

21世纪高职高专规划教材

计算机应用系列

Java应用案例教程

张红梅 主编 姜晓刚 吉雪芸 副主编



清华大学出版社

21 世纪高职高专规划教材

计算机应用系列

Java应用案例教程

张红梅 主编

姜晓刚 吉雪芸 副主编

清华大学出版社
北 京

内 容 简 介

本书是 2009 年度教育部教指委精品课程(见学校网站)、山东省精品课程的配套教材。

全书着重强调应用与基础相结合,通过导入案例、理论讲解的方式,将相关知识点融入案例之中。读者通过完成案例,逐步掌握 Java 语言的具体应用。同时,本书通过大量的例题、实训,来强化读者的 Java 语言应用能力。本书共分 9 个模块,主要内容包括搭建 Java 开发环境、Java 开发基础、Java 面向对象编程、图形用户界面、集合框架的使用、输入输出、数据库编程、Java 多线程开发以及网络通信。

本书可作为专科及本科应用型、技能型人才培养使用的教材,也可供从事 Java 开发和应用的相关人员学习和参考使用。

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 应用案例教程/张红梅主编. —北京:清华大学出版社,2010.1

21 世纪高职高专规划教材. 计算机应用系列

ISBN 978-7-302-21495-3

I. ①J… II. ①张… III. ①JAVA 语言—程序设计—高等学校:技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2009)第 214069 号

责任编辑:张龙卿

责任校对:刘 静

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 刷 者:北京密云胶印厂

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:185×260 印 张:19.25 字 数:464 千字

版 次:2010 年 1 月第 1 版 印 次:2010 年 1 月第 1 次印刷

印 数:1~4000

定 价:33.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:031108-01



前言

Java 语言是美国 SUN 公司开发的一种完全面向对象的程序设计语言,从 1995 年 5 月诞生至今,随着 Internet 的迅猛发展而成长壮大,现已成为网络上的主流编程语言。在商业、通信和工业等很多领域得到广泛的应用。

Java 语言作为一种完全面向对象的程序设计语言,很好地支持和采用了面向对象设计的思想和精髓。因此采用 Java 语言作为面向对象教学语言,可以浅显、快速、透彻地培养读者面向对象程序设计的思想。

编者针对国家级精品课程的建设要求,以及更好地体现工学结合的教学思想,在内容编排上,本书通过经典的实际案例来讲述面向对象和 Java 编程的基本思想、方法和技术。各个模块都配有与案例知识相应的实训项目,使读者通过项目的开发,巩固相关的内容,并掌握实际编程技能。教材内容以最新的 Java 版本为背景,紧跟 Java 技术发展潮流。

本书的特色体现在:

(1) 选择经典案例导入

本书尽可能选用读者所接触到的现实世界的实例进行导入,并详细给出实现步骤和运行结果,以提高读者的学习兴趣。同时,对一些综合性比较强的模块,所选用的案例均来自于实际应用,这样读者就可以把学习的内容快速转化为自己的实际编程经验。

(2) 根据实际需求选取内容

本书根据 Java 企业应用情况,去掉了传统教材中的 Applet 和 AWT 这两部分内容,而着重介绍了 Application 和 Swing 的应用;在数据库编程部分,着重讲述了如何使用流行的 JDBC 技术连接数据库,并介绍了数据库应用程序的设计方法,从而可以真正实现“学有所用”。

(3) 注重面向对象思想的培养

由于面向对象程序设计思想的建立是一个逐步完善的过程,从人的认知规律出发,本书在编排上遵循“从简单到复杂”、“从抽象到具体”的原则,较好地将面向对象程序设计思想通过层层拓展的方式展现给读者。

本书建议学时为 128 学时,其中实践教学建议 78 学时,具体学时分配见学识分配表。

本书由张红梅担任主编,姜晓刚、吉雪芸担任副主编。参与本书编写的教师均具有丰富的 Java 应用开发及 Java 课程教学经验。其中,第 1、3、5、7、9 模块由张红梅编写,第 4、6 模块由姜晓刚编写,第 2、8 模块由吉雪芸编写。全书由张红梅负责规划与统稿。

