



普通高等教育“十一五”国家级规划教材

Java 程序设计

第2版

◎ 主编 马世霞



配电子课件



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”国家级规划教材

Java 程序设计

第2版

主 编 马世霞

副主编 郭祖华

参 编 戴 冬 李敬伟

李扬波 刘 丹



机械工业出版社

本书以 Java 2 技术为背景,由浅入深、通俗易懂地介绍了 Java 编程语言。

全书共分为 12 章,分别介绍了 Java 概论,Java 基本语法,面向对象编程,异常处理,Applet 程序设计,图形用户界面,输入与输出,多线程,集合框架,访问数据库,网络编程,游戏。本书附录中还提供了相关实验。

书中列举了许多实例,每章都有知识测试,帮助读者提高解决实际问题的能力。

本书以高职高专学生为主要对象,既可作为高职高专计算机类的教材及职业培训教材,也可作为其他专业的选学教材。

本书配有电子课件、习题答案、源代码、测试题库等相关资料,选用本书的老师可以登录机械工业出版社教育服务网 www.cmpedu.com 下载。咨询邮箱: cmpgaozhi@sina.com。咨询电话: 010-88379375。

图书在版编目(CIP)数据

Java 程序设计/马世霞主编. —2 版. —北京:机械工业出版社,2014.1

普通高等教育“十一五”国家级规划教材

ISBN 978-7-111-44702-3

I. ①J… II. ①马… 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 265324 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:王玉鑫 责任编辑:李大国

责任校对:薛娜 封面设计:张静

责任印制:张楠

北京诚信伟业印刷有限公司印刷

2014 年 1 月第 2 版第 1 次印刷

184mm×260mm·15.25 印张·376 千字

0 001—3000 册

标准书号:ISBN 978-7-111-44702-3

定价:29.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

前 言

计算机网络技术正在以前所未有的速度迅猛发展,在网络程序设计应用领域,从电子商务、远程教学到网络游戏等都在纷纷使用 Java 技术。Java 手机编程和基于 Java 技术的各种芯片的应用在日常生活中也随处可见。Java 语言已成为目前最具吸引力且功能强大的程序设计语言之一。Java 语言是完全面向对象的,并且具有容易学习、功能强大、程序的可读性好等优点。Java 语言的编程技术正逐步成为计算机网络程序设计的主流。

Java 语言不仅可以用来开发大型的应用程序,而且在 Internet 上有着重要而广泛的应用。Java 具备了“一次撰写,到处运行”的特点,尤其是 Java Swing 的推出,使 Java 的功能更加强大。

由于教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具,也是深化教育教学改革,全面推进素质教育,培养创新人才的重要保证,因此,本教材在内容的编排上做了精心的设置与选取,注重基本知识的理解与基本技能的培养。全书思路清晰、结构严谨,叙述由浅入深、循序渐进,用语规范,全面准确地讲述了基本语法和面向对象技术等理论内容,完整地介绍了 Java 2 面向对象程序设计的要点和难点。尤其在结构上特别注重前后内容的连贯性,做到了抓住关键、突出重点、分解难点,体现了“理论性、实用性、技术性”三者相结合的编写特色。同时,将实用性强的应用程序穿插在理论叙述中,以实例体现和巩固理论基础知识,并结合新技术的发展趋势,介绍了网络通信机制等内容。这些实例汇集了作者多年从事计算机教学和软件开发过程中的案例精品。

本书是一本实用教程,内容详尽,实例丰富,注重培养解决问题的能力。每章都附有知识测试题,便于教师教学和检验学生的学习效果。全书共分 12 章:Java 概论,Java 基本语法,面向对象编程,异常处理,Applet 程序设计,图形用户界面,输入与输出,多线程,集合框架,访问数据库,网络编程,游戏。本书在附录中提供了实验。

本书由马世霞任主编,郭祖华任副主编,参加本书编写的还有戴冬、李敬伟、李扬波、刘丹。

本书既可作为高等院校 Java 程序设计课程的教材和教学参考书,也可作为 Java 编程人员的参考书。

特别说明:本书中所有例程源代码之前的序号均是方便了程序分析而另外加的,读者书写源程序时请务必将序号去掉。

编 者

目 录

前言

第一部分 基础篇

第 1 章 Java 概论	2
1.1 Java 产生的背景	3
1.2 Java 平台简介	3
1.3 Java 平台和虚拟机	3
1.4 运行环境安装与测试	4
1.5 初识两类 Java 程序	7
1.6 Java 编程规范	11
1.7 本章小结	11
1.8 知识测试	12
第 2 章 Java 基本语法	13
2.1 简单数据类型	14
2.2 控制语句	25
2.3 数组	34
2.4 本章小结	37
2.5 知识测试	38
第 3 章 面向对象编程	41
3.1 面向对象的思想	42
3.2 类	43
3.3 对象	48
3.4 继承与多态	51
3.5 抽象类和接口	57
3.6 包	62
3.7 系统常用类	63
3.8 本章小结	73
3.9 知识测试	73
第 4 章 异常处理	75
4.1 异常处理的概念	76
4.2 异常类	76
4.3 异常处理的方法	78
4.4 创建自己的异常类	84
4.5 本章小结	85
4.6 知识测试	85
第 5 章 Applet 程序设计	87
5.1 Applet 的生命周期和 Applet 的方法	88
5.2 Applet 标记	91

