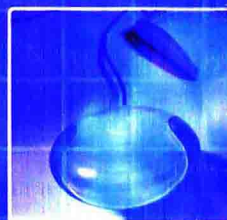




高等学校**应用型特色**规划教材

Java

程序设计



吴金丹 鞠凤娟 编著

赠送
电子教案

清华大学出版社

高等学校应用型特色规划教材

Java 程序设计

吴金舟 鞠凤娟 编著

清华大学出版社
北 京

内 容 简 介

Java 语言具有面向对象、跨平台、安全、稳定、多线程等优良特性,是目前软件设计中一种极为强大的编程语言。本书共分 13 章,循序渐进地介绍了 Java 语言的入门基础、面向对象知识和核心 API 应用,包括标识符、关键字、数据类型、常量、变量、运算符、表达式和语句、程序的控制结构、数组、字符串、类和对象、封装、继承、多态、接口与包、异常处理、Java 输入输出流、Java 图形用户界面编程、Java 图形与多媒体处理、多线程、Java 数据库编程基础、Java 网络编程基础等知识。

本书注重结合实例,每章基本上都是从实例入手,然后系统地介绍实例所涉及的知识点,整本书从前至后注重应用性,由浅入深、逻辑性强,并且每章末尾均配有小结和思考与练习。

本书适合作为高等院校计算机专业本、专科学生的教材或参考书,也可作为 Java 语言自学者的参考用书。

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/吴金舟等编著. --北京:清华大学出版社,2014
(高等学校应用型特色规划教材)
ISBN 978-7-302-34020-1

I. ①J… II. ①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 234316 号



责任编辑:张彦青 桑任松
封面设计:杨玉兰
责任校对:周剑云
责任印制:李红英

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>, 010-62791865

印 刷 者:北京鑫丰华彩印有限公司
装 订 者:三河市新茂装订有限公司
经 销:全国新华书店
开 本:185mm×260mm 印 张:19.5
版 次:2014 年 1 月第 1 版
印 数:1~3000
定 价:38.00 元

字 数:473 千字
印 次:2014 年 1 月第 1 次印刷

产品编号:053278-01

前 言

Java 语言于 1995 年底在 Internet 舞台一亮相，便名声大噪，并在之后迅速发展，逐步取代了面向过程的程序设计技术，成为软件开发的主流技术，在开发领域中占据着重要的地位，广泛适用于智能卡、手持式电子消费类产品、桌面应用、企业级应用等多个领域。其固有的可靠性和安全性不仅令 Java 程序员放心，也令使用 Java 程序的用户放心。Java 内建了对网络编程、数据库连接、多线程等高级程序设计任务的支持，具有广阔的发展前景。Java 语言程序设计课程不仅是计算机专业重要的专业课，更是计算机软件开发专业的必学课程。本教材主要面向应用型高等院校计算机类专业的学生，培养技术应用型人才。内容的构造力求体现“以应用为主体”，强调理论知识的理解和运用，实现本、专科教学以实践体系为主和以技术应用能力培养为主的目标。

本书是两位作者在总结开发经验和多年教学经验的基础上编写的。通过 31 个典型的实例，按照“入门基础”、“面向对象”和“核心 API 应用”3 个层次由浅入深，由易到难地介绍了 Java 的入门基础知识、面向对象知识和 Java 核心 API 的应用知识。

全书共分 13 章：第 1~3 章为入门基础部分，讲述 Java 语言的基本概念、Java 数据类型、表达式和程序控制结构。第 4~7 章为面向对象部分，讲述 Java 语言的面向对象知识。第 8~13 章为核心 API 应用部分，讲述 Java 语言中核心 API 的应用。此外，书中有 3 个附录，分别是：Java 关键字、运算符的优先级和结合方向、各章思考与练习参考答案。书中的所有程序都在 JDK 7 环境下调试通过。

Java 是实践性和应用性都很强的课程，本书注重理论与实践相结合，每章都给出了不同层次、不同难度的思考题，并且给出了参考答案。通过习题与实例，使学生掌握所学知识，并能灵活运用所学知识解决实际问题。

