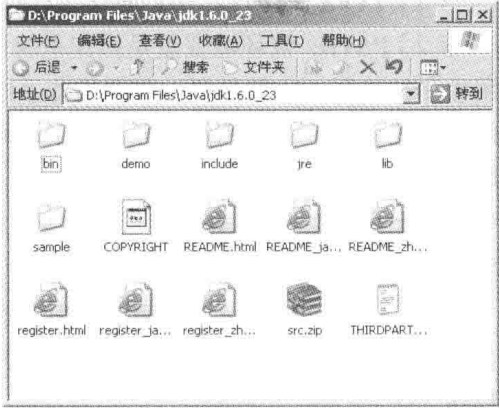


图 1-4 JDK 目录结构

1.2 Java 的优势和特点

经过十余年的发展，如今的 Java 语言比当初语言特性更完善、应用面更全、运行效率更高。从技术角度讲，Java 语言的重要特点如下：

- （1）面向对象：继面向过程编程之后，近年来最主流的编程方式就是面向对象编程。面向对象的设计和编程方式特别适用于更复杂、更庞大的应用软件开发。
- （2）健壮性与安全性：Java 提供了完善的内存管理机制，开发者可以通过简单的方式使用内存空间并有效地避免内存溢出。Java 同时内置了安全机制，能够有效地控制应用程序的访问权限，在网络开发环境中，此特性为开发可靠的企业级应用程序提供了保障。
- （3）高性能：通过优化的运行机制 Java 可以提供不逊于其他语言的运行速度，并可以通过调用原生代码的方式提高关键程序的性能。
- （4）平台无关：平台无关特性可以让 Java 程序运行在不同的软硬件或网络环境中，代码不经修改可以发布到不同的平台，极大地增强了软件的生命力和投资价值。
- （5）多线程开发：Java 支持多线程开发并提供了完善的并发访问控制，多线程的应用可以提高程序的性能，充分利用硬件资源（如在多核 CPU 的硬件环境下）。
- （6）分布式应用：有些复杂的应用程序系统，单台计算机难以满足需求，这时需要将程序发布到多台计算机上共同计算，这种应用称为分布式应用。Java 程序可以进行此类分布式的软件开发。

1.3 JDK 的配置

应用软件都是运行在操作系统中的，操作系统就是一个大管家，它管理所有运行的软件、硬件。软件需要在操作系统中“注册”，操作系统才能识别这个软件。比如，在机器上安装了 QQ，双击桌面上的图标，QQ 就能够运行，这是因为安装了 QQ 后，相当于 QQ 程序在操作系统中注册了。

同样的道理，安装 JDK 之后，还需要经过一系列的配置才能正确地运行 Java 程序。JDK 的配置涉及 3 个环境变量：

(1) JAVA_HOME: JDK 的安装目录。

(2) PATH: 该环境变量是操作系统固有的环境变量,作用是设置供操作系统去寻找和运行应用程序的路径。也就是说,如果操作系统要运行某个命令,会按照 PATH 环境变量指定的目录去依次查找,以最先找到的为准,由于该环境变量中可能配置多个路径,在 Windows 中,同一个环境变量的多个不同的路径之间使用分号(;)隔开,比如要编译一个 Java 程序,需要用到 javac 命令,这个命令在 JDK 安装目录的/bin 目录中,所以要在 PATH 环境变量中添加一个 JAVA_HOME 下的 bin 目录。

(3) CLASSPATH: 作用和 PATH 类似,Java 程序是在虚拟机(本章后面的课程有对虚拟机的介绍)中运行的,Java 虚拟机按照 CLASSPATH 环境变量指定的目录顺序去查找可以运行的 Java 程序。

假设 JDK 的安装目录为 D:\Program Files\Java\jdk1.6.0_23,环境变量配置如下:

```

JAVA_HOME= D:\Program Files\Java\jdk1.6.0_23
PATH=操作系统中原来的值; %JAVA_HOME%\bin
CLASSPATH=.;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar

```

提示: CLASSPATH 变量设置等号后面第一个字符是“.”表示当前目录,也就是首先到当前目录中搜索。%JAVA_HOME%是前面定义的 JAVA_HOME 变量的引用,其中的值为 D:\Program Files\Java\jdk1.6.0_23,它与后面的 \lib\tools.jar 组成一个完整的路径,即 D:\Program Files\Java\jdk1.6.0_23\dt.jar。

具体的配置过程如下:

(1) 右击“我的电脑”,选择“属性”命令,选择“高级”选项卡,如图 1-5 所示。

(2) 单击“环境变量”,在弹出窗口中的系统环境变量中单击“新建环境变量”,配置 JAVA_HOME 环境变量,如图 1-6 所示。

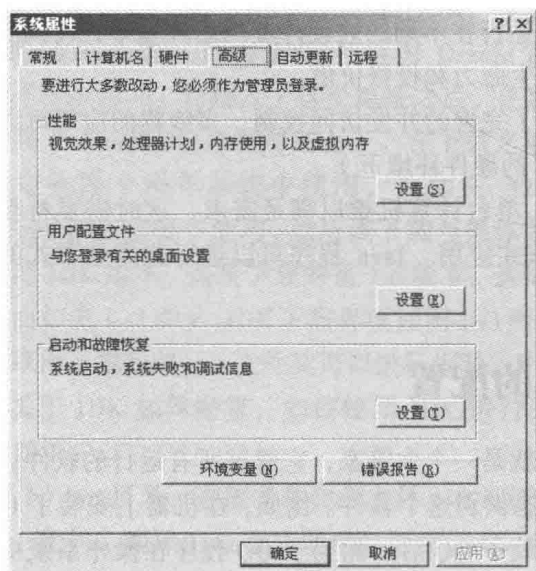


图 1-5 “高级”选择卡

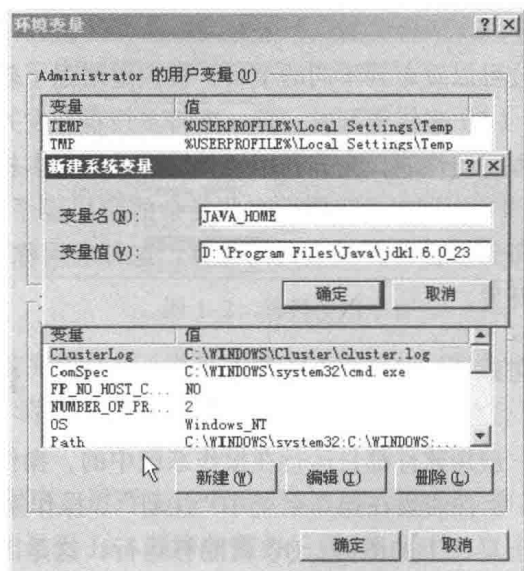


图 1-6 配置 JAVA_HOME 环境变量

(3) 找到“系统变量”中的 Path (见图 1-7),选中后单击“编辑”按钮,在弹出的对话框中输入变量名和变量值,如图 1-8 所示。

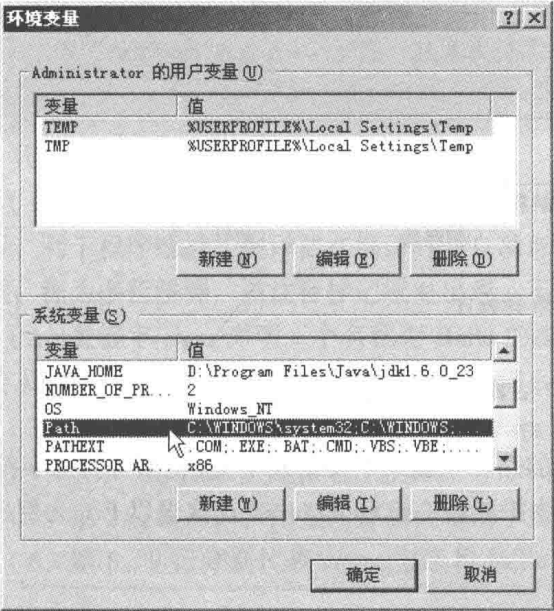


图 1-7 系统中的 path 环境变量

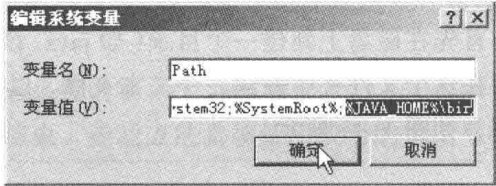


图 1-8 设置 Path 环境变量

提示：不要删除原有的内容，因为原来的内容在其他软件运行时需要用到。

(4) 按照新建 JAVA_HOME 的方式就可以新建 CLASSPATH 环境变量。

1.4 JRE 的概念