本书所有的实例均在 JDK1.6 环境下调试通过,需要实例源代码的读者可与编者联系:zhmgmz@163.com。

由于编者水平有限,编写时间仓促,书中若有不妥之处,欢迎广大读者批评指正。



学时分配表

序 号	授 课 内 容	学 时 分 配	
		理 论	实 践
1	搭建 Java 开发环境	2	2
2	Java 开发基础	6	10
3	Java 面向对象编程	8	12
4	Java 图形用户界面开发	6	6
5	集合框架的使用	8	16
6	Java 输入/输出	6	10
7	Java 数据库编程	6	8
8	Java 多线程开发	4	8
9	网络通信	4	6
合 计		50	78

编 者

2009 年 10 月

目 录



模块 1 搭建 Java 开发环境	1
1.1 相关实践知识	1
1.1.1 JDK 的安装	1
1.1.2 配置环境变量	3
1.1.3 运行第一个 Java 程序	4
1.2 相关理论知识	5
1.2.1 Java 概述	5
1.2.2 Java 语言的特点	6
1.2.3 Java 程序类型	7
1.2.4 Java 语言实现机制	8
1.2.5 Java 集成开发工具的比较和选择	9
1.3 实训	11
1.4 习题	11
模块 2 Java 开发基础	13
2.1 案例一 打印出水仙花数	13
2.1.1 导入案例	13
2.1.2 相关理论知识	14
2.1.3 打印结果	17
2.1.4 实训	19
2.1.5 习题	19
2.2 案例二 输出学生成绩的等级	20
2.2.1 导入案例	20
2.2.2 相关理论知识	22
2.2.3 Java 中的数组	36
2.2.4 实训	39
2.2.5 习题	43
模块 3 Java 面向对象编程	44
3.1 案例一 人员信息输出	44



3.1.1	导入案例	44
3.1.2	相关理论知识	47
3.1.3	实训	59
3.1.4	习题	59
3.2	案例二 学生信息输出管理	60
3.2.1	导入案例	60
3.2.2	相关理论知识	62
3.2.3	实训	71
3.2.4	习题	72
3.3	案例三 学生信息综合管理	72
3.3.1	导入案例	73
3.3.2	相关理论知识	75
3.3.3	实训	81
3.3.4	习题	82
模块4 Java 图形用户界面开发		83
4.1	案例一 用户登录界面	83
4.1.1	导入案例	83
4.1.2	相关实践知识	85
4.1.3	相关理论知识	94
4.1.4	实训	107
4.1.5	习题	108
4.2	案例二 学生信息添加程序	108
4.2.1	导入案例	108
4.2.2	相关理论知识	116
4.2.3	实训	127
4.2.4	习题	128
4.3	案例三 简单计算器	129
4.3.1	导入案例	129
4.3.2	相关实践知识	137
4.3.3	相关理论知识	142
4.3.4	实训	149
4.3.5	习题	149
4.4	案例四 JTable 的应用	149
4.4.1	导入案例	150
4.4.2	相关理论知识	162
4.4.3	实训	166
4.4.4	习题	172



模块 5 集合框架的使用	174
5.1 导入案例：电话记事本	174
5.2 相关理论知识	177
5.2.1 集合	177
5.2.2 对集合元素的访问	182
5.2.3 映射(Map)	185
5.2.4 Collections 类	188
5.3 实训	189
5.4 习题	189
模块 6 Java 输入/输出	190
6.1 案例一 账户数据管理	190
6.1.1 导入案例	190
6.1.2 相关理论知识	198
6.1.3 实训	207
6.1.4 习题	212
6.2 案例二 学生数据管理	212
6.2.1 导入案例	212
6.2.2 相关理论知识	220
6.2.3 实训	227
6.2.4 习题	228
模块 7 Java 数据库编程	229
7.1 案例一 Java 连接 SQL Server 数据库	229
7.1.1 导入案例	229
7.1.2 相关理论知识	232
7.1.3 实训	240
7.1.4 习题	240
7.2 案例二 员工信息查询系统	240
7.2.1 导入案例	240
7.2.2 相关理论知识	242
7.2.3 实训	250
7.2.4 习题	250
模块 8 Java 多线程开发	251
8.1 案例一 生产者和消费者的问题	251
8.1.1 导入案例	251
8.1.2 相关理论知识	255
8.1.3 实训	265