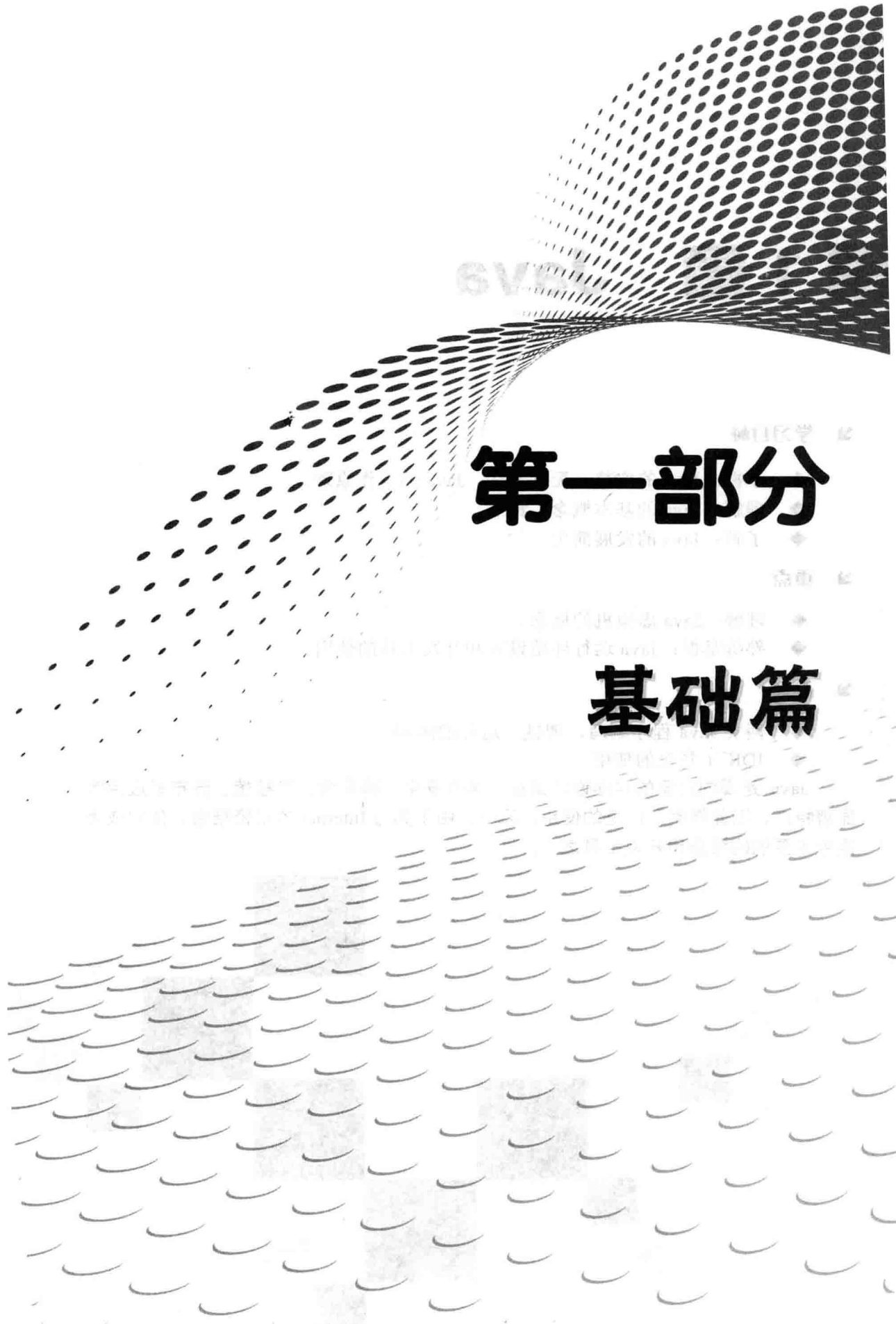
5.3 Applet 通信	92
5.4 Applet 程序示例	97
5.5 本章小结	99
5.6 知识测试	99
第 6 章 图形用户界面	101
6.1 Java GUI 概述	102
6.2 常用容器与组件	103
6.3 事件处理概述	108
6.4 布局管理器	118
6.5 复杂组件与事件处理	123
6.6 本章小结	131
6.7 知识测试	131
第 7 章 输入与输出	132
7.1 I/O 流概述	133
7.2 字节流	134
7.3 字符流	143
7.4 本章小结	148
7.5 知识测试	149