本教材建议授课学时数为 90 学时，其中上机实践学时数为 30 学时以上。教师可根据学时数、专业和学生的实际情况选讲应用案例中一些较难的例子。

本书可作为高等院校计算机专业本、专科学生的教材或参考书，也可作为 Java 语言学习者的自学用书。

本书由江西科技学院吴金舟、鞠凤娟、肖令平、刘燕编写和校正，其中肖令平负责编写第 1 章，吴金舟负责编写第 4 章、第 6 章、第 8 章、第 10 章和第 12 章，鞠凤娟负责编写第 3 章、第 5 章、第 7 章、第 9 章、第 11 章和第 13 章，刘燕负责编写第 2 章，吴金舟和鞠凤娟共同完成了附录部分的编写工作。本书可提供本教材中的实例源代码、PPT 课件等教学资源。

由于编者水平有限，书中难免存在疏漏、不足之处，欢迎各位同行和广大读者对本书提出建议和修改意见。

目 录

第 1 章 Java 语言概述	1
1.1 “初识 Java——Hello World” 实例	1
1.1.1 Java 语言的发展历程	2
1.1.2 Java 语言的特点	3
1.1.3 Java 程序的分类	4
1.1.4 Java 程序的开发过程	7
1.2 Java 的开发环境	8
1.2.1 JDK 的安装	8
1.2.2 JDK 的配置	10
1.2.3 JDK 的使用	11
1.3 集成开发工具 Eclipse 的使用	13
1.3.1 下载和安装 Eclipse	13
1.3.2 Eclipse 的使用	15
本章小结	19
思考与练习	19
第 2 章 Java 语言基础	20
2.1 “学生信息描述” 实例	20
2.2 “员工工资扣税档次” 实例	25
2.3 “简易日历表” 实例	35
本章小结	47
思考与练习	47
第 3 章 数组和字符串	50
3.1 “Fibonacci 数列和杨辉三角形求值” 实例	50
3.2 “凯撒密码” 实例	60
本章小结	71
思考与练习	71
第 4 章 类和对象	74
4.1 “用户类的设计” 实例	74
4.1.1 面向对象的基本概念	75
4.1.2 类的定义	77
4.1.3 对象的创建和使用	78
4.1.4 对象的销毁	79
4.2 “用户登录设计” 实例	80
4.2.1 方法	81
4.2.2 构造方法	85
4.3 “班费管理类的设计” 实例	86
4.3.1 类变量	87
4.3.2 类方法	87
本章小结	88
思考与练习	89
第 5 章 类的封装、继承和多态	91
5.1 “银行账户” 实例	91
5.1.1 类的封装	92
5.1.2 访问权限	94
5.2 “银行开户类型设计” 实例	99
5.3 “猜水果名称” 实例	107
5.4 “定时打印字符串” 实例	113
本章小结	118
思考与练习	118
第 6 章 接口与包	121
6.1 “图书定价标准类的设计” 实例	121
6.1.1 抽象类的概念	123
6.1.2 抽象类的实现	124
6.2 “汽车类型设计” 实例	124
6.2.1 接口的概念	126
6.2.2 接口的实现	127
6.3 “图形类的设计” 实例	128
6.3.1 包的概念	129
6.3.2 包的声明与导入	129
本章小结	131
思考与练习	131
第 7 章 异常处理	134
7.1 “简单的算术运算” 实例	134