8.1.4 习题	268
8.2 案例二 观察者和报告者问题	269
8.2.1 导入案例	269
8.2.2 相关理论知识	274
8.2.3 实训	278
8.2.4 习题	279
模块 9 网络通信	282
9.1 导入案例：点对点通信	282
9.2 相关理论知识	286
9.2.1 使用 URL 访问网络资源	286
9.2.2 InetAddress 类	289
9.2.3 Socket 通信	290
9.3 实训	296
9.4 习题	296



模块1 搭建Java开发环境

教学目标

- 能够搭建 Java 的开发环境。
- 了解 Java 语言产生的原因和应用领域。
- 掌握 Java 语言的特点。
- 会选择 Java 开发工具。

1.1 相关实践知识

1.1.1 JDK 的安装

1. 安装环境要求

Java 对计算机的要求不高,以下是基于 Windows 平台的计算机最低要求。

硬件要求: CPU PⅡ 以上,64MB 内存,100MB 硬盘空间。

软件要求: Windows 98/NT/2000/XP,Internet Explorer 5.0。

2. 下载 JDK

(1) 打开浏览器,在“地址”栏中输入网址 <http://java.sun.com/javase/downloads/index.jsp>,按 Enter 键,打开 Java 官方网站 JDK 下载主页面,如图 1-1 所示。

(2) 单击在“JDK 6 Update 14”版本右面的 Download 按钮,出现如图 1-2 所示的页面,在 Platform 下拉列表中选择 Windows,在 Language 下拉列表中选择 Multi-language,选中同意协议的复选框,单击 Continue 按钮。

(3) 在新打开的页面中选择“Java SE Development Kit 6u14” jdk-6u14-windows-i586.exe”进行下载即可,大小为 73.48MB。

3. 安装 JDK

运行 jdk-6u14-windows-i586.exe 来安装 JDK,安装过程中可以设置安装路径并选择组件,系统默认安装路径为 C:\Program Files\Java\jdk1.6.0_14,默认“组件选择”是全部