第二部分 提高篇

第 8 章 多线程	152
8.1 线程概述	153
8.2 线程的实现	155
8.3 线程的同步	158
8.4 本章小结	163
8.5 知识测试	163
第 9 章 集合框架	165
9.1 集合框架概述	166
9.2 Collection	166
9.3 List	170
9.4 Map	172
9.5 Set 常用方法	175
9.6 本章小结	178
9.7 知识测试	178
第 10 章 访问数据库	181
10.1 概述	182
10.2 JDBC 应用程序接口	183

10.3	配置 ODBC 数据源	185
10.4	数据库编程实例	188
10.5	知识测试	190
第 11 章	网络编程	191
11.1	网络编程的基本概念	192
11.2	传输层协议 TCP 和 UDP	193
11.3	Java 与统一资源定位符 (URL)	194
11.4	Java 与 Socket 编程	199
11.5	本章小结	212
11.6	知识测试	212
第 12 章	游戏	213
12.1	打字游戏介绍	214
12.2	游戏实现	214

附录 实验	222
实验 1 熟悉 Java 编程环境和 Java 程序结构	223
实验 2 Applet 程序设计	224
实验 3 Java 基本语法	225
实验 4 面向对象基础	226
实验 5 异常处理	227
实验 6 图形用户界面	229
实验 7 输入/输出	230
实验 8 多线程	231
实验 9 图形、动画与多媒体	233
实验 10 数据库	234
实验 11 网络编程	235
参考文献	238



第一部分

基础篇

第 1 章 Java 概论

学习目标

- ◆ 掌握：Java 的安装、配置方法，Java 的工作原理。
- ◆ 理解：Java 的基本概念、特点。
- ◆ 了解：Java 的发展简史。

重点

- ◆ 理解：Java 虚拟机的概念。
- ◆ 熟练掌握：Java 运行环境设置和开发工具的使用。

难点

- ◆ 两类 Java 程序编写、调试、运行的区别。
- ◆ JDK 工具包的使用。

Java 是面向对象的程序设计语言，具有安全、跨平台、可移植、分布式应用等显著特点，因此得到了广泛的使用。同时，由于其与 Internet 的紧密结合，使它成为当今主要的网络应用开发工具之一。

■ 1.1 Java 产生的背景

1990 年, 美国 Sun 公司启动了一个叫 Green 的项目, 其原先的目的是为家用消费电子产品开发一个分布式代码系统, 这样人们可以把信息发给电冰箱、烤面包机、微波炉、电视机等家用电器, 对它们进行控制, 和它们进行信息交流。

当时最流行的编程语言是 C 和 C++, Sun 公司的研究人员起初曾考虑是否可以采用 C++ 来编写程序, 但是研究表明, 由于家电价格较低, 而 C++ 过于复杂和庞大, 安全性又差, 运行语言所需的内存和处理芯片的花费, 已超过家电本身的成本, 为了解决此问题, 最后基于 C++ 开发了一种新的语言 Oak (橡树)。Oak 是一种用于网络的精巧而安全的语言, 但由于后来发现 Oak 已是 Sun 公司另一个语言的注册商标, 所以才改名为 Java。Java 的取名也有一段趣闻。有一天, 几位 Java 成员组的成员正在讨论给这个新的语言取什么名字, 当时他们正在咖啡馆喝着 Java 咖啡 (太平洋岛屿爪哇盛产的一种味道非常美妙的咖啡), 有一个人灵机一动说就叫 Java 怎样, 希望 Java 语言就像端到编程人员面前的热腾腾的咖啡一样, 让人感觉很舒服。建议得到了其他人的赞赏, 于是, Java 这个名字就这样传开了。1995 年 5 月, Sun 公司对外正式发布了 Java 语言。

■ 1.2 Java 平台简介

1999 年 6 月, Sun 公司推出以 Java2 平台为核心的 J2SE、J2EE 和 J2ME 三大平台。

(1) J2SE, 即 Java2 Standard Edition 是 Java 的标准版, 用于小型程序。

(2) J2EE, 即 Java2 Enterprise Edition 是 Java 的企业版, 用于大型程序。

(3) J2ME, 即 Java2 Micro Edition 是 Java 的微型版, 用于手机等程序。

J2SE、J2EE、J2ME 语言是相同的, 只是捆绑的类库 API 不同。J2SE 是基础; 压缩一点, 再增加一些 CLDC 等方面的特性就是 J2ME; 扩充一点, 再增加一些 EJB 等企业应用方面的特性就是 J2EE。本书主要讲述 J2SE。

■ 1.3 Java 平台和虚拟机

平台是支持程序运行的软硬件环境。Java 平台是指在 Windows、Linux 等系统平台支持下的一种 Java 程序开发平台, 主要由 Java 虚拟机 (JVM, Java Virtual Machine) 和 Java 应用程序接口 (Java API) 两部分组成。Java 虚拟机易于移植到不同硬件的平台上, 是 Java 平台的基础; Java 应用程序接口由大量已做好的 Java 组件 (组件是一种类) 构成, 该接口提供了丰富的 Java 资源, 使 Java 程序开发的效率比其他语言大大提高。