7.2 “输出字符序列”实例.....	142	10.3.2 声音文件的处理	226
本章小结.....	147	本章小结.....	227
思考与练习.....	147	思考与练习.....	227
第 8 章 输入与输出流	150	第 11 章 多线程	230
8.1 “回声程序”实例.....	150	11.1 “自由落体和平抛运动”实例.....	230
8.1.1 流的概念与原理.....	151	11.2 “生产者-消费者”实例	239
8.1.2 字节流处理类.....	151	本章小结.....	245
8.1.3 字符流处理类.....	153	思考与练习.....	245
8.2 “响应键盘输入处理”实例.....	154	第 12 章 数据库编程基础	248
8.2.1 标准输入.....	155	12.1 “连接 Access 数据库”实例	248
8.2.2 标准输出.....	156	12.1.1 关系型数据库简介	250
8.2.3 标准错误.....	156	12.1.2 JDBC 的基本功能.....	250
8.3 “记事本”实例.....	157	12.1.3 JDBC 连接数据库的方式.....	251
8.3.1 文件类(File 类).....	158	12.1.4 JDBC API 的类和接口	252
8.3.2 文件类的属性和操作.....	159	12.1.5 数据库编程的一般过程	255
8.3.3 文件的读写.....	161	12.2 “用户注册”实例.....	256
本章小结.....	164	12.2.1 数据库的创建	258
思考与练习.....	165	12.2.2 数据库的连接	259
第 9 章 图形用户界面编程	168	12.2.3 数据库的操作	260
9.1 “注册界面设计”实例.....	168	12.2.4 TableModel 接口.....	262
9.2 “系统登录功能设计”实例.....	204	本章小结.....	262
本章小结.....	213	思考与练习.....	262
思考与练习.....	213	第 13 章 网络编程基础	264
第 10 章 图形与多媒体处理	216	13.1 “访问搜狐网络资源”实例.....	264
10.1 “网页向 Applet 传值”实例.....	216	13.2 “聊天室”实例.....	273
10.1.1 Applet 简介.....	217	本章小结.....	279
10.1.2 Applet 程序的生命周期.....	218	思考与练习.....	279
10.1.3 Applet 程序的建立与运行....	219	附录 A Java 关键字	282
10.1.4 HTML 与 Applet 程序	220	附录 B 运算符的优先级和结合方向	283
10.2 “画笔功能”实例.....	221	附录 C 各章思考与练习参考答案	285
10.2.1 字体与颜色.....	222	参考文献	306
10.2.2 绘制基本图形.....	223		
10.3 “幻灯片播放”实例.....	224		
10.3.1 图片的处理.....	226		

第 1 章 Java 语言概述

教学目标与要求:

本章从一个最简单的 Java 程序入手, 介绍两种 Java 程序的基本结构及编译运行的过程, 进而介绍 Java 开发环境的安装、配置及使用, 同时简要介绍 Java 集成开发工具 Eclipse。通过本章的学习, 读者应该掌握以下内容:

- Java 语言的特点。
- Java 两种程序的开发方法、过程和各自的特点。
- Java 开发环境的安装和配置。
- Java 集成开发工具 Eclipse 的使用。

教学重点与难点:

Java 程序的分类; 两种 Java 程序的开发方法、过程和各自的特点; Java 开发环境的配置和使用。

1.1 “初识 Java——Hello World” 实例

1. 实例目的

- (1) 掌握 Java 语言的分类。
- (2) 掌握 Java 程序的分类。
- (3) 掌握 Java 程序的开发过程。

2. 实例内容

通过 Java 语言编写一个控制台应用程序, 通过 Java 编译器进行编译, 运行后在控制台输出 “Hello World”。

3. 技术要点

创建一个名为 “HelloWorld.java” 的 Java 应用程序, 需要使用 JDK 中的 Java 编译器对其进行编译。为了使读者更清楚地了解程序的执行过程, 在控制台界面下进行编译和运行。具体步骤如下。

(1) 在文本文件中编辑完成 Java 源程序

- ① 在 C 盘新建 java 目录, 在该目录下新建文件名为 HelloWorld.txt 的文本文件。
- ② 在文本文件中输入程序代码, 如图 1.1 所示。

③ 代码输入完成后, 先进行保存, 然后修改文本文件的后缀名, 即由 HelloWorld.txt 改为 HelloWorld.java。

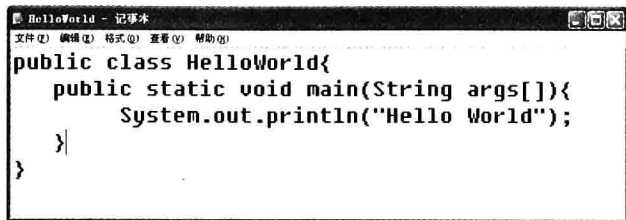


图 1.1 在文本文件中输入代码

(2) 编译 Java 程序

① 从电脑桌面依次单击“开始”→“运行”，弹出“运行”对话框。在“打开”组合框中输入“cmd”，按 Enter 键，就进入控制台命令窗口，如图 1.2 所示。

② 通过“cd java”命令进入到 Java 源代码所在的目录。

③ 输入“javac HelloWorld.java”命令，然后按 Enter 键，如果程序编译成功，则在文件夹下生成一个 HelloWorld.class 类文件，如图 1.3 所示。