JRE (Java RunTime Environment, Java 运行时环境) 是用来运行、测试和传输应用程序的 Java 平台。它包括 Java 虚拟机、Java 平台核心类和支持文件。它不包含开发工具，如编译器、调试器和其他工具。

当编写 Java 程序时需要 JDK，JDK 中就包含了一个 JRE。如果只是要运行 Java 程序，直接下载并安装 JRE 即可。

1.5 程序语言的编译和解释

计算机并不能直接地接受和运行用高级语言编写的源程序，源程序必须通过“翻译程序”翻译成计算机所能够理解的可运行的目标程序（由 0、1 组成的二进制程序），计算机才能识别和运行。这种“翻译”通常有两种方式：编译方式和解释方式。

编译方式是指利用事先编好的称为编译程序的机器语言程序进行“翻译”。当用户将用高级语言编写的源程序输入计算机后，编译程序便把源程序整个翻译成用机器语言表示的目标程序，然后计算机再运行该目标程序，以完成源程序要处理的运算并取得结果，如图 1-9 所示。

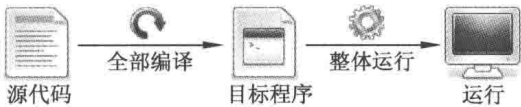


图 1-9 编译程序

解释方式是指源程序进入计算机后,解释程序边扫描边解释,逐句输入逐句翻译,计算机逐句运行,并不产生目标程序,如图 1-10 所示。

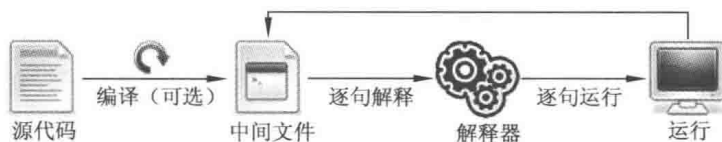


图 1-10 解释程序

1.6 第一个 Java 程序

首先在磁盘上新建一个目录(如 java,该目录的路径中建议不要出现中文,空格等特殊字符),然后在这个文件夹中新建一个文本文件,以.java 为扩展名(如 Hello.java)。这里以 E:\ 为例进行说明,如图 1-11 所示。

注意: 文件的扩展名是 .java,而有时候用户看到的扩展名虽然是 java,但实际上不是,这是因为操作系统的默认设置中隐藏了常见的文件扩展名。此时,需要去掉这个设置才能看到真实的扩展名,操作步骤如下:

(1) 选择 E 盘中的 java 文件夹,选择“工具”→“文件夹选项”命令,如图 1-11 所示。

(2) 选择“查看”选项卡,取消选择“隐藏已知文件类型的扩展名”复选框,如图 1-12 所示。



图 1-11 选择“文件夹选项”命令

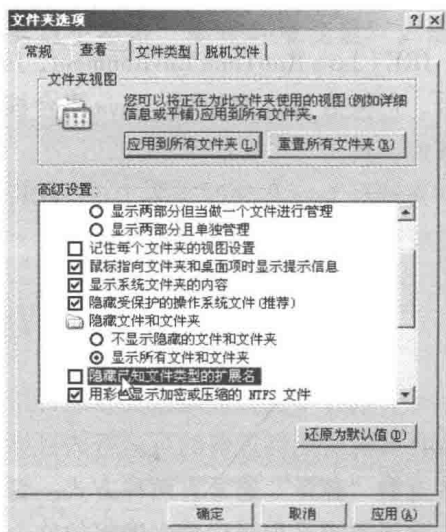


图 1-12 取消选择“隐藏已知文件类型的扩展名”复选框

使用文本编辑器打开 Hello.java 文件,在其中输入如下代码:

```

/**
 * 我的第一个 Java 程序
 */
public class Hello {
    public static void main(String[] args){
  
```

```
//简单的输出结果
System.out.println("这是我的第一个 Java 程序");
}
}
```

代码说明:

(1) 第 1~3 行代码为 Java 中的多行注释。注释是 Java 中的特殊语句,是为了对程序员进行提示,利于程序维护,编译器在编译时会自动忽略注释内容。这里的注释以“/*”开头,以“*/”结尾,称为多行注释。该注释标记成对出现,不能嵌套。

(2) 关键字 `class` 声明一个名称为 `Hello` 的类,并且使用关键字 `public` 将其修饰为公有。Java 中要求一个源文件中如果存在 `public` 的类,那么这个 `public` 类的名称和 Java 源文件名必须相同。

(3) 此行定义 `main()` 方法,`main()` 方法是 Java 程序运行的入口,也就是说程序从这里开始运行。

(4) “//”在 Java 中代表单行注释。

(5) 第 7 行语句的作用是向控制台输出字符串“这是我的第一个 Java 程序”。

(6) 第 8、9 行分别代表 `main()` 方法和 `Hello` 类的结束,类或方法必须使用成对的大括号将其内容括起来。

提示: 在编辑 Java 程序时,需要注意关键字的大小写,Java 程序严格区分大小写。

1.7 编译和运行第一个 Java 程序

前面已经写好了一个 Java 程序源代码,但此时 Java 程序还不能运行。要想使计算机能够运行 Java 程序,需要将 Java 源代码转换成计算机所能够理解的二进制码,这个过程就是“编译”,Java 的源代码经过编译之后,会生成一个以 `class` 结尾的文件。

那么如何进行编译呢?这时就需要用到 JDK 的 `bin` 目录中的 `javac.exe` 命令,前面在环境变量中已经设置了它存放的位置,这时可以在 DOS 窗口中直接使用这个命令。

选择“开始”→“运行”命令,输入 `cmd` 后打开 DOS 窗口,定位到 `E:\java` 目录中,分别输入 `javac Hello.java` 对源文件进行编译,输入 `java Hello` 运行 Java 程序,如图 1-13 所示。

在 Windows 中查看这个目录,发现此时多了一个文件, `Hello.class` 文件,这就是编译以后的字节文件,如图 1-14 所示。

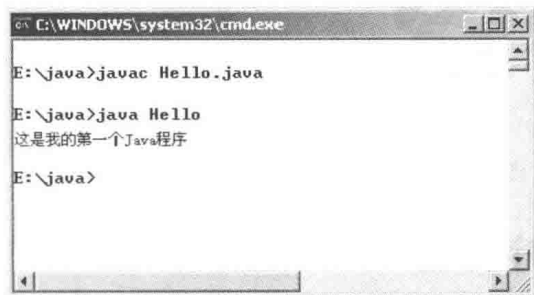


图 1-13 编译和运行 Java 程序



图 1-14 编译后形成字节码文件

源代码经过编译以后,就可以运行 Java 程序。与在 Windows 中运行程序不同,运行 Java 程