Java 破解各机器使用不同的机器语言的策略就是它定义出自己的一套虚拟机, 即 Java 虚拟机, 它是分布在不同平台上的解释器。Java 虚拟机的工作原理如图 1-1 所示。

为了实现将程序动态地转换成与平台无关的形式, Java 编程人员在编写完软件后, 首先通过 Java 编译器, 将 Java 源程序编译为 JVM 的字节码 (Byte Code), 并生成以 “.class” 为扩展名的文件, 同时在 Java 虚拟机上, 有一个 Java 解释器用来解释 Java 字节码。Byte Code 就是中介语言 (一种共通语言), 任何一台机器只要配备了 Java 解释器, 都可读懂并运行这个程序。

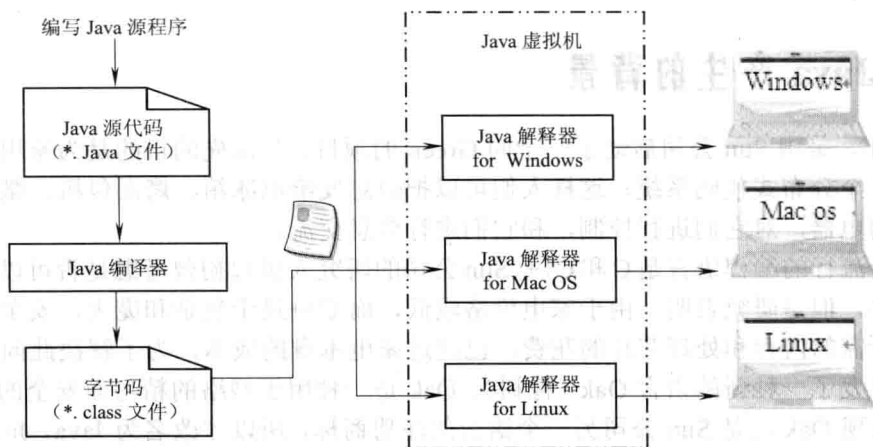


图 1-1 Java 虚拟机的工作原理图

编译器：对源代码进行半编译，生成与平台无关的字节码文件。

解释器：分布在网络中不同的操作系统平台上，用于对字节码文件解释执行。

解释就是取出一条指令并执行，有点像日常的口译；而编译有点像笔译，全部翻译以后才去执行。经过 Java 解释器的解释，平台就可以执行各种各样的 Java 程序。正因为如此，Java 程序才具有了“一次编写，到处运行”的特点，即 Java 的平台无关性。

■ 1.4 运行环境安装与测试

Java 开发环境的基本要求非常低，只需一个 Java 开发工具包 (JDK, Java Development Kit)，再加上一个纯文本编辑器即可。为了提高开发效率，可以使用功能强大的文本编辑工具，如记事本、UltraEdit 等。对于熟练的开发人员，为了进一步提高开发效率，还可以使用具有可视化功能的 Java 专用开发工具，如 Jbuilder、J++、NetBeans。本书程序以记事本为编辑工具。

1.4.1 Java 开发包的安装

Java 开发工具包 (JDK) 是一种开发环境，用于使用 Java 编程语言生成应用程序、Applet 和组件。不同的操作系统（如 Windows、UNIX / Linux、Mac OS）有相应的 Java 开发包安装程序。读者可以登录 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 获取 Java 开发包安装程序。本书中使用 Windows 操作系统环境下的 Java 开发包，所给的例子程序均在 JDK 1.6.0 环境下调试通过。

1. JDK 的内容

在 JDK 根目录下，有 bin, jre, lib, demo, include 子目录和一些文件，其功能如下：

(1) 开发工具

开发工具位于 bin 子目录中，是指工具和实用程序，可进行 Java 程序开发、编译、运行、调试等。有关详细信息，请参见工具文档。

(2) 运行时环境

运行时环境位于 jre 子目录中，是 J2SE 运行时环境的实现。该运行时环境包含 Java

虚拟机、类库以及其他文件，可支持执行以 Java 编程语言编写的程序。

(3) 附加库

附加库位于 lib 子目录中，是开发工具需要的附加类库和支持文件。

(4) 演示 Applet 和应用程序

demo 子目录中包含演示 Java Applet 和 Java 应用程序的示例，有含源代码的 Java 平台编程示例，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) c 头文件

c 头文件位于 include 子目录中。包含支持使用 Java 本机界面、JVM™ 工具界面以及 Java 平台的其他功能进行本机代码编程的头文件。

(6) 源代码

源代码位于 src.zip 中。包含组成 Java 2 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）。此源代码仅用于提供信息，以便帮助开发者学习和使用 Java 编程语言。