④ 在控制台窗口中接着输入“java HelloWorld”，按 Enter 键运行，控制台窗口中将输出程序的运行结果，显示“Hello World”字符串。

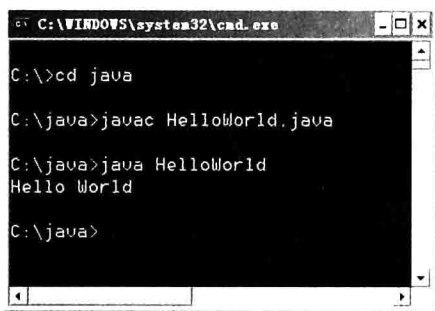


图 1.2 控制台命令窗口



图 1.3 编译后获得类文件

1.1.1 Java 语言的发展历程

Java 是一种可以撰写跨平台应用软件的面向对象的程序设计语言，是 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE、Java EE、Java ME)的总称。

Java 语言的前身是 Oak 语言，是 Sun 公司在 1991 年由一个叫 Green 的项目组专门为家用消费电子产品开发的一个分布式代码系统，主要用来与家电产品如电视机、冰箱、空调等进行信息交互和控制。由于商业原因，Oak 语言一直没有被推广出去。

1994 年，随着互联网的快速发展，Sun 公司的 Bill Joe 参加了 Green 小组，决定将 Oak 语言应用于互联网的万维网的发展中，并编制了 HotJava 浏览器。通过 HotJava 浏览器在互联网上的应用，充分展现了 Oak 语言的网络方向的特点。同时 Bill Joe 决定通过互联网让世界上的无数的软件开发人员可以免费使用 Oak 语言，这更加推动了 Oak 语言的发展。

1995 年，Sun 公司将 Oak 语言正式命名为 Java，同时召开了 Java 语言的发布会，推出了 Java 和 HotJava 的最初版本。

1998 年，Sun 公司发布了 Java 2 企业平台 J2EE。

1999 年, Sun 公司发布了 Java 的三个版本, 即标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。

2000 年, Sun 公司发布了 JDK 1.4。

2005 年, 在 JavaOne 大会上, Sun 公司发布了 Java SE 6。此时, Java 的各种版本已经更名, 以取消其中的数字“2”, 即 J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2009 年, Oracle 公司以 74 亿美元收购 Sun 公司, 取得 Java 的版权。

2011 年, Oracle 公司发布了 Java 7 的正式版。

Java 平台由 Java 虚拟机和 Java 应用编程接口构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口, 可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个 Java 平台之后, Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只编译一次, 可以在各种系统中运行。

Java 分为三个体系, 即 Java 平台标准版、Java 平台企业版和 Java 平台微型版。

Java 技术具有卓越的通用性、高效性、平台移植性和安全性, 广泛应用于个人计算机、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专业社群。在全球云计算和移动互联网的产业环境下, Java 更具备了显著的优势和广阔的发展前景。

1.1.2 Java 语言的特点

Java 语言是当前最流行的网络编程语言, 与 C++ 语言相比, 有着特殊的优点, 如简单性、平台无关性、面向对象、分布式、安全性、支持多线程、健壮性、动态性等。

1. 简单性

与 C++ 相比, Java 不再支持运算符重载、多级继承及广泛的自动强制等易混淆和极少使用的特性, 而增加了内存空间自动垃圾收集的功能。复杂性的省略和实用功能的增加使得程序开发变得简单而又可靠。

2. 平台无关性

平台无关性是 Java 最吸引人的地方。Java 是一种网络语言, 而网络上充满了各种不同类型的机器和操作系统。Java 采用了解释执行而不是编译执行的运行环境, 首先编译成字节码, 然后装载与校验, 再解释成不同的机器码来执行, 即“Java 虚拟机”的思想, 屏蔽了具体的平台环境的要求。