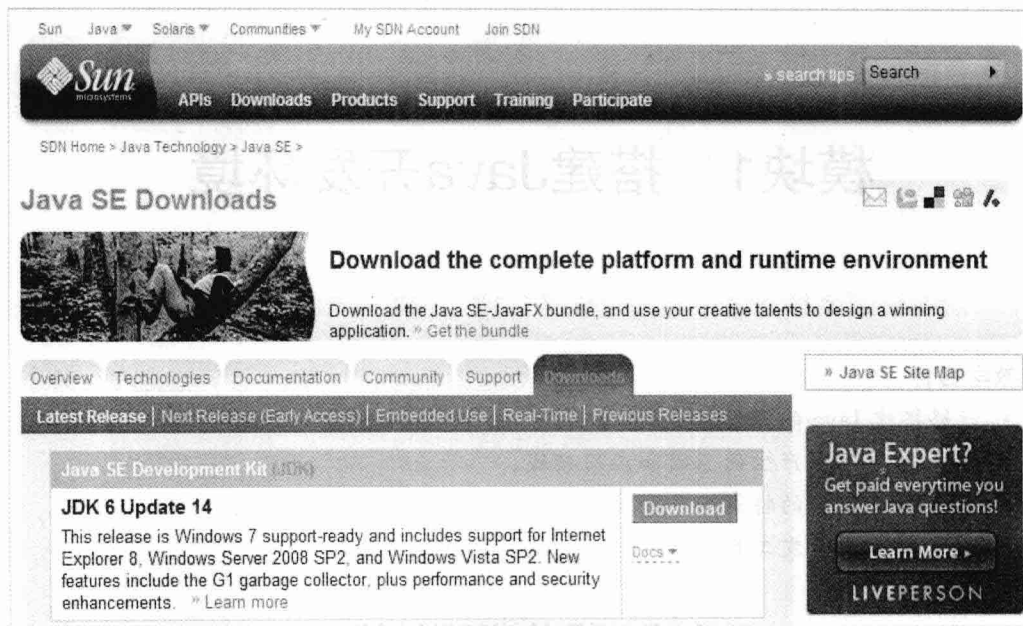


图 1-1 JDK 下载主页面

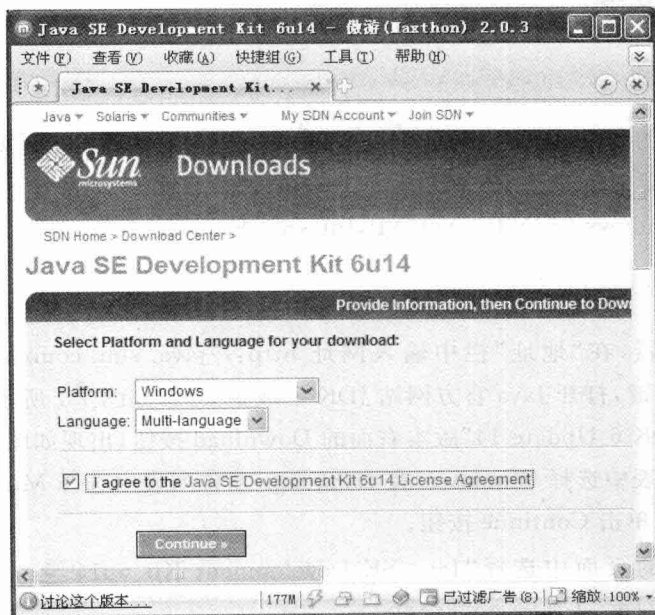


图 1-2 在 JDK 下载页面中选择平台及语言

安装。

安装成功后, JDK 的目录结构如图 1-3 所示。

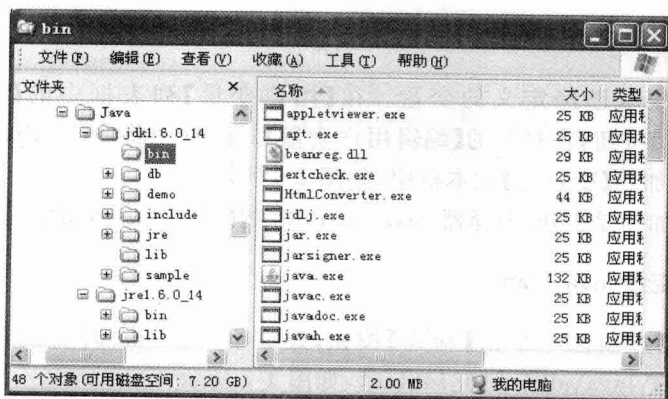


图 1-3 JDK 的目录结构

1.1.2 配置环境变量

由于 Java 是与平台无关的,安装 JDK 时 Java 不会自动设置路径,也不会修改注册表,需要用户自己设置环境变量,但不需要修改注册表。设置的环境变量包括 path、java_home 和 classpath。

在桌面上右击【我的电脑】,在弹出的快捷菜单中选择【属性】命令,在弹出的【系统属性】对话框中选择【高级】选项卡,如图 1-4 所示。

单击【环境变量】按钮,打开【环境变量】对话框,如图 1-5 所示。

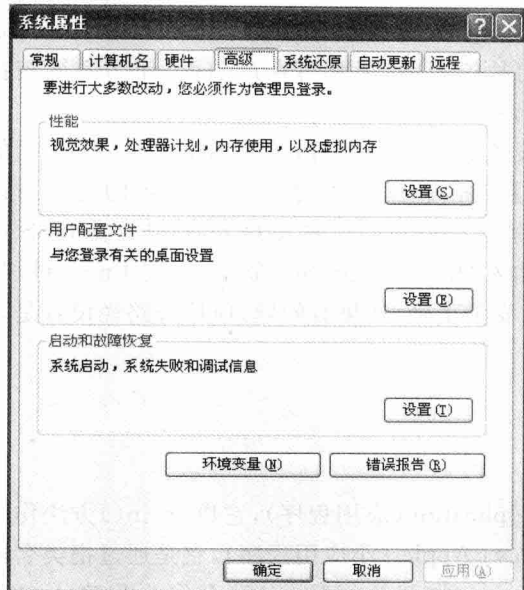


图 1-4 【系统属性】对话框

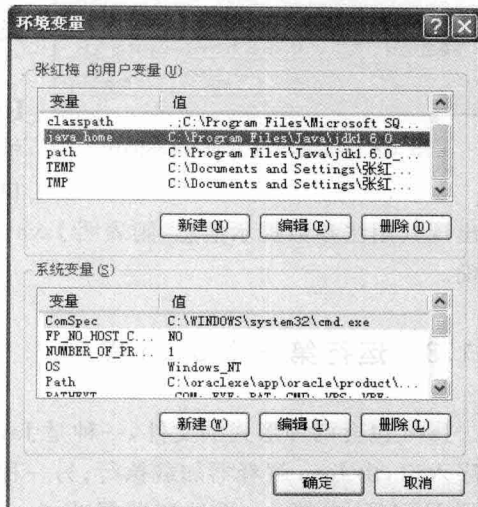


图 1-5 【环境变量】对话框



1. 编辑用户变量 Path

Path 是 Windows 已经定义的变量。在【用户变量】列表框中的【变量】一栏,找到“Path”,单击【编辑】按钮,在打开的【编辑用户变量】对话框中将“;C:\Program Files\Java\jdk1.6.0_14\bin”加到【变量值】文本框中,如图 1-6 所示。