2. JDK 的基本命令

JDK 包含用于开发和测试以 Java 编程语言编写并在 Java 平台上运行的程序的工具。这些工具被设计为从命令行使用。除了 appletviewer 以外，这些工具不提供图形用户界面。JDK 的基本命令包括 javac, java, jdb, javap, javadoc, appletviewer。

(1) javac: Java 编译器，用来将 Java 程序编译成字节码。

命令格式：javac [选项] 源程序名

(2) java: Java 解释器，执行已经转换成字节码的 Java 应用程序。

命令格式：java [选项] 类名 [参数]

(3) jdb: Java 调试器，用来调试 Java 程序。

启动 jdb 的方法有两种：第一种方法格式与 Java 解释器类似；第二种是把 jdb 附加到一个已运行的 Java 解释器上，该解释器必须是带 -debug 项启动的。

(4) javap: 反编译，将类文件还原回方法和变量。

命令格式：javap [选项] 类名

(5) javadoc: 文档生成器，创建 HTML 文件。

命令格式：javadoc [选项] 源文件名

(6) appletviewer: 小应用程序 Applet 浏览工具，用于测试并运行 Applet。

命令格式：appletviewer [选项] URL

其中 URL 是包含被显示 Applet 的 HTML 文件的统一资源定位符，当 HTML 文件位于本地机上时，只需写出文件名。

3. Java 的安装与配置

JDK 的默认路径为 “C:\program Files\Java\jdk1.6.0”，用户可选择安装路径。具体安装步骤如下：

在得到 Java 开发包后，首先需要进行安装。双击 Java 开发包 (JDK 1.6.0) 安装程序，出现安装界面，随后，安装程序会弹出许可证协议界面如图 1-2 所示。单击“接受”按钮，继续安装。在弹出的图 1-3 所示的安装界面中单击“更改”按钮，设置安装路径为 D:\Java\jdk1.6.0，单击“下一步”按钮，弹出如图 1-4、图 1-5 所示画面，安装成功。

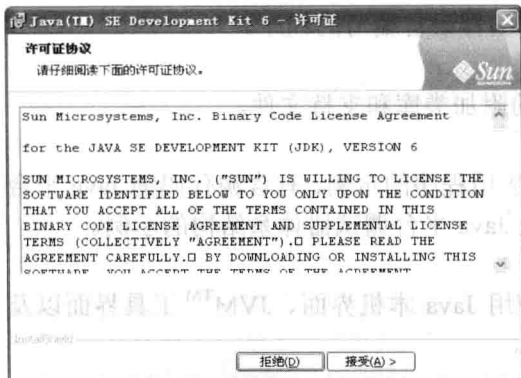


图 1-2 许可证协议阅读界面

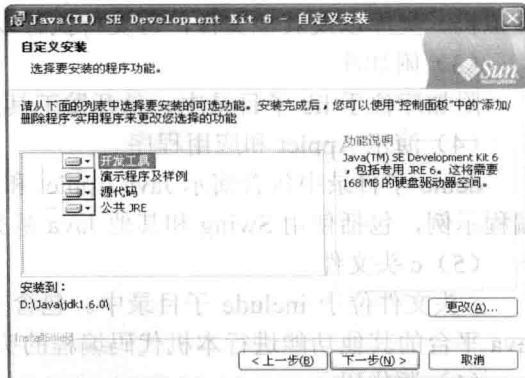


图 1-3 安装内容、安装路径

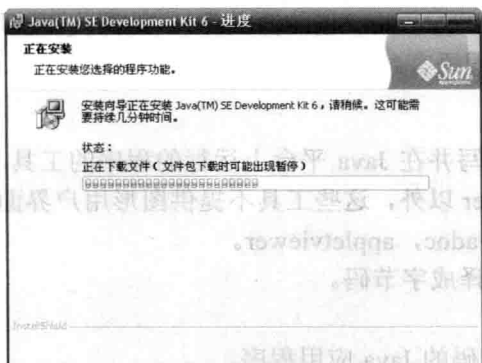


图 1-4 进度显示

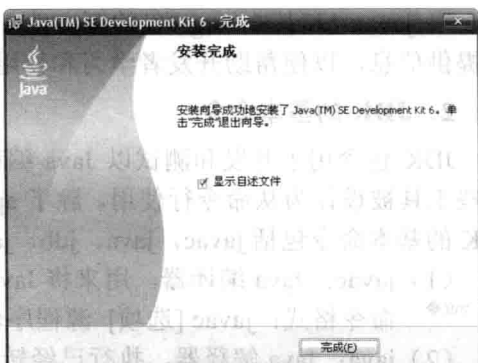


图 1-5 安装完成界面

在 bin 目录下有两个重要的文件：javac.exe、java.exe，分别是编译程序和 JVM。在使用这两个文件前，最好先设置变量 path，这样无论在哪个目录下都能编译和执行 Java 文件。