3. 面向对象

面向对象的技术具有继承性、封装性和多态性等多种优点, Java 在保留这些优点的基础上, 又具有动态编程的特性, 更能发挥出面向对象的优势。

4. 分布式

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议传送和接受信息的方法。Java 应用程序通过 URL 对象访问网络资源, 这使得程序员使用网络上的文件

就像使用本地文件一样容易。

5. 安全性

作为网络开发语言，Java 有建立在公共密钥技术基础上的确认技术，提供了足够的安全保障。Java 在运行应用程序时，严格检查其访问数据的权限，如不允许网络上的应用程序修改本地的数据。同时，Java 程序运行稳定，轻易不会出现死机现象。

6. 支持多线程

多线程机制使应用程序能同时进行不同的操作，处理不同的事件。Java 有一套成熟的同步语言，保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程来完成特定的任务。

7. 健壮性

健壮性反映出程序的可靠性。Java 的几个内置的特性使程序的可靠性得到改进。

- (1) Java 是强类型语言。编译器和类载入器保证所有方法调用的正确性，防止隐式类版本的不兼容性。
- (2) Java 没有指针，不能引用内存指针，避免了内存或数组越界访问。
- (3) Java 进行自动内存回收，编程人员无法意外释放内存，不需要判断应该在何处释放内存。
- (4) Java 在编译和运行时，都要对可能出现的问题进行检查，以消除错误的产生。另外，在编译的时候，还可揭示出可能出现但尚未被处理的异常，以防止系统的崩溃。

8. 动态性

Java 在类库中可以自由地加入新方法和实例变量，而不影响用户程序的执行；同时，Java 通过接口来支持多重继承，使其具有更灵活的方式和扩展性。

Java 语言除了具有上述主要特点外，还具有高性能、解释性和可移植性等特点。

1.1.3 Java 程序的分类

根据程序结构和运行环境的不同，Java 程序可以分为两类：Java 应用程序(Java Application)和 Java 小应用程序(Java Applet)。

应用程序以 main()方法作为程序入口，由 Java 解释器加载执行。Java 应用程序是完整的程序，能够独立运行。

而 Java Applet 小应用程序不使用 main()方法作为程序入口，需要嵌入到 HTML 网页中运行，由 Appletviewer 或其他支持 Java 的浏览器加载执行，不能独立运行。

无论是哪种 Java 源程序，都用扩展名为“.java”的文件来保存。

1. Java Application 程序

Java Application 程序是可独立运行的 JVM 程序，它由一个或多个类组成，其中必须有一个类中定义了 main()方法，main()方法就像 C 语言的 main 方法一样，是 Java Application 程序运行的入口。

编写和运行 Java Application 程序需要按下列步骤进行。

(1) 创建一个 Java Application 源程序(.java 文件)

创建一个名为 HelloWorldApp.java 的文件,可在任何字符编辑器上输入并保存下列 Java 源程序代码:

```
public class HelloWorldApp {  
    public static void main(String args[]) {  
        System.out.println("Hello Java!"); //在控制台上输出字符串"Hello Java!"  
    } //main 方法结束  
} //class 定义结束
```

其实质是创建一个名为 HelloWorldApp 的类,并把它保存在与它相同名字的文件中(即 HelloWorldApp.java 文件)。

一个 Java 源程序是由若干个类组成的,本程序中,只有一个类。class 是 Java 的关键字,用来定义类。public 也是关键字,用来声明一个类是公共类。

源文件的命名规则如下:如果一个 Java 源程序中有多个类,那么只能有一个类是 public 类;如果有一个类是 public 类,那么 Java 源程序的名字必须与这个类的名字完全相同,扩展名是“.java”;如果源文件中没有 public 类,那么源文件的名字只要与某个类的名字相同,而且扩展名是“.java”就可以了。