向 Path 中添加的是 JDK 编译器 Javac.exe 和解释器 Java.exe 的路径。

2. 新建环境变量 java_home

在【用户变量】选项区中单击【新建】按钮,建立 java_home 环境变量,【变量值】设为“C:\Program Files\Java\jdk1.6.0_14\bin”,如图 1-7 所示。

单击【确定】按钮,配置成功。

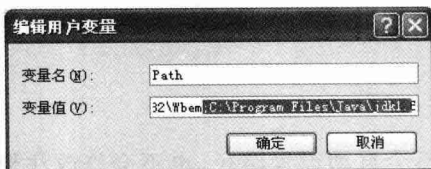


图 1-6 编辑用户变量 Path

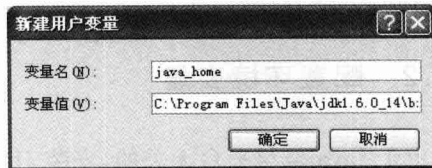


图 1-7 新建环境变量 java_home

3. 新建 classpath

classpath 是 JDK 包(类库)的路径,在【用户变量】选项区中单击【新建】按钮,建立环境变量 classpath,如图 1-8 所示。

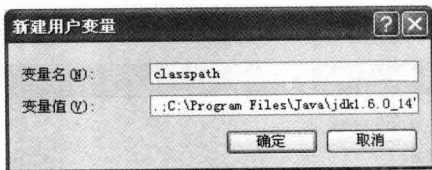


图 1-8 新建环境变量 classpath

classpath 变量值中的“.”表示在任意当前目录下均可执行 JDK,通常写在最前面,作为系统查找类的第一个路径。

设置好环境变量后,单击 Windows 任务栏的【开始】按钮,选择【运行】命令,在弹出的运行对话框中输入“cmd”命令,进入 DOS 窗口。在 DOS 窗口下输入“Java”或“Javac”命令后,按 Enter 键,如

果出现其用法参数提示信息,则表明 Java JDK 安装正确;如果有问题,应检查路径设置是否正确。

1.1.3 运行第一个 Java 程序

Java 包含两种形式的文件,一种是 Java Application(应用程序),它以 main()方法作为程序入口,由 Java 解释器加载执行;另一种是 Java Applet(小应用程序),它是通过把类名嵌入到 HTML 文档中,通过浏览器或 Appletviewer 加载执行的,它没有作为程序入口的 main()方法。下面看第一个 Java Application 程序。

(1) 在 DOS 窗口下,设定当前目录为 D:\>,输入“notepad”命令后按 Enter 键,打开记



事本。

```
D:\>notepad
```

(2) 在记事本中输入下面的程序,程序的作用是在屏幕上显示“Hello world!”字符串。

```
//第一个 Java 程序
public class HelloWorld{
    public static void main(String [] args){
        System.out.println("Hello world!");    //在屏幕上显示“Hello world!”字符串
    }
}
```

在记事本下,单击【文件】菜单的【保存】命令,打开【另存为】对话框,在【保存在】下拉列表中选择要保存的文件夹,在【文件名】文本框中输入“HelloWorld.java”,在【保存类型】下拉列表中选择“所有文件”,单击【保存】按钮,保存文件。