1.4.2 环境变量设定

设定环境变量是为了能够正常使用所安装的开发包，主要包括两个环境变量：Path 和 Classpath。Path 称为路径环境变量，用来指定 Java 开发包中的一些可执行程序（Java.exe、Javac.exe 等）所在的位置；Classpath 称为类路径环境变量。不同的操作系统上，设定环境变量的方法是不同的。

在 Windows XP 操作系统下，设置 Path 变量：右键单击“我的电脑”，在弹出的快捷菜单中选择“属性”，弹出“系统属性”对话框，在对话框中选择“高级”选项卡，在该页面单击“环境变量”按钮，打开“环境变量”对话框，如图 1-6 所示。

找到变量 Path，双击该行就可以编辑该环境变量的值。在该变量已有的值后，再添加

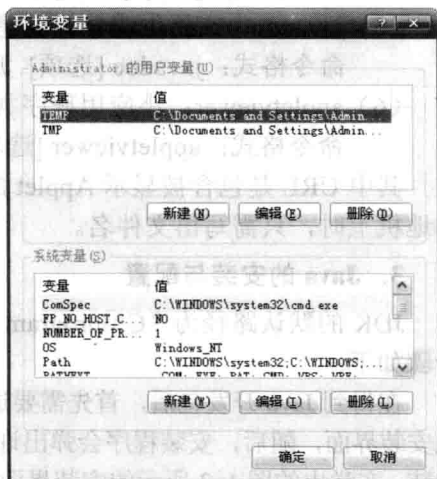


图 1-6 环境变量设置

“D:\Java\jdk1.6.0\bin”（注意：不包括引号，分号“;”不能缺少），单击“确定”按钮进行保存工作，如图 1-7 所示。

设置 Classpath 类路径环境变量：在“系统变量”列表框里，单击“新建”按钮，在“新建系统变量”对话框里，添加 Classpath，并为设变量值为“.;D:\Java\jdk1.6.0\jre\lib”（注意：不包括引号，“.”不能缺少），如图 1-8 所示。至此，完成环境变量的设定工作。

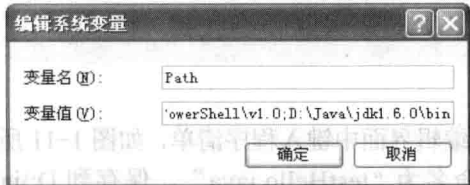


图 1-7 Path 环境变量设置

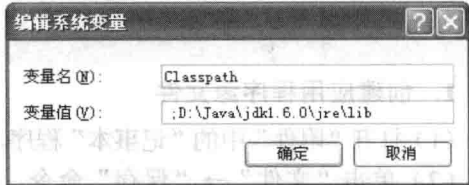


图 1-8 Classpath 环境变量设置

1.4.3 环境测试

单击“开始”→“程序”→“附件”→“命令提示符”命令，然后在 D:\Java>提示符下，输入以下两条命令，命令的实例演示如下：

(1) 版本测试：java -version

结果显示 Java 版本如图 1-9 所示。否则需要重新安装 Java。

(2) 环境测试：javac

若显示如图 1-10 所示，则说明 Path 设置有问题，需要修改系统变量 Path 的值。否则说明 Path 设置成功，现在可以编写 Java 程序了。

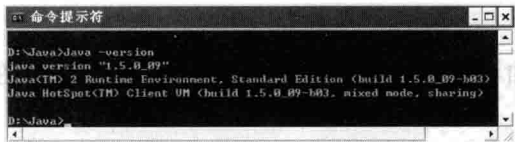


图 1-9 version 命令显示窗口



图 1-10 环境变量设置

1.5 初识两类 Java 程序

按照运行环境的不同，可将普遍使用的 Java 程序分为两种：Java 应用程序（Java Application）和 Java 小程序（Java Applet）。Java 应用程序在本机上由 Java 解释程序来激活 Java 虚拟机，而 Java 小程序则通过浏览器来激活 Java 虚拟机。此外，它们的程序结构也不相同。

1.5.1 Java 应用程序（Java Application）

Java 应用程序都是以类的形式出现的，程序中可以包含一个类，也可以包含多个类。

Java 源程序可以在任何一种文本编辑器上编辑，如记事本编辑器、UltraEdit 编辑器等，它不需要特定的编程环境，本书程序采用记事本编辑器。在记事本编辑器中书写 Java 源程序，并将其保存为扩展名为“.java”的文件。

特别说明：本书中所有例程源代码之前的序号均是为了方便程序分析而另外加的，读者书写源程序时请务必将序号去掉。

【例 1-1】源程序名为“testHello.java”的程序，在屏幕上输出“Java 欢迎你！”。

```

1  public class testHello
2  {
3      public static void main (String arg[])
4      {
5          System.out.println("Java 欢迎你！");
6      }
7  }

```

1. 创建应用程序源文件

(1) 打开“附件”中的“记事本”程序，在文本编辑界面中键入程序清单，如图 1-11 所示。

(2) 单击“文件”→“保存”命令，将文件命名为“testHello.java”，保存到 D:\java> 目录下。注意：文件名必须和清单所声明的类名即“testHello”保持一致，而且 Java 是区分大小写的，且扩展名必须是“.java”。

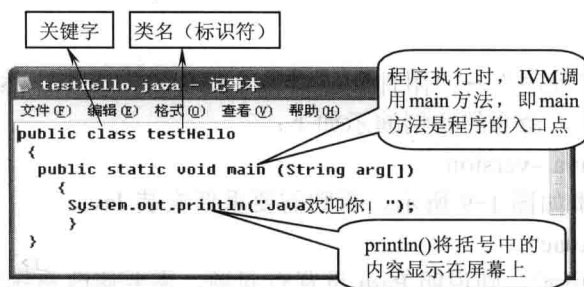


图 1-11 编写 java 代码的记事本界面

2. 编译

文件保存成功之后，从“命令提示符”窗口中进入到“D:\Java\jdk1.6.0>”目录。在此目录下，进行测试。依次输入命令：

(1) 输入编译程序 javac 命令：javac testHello.java

(2) 输入显示文件目录命令：dir

这时会发现目录下多了一个“testHello.class”文件，这是 Javac 编译器将源代码编译成字节代码生成类文件的结果。再由 Java 解释并执行“testHello.class”类文件。

3. 运行

输入运行程序命令：java testHello

运行结果如图 1-12 所示。

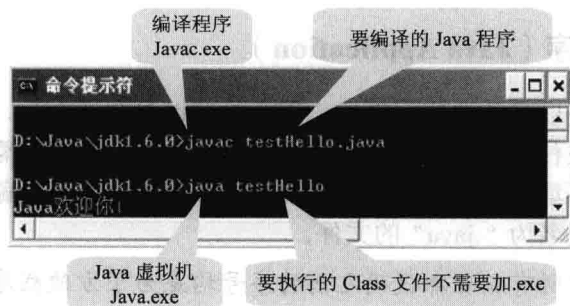


图 1-12 例 1-1 程序运行结果

4. 程序分析

输出的“Java 欢迎你！”，是 Java 编译器直接执行 Java 应用程序字节代码的结果。

第 1 句：类的声明，声明为“testHello”的类（class）。下面对每个单词进行分析：

public：说明类的属性为公共类。

class：声明一个类，代表以下的内容都是这个类的内容。

第 2，7 句：大括号是成对出现的，大括号所括起来的程序区域隶属于“testHello”类，“{”表示类“testHello”从这里开始，最后的结束是在“}”处。

第 3 句：是 Java 程序的一个特殊方法，又称 main 方法。对第 3 行每个单词进行分析：

public：访问控制符，表示 main 方法为公共的，不能省略。

static：是将 main 方法声明为静态的，在这里这个关键字也不能省略。

String arg[]：用来接收命令行传入的参数，String 是声明 arg[] 可存储字符串数组。本程序中没用到这个参数，但这个参数不能省略，否则会出错。

第 4，6 句：大括号也必须是成对出现的。

第 5 句：是将 println（）小括号中的内容“Java 欢迎你！”显示到屏幕上。注意语句后面的“；”不能省略，它代表单一程序语句完成。

System.out.println 的解释：System 是 Java 中的 System 类，out 是指 System 类中的一个变量，println 是 out 变量的一个方法。

1.5.2 Java 小程序（Java Applet）

Java 小程序即 Java Applet，它不能单独运行，必须通过 HTML 调入后，方能执行实现其功能，它既可以在 Appletviewer 下运行，也可以在支持 Java 的 Web 浏览器中运行。Applet 程序中必须有一个类是 Applet 类或 JApplet 类的子类，也是 Applet 的主类。

【例 1-2】Applet 小程序示例，源程序名为“Hello.java”。显示“你好 Java！”的 Applet 程序。程序清单如下：

1. 创建 Applet 源文件

```
1  import java.applet.Applet; //导入 applet 包的 Applet 类
2  import java.awt.Graphics;  //导入 awt 包的 Graphics 类
3  public class Hello extends Applet //Applet 的初始化事件
4  {
5      public void paint(Graphics f)
6      {
7          f.drawString("你好 Java!", 10, 50); //显示“你好 Java!”
8      }
9  }
```

(1) 打开“附件”中的“记事本”程序，在文本编辑界面中键入程序清单。

(2) 单击“文件”→“保存”命令，将文件命名为“Hello.java”，保存到 D:\java>目录下。

2. 编译 Applet 小程序

Applet 小程序 Hello.java 编写完成后，需对其进行编译，自动生成字节码文件 Hello.class，

其编译方法与 Java 应用程序相同。打开 DOS 窗口, 输入 “javac Hello.java” 命令编译生成 Hello.class 文件, 如图 1-13 所示。