(3) 编译“HelloWorld.java”。

```
D:\> javac HelloWorld.java
```

执行编译命令“javac.exe”,对文件进行编译,文件名与源程序中定义的类型完全相同,首字母仍然大写。编译完成后,若未出现任何提示消息,说明编译成功。

(4) 运行“HelloWorld”程序。

```
D:\> java HelloWorld
```

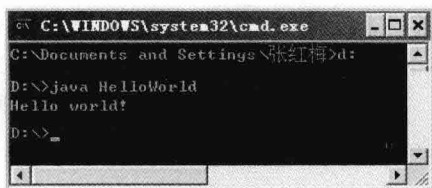


图 1-9 程序运行结果

运行“java.exe”命令,运行编译产生的字节码文件“HelloWorld.class”,结果如图 1-9 所示。

1.2 相关理论知识

1.2.1 Java 概述

Java 包括 Java 编程语言、开发工具和环境、Java 类库等。

Java 语言是由美国 SUN 公司开发的一种完全面向对象的程序设计语言。Java 语言由 C++ 语言发展而来,但比 C++ 语言简单。Java 语言具有简单、与平台无关、高效、健壮、安全、多线程等特点,它是当前网络编程中的首选语言。

从 1995 年 5 月诞生至今,Java 语言随着 Internet 的迅猛发展而成长壮大,现已成为 Internet 上的主流编程语言。J2ME/J2SE/J2EE 三大平台强大的应用系统设计能力,使 Java 无处不在。

1. 前身

Java 的故事最早与一棵大橡树有关。1991 年,SUN 公司成立 GREEN 项目组,致力于



开发一种用于消费电子设备控制的嵌入式系统,如交互式有线电视(CABLE TV)和家用电器的设备等。由于这些电子设备品种繁多并且标准各异,GREEN 项目组希望该控制系统具有简单、可靠、安全、容易联网和跨平台等特性,并且具有支持系统开发的编程工具。

该系统最初采用 C++ 语言开发,但由于 C++ 语言太繁琐而且安全性差,不能满足要求,于是 GREEN 项目组研究开发了一种新语言,取名为 OAK(橡树),这是因为项目组负责人办公室窗外有一棵大橡树。

后来由于商标冲突,OAK 被改为 Java。Java 取名于印度尼西亚位于印度洋中的一个岛屿,中文称为爪哇,它盛产咖啡。语言的标志就是一杯热咖啡。虽然可编程控制设备的消费市场并没有像预期那样大规模地发展,但是 Java 并没有因此而随之埋没。

2. JDK 版本

1995 年 5 月,SUN 公司发布了第 1 版开发工具包(Java Development Kit,JDK),JDK 提供运行时环境。JDK 的版本从最初的 1.0 到最新的 1.6,经历了几次更新。1998 年年底发布的 1.2 版本是一个重要版本,SUN 称之为 Java 2 SDK(Software Development Kit),此后的 Java 就称为 Java 2 平台。Java 2 SDK 增加了许多扩展的新特性,同时废弃了原版本的许多方法。新特性的重点是用新方法构建程序,如使用类库或者使用应用程序接口(Application Programming Interface,API)。

SUN 公司采用开放性策略,在其网站 <http://java.sun.com> 上可以免费获取 JDK,这也是 Java 语言能够迅速发展的重要因素。不同的操作系统平台需要使用不同的 JDK。

1.2.2 Java 语言的特点

Java 诞生于网络时代,虽然它的历史短暂,但它的发展壮大并不是偶然,而是有着内在的基础和外在的机遇。

Java 语言建立在成熟的算法语言和坚实的面向对象理论的基础之上,具有强大的应用系统设计能力。而它所具备的平台无关性和安全性等特点是其具有能够充分适应网络需要的无可比拟的优势。

Java 成为目前网络编程的首选语言,充分说明了 Java 语言的设计思想和其所具有的特点适应了网络发展的特殊需要。

Java 语言具有以下几个特点。

1. 结构简单,简单易学

衍生自 C++ 的 Java 语言,出于安全稳定性的考虑,去除了 C++ 中不容易理解和掌握的部分,如最典型的指针操作等。Java 语言的程序构成与 C 语言和 C++ 语言类似,基本语法部分与 C 语言几乎一模一样,这样无论是掌握了 Java 语言再学 C 语言,还是掌握了 C 语言再来学习 Java 语言,都会感到易于入门。

2. 面向对象

在现实世界中,任何实体都可以看做是一个对象,对象具有状态和行为两大特征。在



Java 语言中,没有采用传统的、以过程为中心的编程方法,而是采用以对象为中心,通过对象之间的调用来解决问题的编程方法。

3. 平台无关性

使用 Java 语言编写的应用程序不需要进行任何修改,就可以在不同的软、硬件平台上运行,因此大大降低了开发、维护 and 管理的开销。这主要是通过 Java 虚拟机(JVM)来实现的。

4. 可靠性

因为 Java 语言最初的设计目的是应用于电子类家庭消费产品,所以要求较高的可靠性。例如,Java 语言提供了异常处理机制,有效地避免了因程序编写错误而导致的死机现象。

5. 安全性

现今的 Java 语言主要应用于网络应用程序的开发,因此对安全性有很高的要求。如果没有安全保证,用户运行从网络下载的 Java 应用程序是十分危险的。Java 语言通过使用编译器和解释器,在很大程度上避免了病毒程序的产生和网络程序对本地系统的破坏,另外 Java 特有的机制是其安全性的保障,同时它去除了 C++ 中易造成错误的指针,增加了自动内存管理等措施,保证了 Java 程序运行的可靠性。

6. 多线程

多线程是当今软件技术的一大重要成果,已成功应用在操作系统、应用开发等多个领域。多线程技术允许同时存在几个执行体,按几条不同的执行路线共同工作,满足了一些复杂软件的需求。Java 不但内置多线程功能,而且提供语言级的多线程支持,即定义了一些用于建立、管理多线程的类和方法,使得开发具有多线程功能的程序变得简单、容易、有效率。

7. 高效的网络编程

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议,用户可以通过 URL 地址在网络上很方便地访问其他对象。Java 的小应用程序(Applet)是动态的、安全的、跨平台的网络应用程序。Java Applet 嵌入 HTML 语言,通过主页发布到 Internet。网络用户访问服务器的 Applet 时,这些 Applet 从网络上进行传输,然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制,用户一旦载入 Applet,就可以像图像、声音、动画等一样从网络上下载。但它并不同于这些多媒体的文件格式,它可以接收用户的输入,动态地进行改变,而不仅仅是动画的显示和声音的播放。

1.2.3 Java 程序类型

Java 程序有两种形式: Application(应用程序)和 Applet(小应用程序)。

Application 是能够独立运行的程序,既可以 DOS 的命令方式运行,也可以图形用户界



面方式运行,它需要独立的编译器来编译后才能运行。

而 Java 小程序 Applet 是使用 Java 语言编写的一段代码,它可以在浏览器环境中运行,它与 Application 的区别主要在于其执行方式的不同,Application 是从其中的 main() 方法开始运行的,而小应用程序则无法使用命令行直接运行,它是嵌在 HTML 编写的 Web 页面中的非独立程序,由 Web 浏览器内包含的 Java 编译器来编译并执行。

虽然 Java 应用程序和 Java 小程序都有各自的使用方法及使用场合,但是它们编译和运行的方法都是一样的,分为以下三步:

- (1) 用“记事本”或其他文本工具编写源程序,保存时文件名与源程序中定义的类(class)的名字必须一致,扩展名为 .java。
- (2) 在 DOS 命令环境下编译源程序,计算机将自动建立一个和源程序名字相同的扩展名为 .class 的文件,此文件是源程序编译产生的字节码文件。
- (3) 运行编译好的字节码文件,输出程序的运行结果。

1.2.4 Java 语言实现机制

1. 开发过程

如图 1-10 所示,在 Java 编程中,首先是在以 .java 为扩展名的文件中写入 Java 的源代码。这些源代码通过 Java 编译器编译成 .class 文件。.class 文件中并不包含特定于用户处理的本地代码,它包含的是字节码,是 JVM(Java Virtual Machine)机器语言,然后 Java 导入工具就能够在 JVM 上运行用户的应用程序了。



图 1-10 Java 程序开发过程

因为在不同的操作系统上都有 JVM,所以同一个 .class 文件既能在 Windows 上运行,也能在 Solaris、Linux、Mac OS 操作系统上运行,如图 1-11 所示。并且有一些虚拟机(如 Java HotSpot 虚拟机)在运行时能够完成额外的步骤,从而提升应用程序的效率,包括找出运行瓶颈和把经常使用的代码重新编译为本地代码。

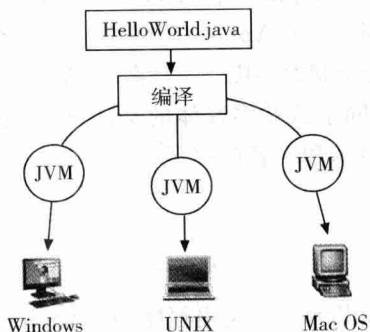


图 1-11 Java 程序的跨平台运行