3. 编写 HTML 代码

对于 Applet 小程序的字节码程序 Hello.class 必须嵌入到 HTML 代码中, 才可以完成其功能。

所以, 还必须为 Hello.class 编写一个 HTML 的代码文件, 将字节码程序引入其中。下面是嵌入 Hello.class 代码的 HTML 程序示例, 文件名为 Hello.html。

```

1  <HTML>
2  <HEAD>
3  <TITLE>Applet Program</TITLE>
4  </HEAD>
5  <BODY>
6  <APPLET CODE="Hello.class" width=400 height=150>
7  </APPLET>
8  </BODY>
9  </HTML>

```

4. 运行 HTML 代码程序

完成 HelloWorld.html 的编写后, Applet 小程序的运行有两种方式: 一种是使用 Internet Explorer 浏览器 (简称 IE) 解释运行它。另一种是使用 appletviewer 命令运行它。

(1) 在浏览器中执行 HTML 程序

双击 Hello.html 文件, 在浏览器中执行小应用程序, 运行结果如图 1-14 所示。

(2) 打开 DOS 窗口, 输入 “appletviewer Hello.html” 命令, 即使用 Applet 阅读器 (JDK 的 appletviewer) 执行小应用程序, 运行结果如图 1-15 所示。



图 1-14 Hello.html 在浏览器中的运行界面

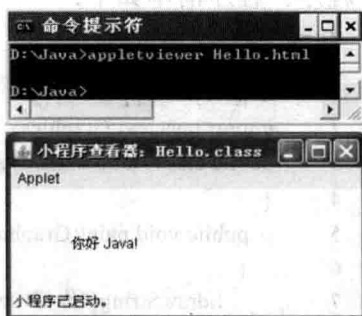


图 1-15 使用 appletviewer 运行 Hello.html 的界面

5. 程序分析

(1) Applet 源文件分析

第 1、2 句是 import 语句, 指出本 Applet 程序所需要的 Applet 类、Graphics 类, 编写 Applet 程序, 通常都要导入 Applet 类。

第 3 句, 由关键字 class 引入类的定义, Hello 为类名, 关键字 extends 说明该类继承

Applet 类, 即定义的 Hello 类是 Applet 的子类, 该类是 public 型。与 Java 应用程序一样, 把一个 public 类存入文件时, 定义的类型必须是文件名。编写好的程序存入 Hello.java 文件中。

第 4、9 句: 括号要成对出现, 所括起来的范围称为这个程序的区域。

第 5 句: 表示程序中只有一个方法: 其中参数 f 为 Graphics 类, 表示当前作画的上下文。

第 6、8 句: 括号要成对出现, 所括起来的范围称为这个方法程序的区域。

第 7 句: 在该方法中, f 调用方法 drawString(), 在坐标 (10, 50) 处输出字符串“你好 Java!”, 其中坐标以像素为单位。

(2) HTML 文件分析

HTML 标记都是成对出现的, 如<HTML>与</HTML>成对出现, 它们表示 HTML 标记的起始和结束。而<APPLET CODE="Hello.class" width=400 height=150>语句是 Applet 的特殊 HTML 标记, 用来告诉浏览器或 Applet 工具应装载的 Applet, 并设定显示窗口的宽度为 400 和长度为 150。

1.6 Java 编程规范

软件开发是一个集体协作的过程, 程序员之间的代码是要经常进行交换阅读的, 因此, Java 源程序有一些约定俗成的命名规定, 主要目的是为了提高 Java 程序的可读性。

如果在源程序中包含有公共类的定义, 则源文件名必须与公共类的名字完全一致, 字母的大小写都必须一样。这是 Java 语言的一个严格的规定。源文件的命名规则如下:

(1) 包名: 包名是全小写的名词, 中间可以由点分隔开, 如 **java.awt.event**;

(2) 类名: 首字母大写, 通常由多个单词合成一个类名, 要求每个单词的首字母也要大写, 如 **class HelloWorld**;

(3) 接口名: 命名规则与类名相同, 如 **interface Collection**;

(4) 方法名: 往往由多个单词合成, 第一个单词通常为动词, 首字母小写, 中间的每个单词的首字母都要大写, 如 **balanceAccount, isButtonPress**;

(5) 变量名: 全小写, 一般为名词, 如 **length**;

(6) 常量名: 基本数据类型的常量名为全大写, 如果是由多个单词构成, 可以用下划线隔开, 如 **int YEAR; int WEEK_OF_MONTH**;



注意

编写程序时一定要用缩进格式, 并加上注释。

1.7 本章小结

本章首先介绍了 Java 基本概念、特点、发展简史。Java 是一种程序设计平台。既是开发环境, 又是应用环境。Java 语言的基本概念如下:

Java 语言=面向对象的程序设计语言

+与机器无关的二进制格式类文件