应用程序的入口是 main()方法,它有固定的书写方式:

```
public static void main(String args[]) {  
    ...  
}
```

main()方法之后的大括号及括号内的内容叫作方法体。一个 Java 应用程序必须有且仅有一个类含有 main()方法,这个类称为应用程序的主类。public、static 和 void 用于对 main()方法进行声明。在一个 Java 应用程序中,main()方法必须被声明为 public、static 和 void,public 声明 main()是公有的方法,static 声明 main()是一个类方法,可以通过类名直接调用,而 void 则表示 main()方法没有返回值。

在定义 main()方法时,String args[]用来声明一个字符串类型的数组 args,它是 main()方法的参数,用来接收程序运行时所需要的参数。

上例的 main 方法中只有一条语句:

```
System.out.println("Hello Java!"); //在控制台上输出字符串"Hello Java!"
```

这个语句是把字符串“Hello Java!”输出到系统的标准输出上,例如系统屏幕。其中 System 是系统类的对象,out 是 System 对象中的一个对象,表示“标准输出”,println 是 out 对象的一个方法,其作用是在系统标准输出上显示形参里指定格式的字符串,并回车换行。“//”代表注释,用来说明这一条语句的功能,注释主要用来提高程序的可读性,不会参与程序的编译。

(2) 对已创建好的 Java 源程序进行编译

该步骤是用 Java 编译器对 Java 源程序(.java)进行编译,生成对应的字节码文件(.class)。如果编译成功,会产生一个有相同文件名的带.class 扩展名的字节码文件。



其命令格式如下:

```
javac HelloWorldApp.java
```

如果编译中不出现错误,将会得到一个名为 HelloWorldApp.class 的文件。编译选项使用缺省方式。

(3) 解释执行已编译成功的字节码文件(.class 文件)

用 Java 解释器对 Java 字节码文件(.class 文件)解释执行。

在上例得到的 HelloWorldApp.class 文件,现在可以用 Java 解释器执行:

```
java HelloWorldApp
```

运行的结果,将会在标准输出设备上输出:

```
Hello Java!
```

Java 解释器在解释执行时,解释处理的是类名,而不是文件名,所以在解释器 java 后面跟随的是类名,而不能写成文件名的形式(HelloWorldApp.class)。其选项也是使用缺省的方式。

2. Java Applet 程序

一个 Java Applet 也是由若干个类组成的,一个 Java Applet 不再需要 main()方法,但必须有且仅有一个类扩展了 Applet 类,即它是 Applet 类的子类,这个类称为 Java Applet 的主类,Java Applet 的主类必须是 public 的,Applet 类是系统提供的类。

Applet 与 Application 的区别在于其执行方式不同。Application 是从 main()方法开始运行的,而 Applet 是在浏览器中运行的,必须创建一个 HTML 文件,通过编写 HTML 代码告诉浏览器载入何种 Applet 以及如何运行。

开发 Java Applet 程序的步骤如下。

首先,编写 Applet 源程序,将其保存为“.java”文件。

编译源程序,生成字节码文件“.class”。如果源文件包含了多个类,则会生成多个扩展名为“.class”的文件,都与源文件存放在相同的目录下。如果对源文件进行了修改,那么必须重新进行编译,再生成新的字节码文件。

编写一个 HTML 文件,即含有 applet 标记的 Web 页,嵌入 Applet 字节码文件“*.class”。最后,运行 Java 小应用程序。

【例 1-1】开发一个输出“Hello Java !!”的 Java 小应用程序。

(1) 以文件名“HelloWorldApplet.java”保存 Java Applet 源程序。程序代码如下:

```
//文件 HelloWorldApplet.java
import java.awt.*;           //引入 java.awt 包中的类
import java.applet.*;        //引入 java.applet 包中的类
public class HelloWorldApplet extends Applet { //继承 Applet
    public void paint(Graphics g) { //重写 paint 方法
        //在 (50, 40) 位置输出字符串
        g.drawString("Hello Java !!", 50, 40 );
    }
}
```

(2) 编写嵌入字节码文件的 HTML 文件 “HelloWorldApplet.html”：

```
<html> <!--标识 HTML 文件的开始-->
<!--告诉浏览器将运行一个 Java Applet-->
<applet code="HelloWorldApplet.class" width="200" height="80"></applet>
</html> <!--标识 HTML 文件的结束-->
```

(3) 使用 JDK 编译 Java 小应用程序：

```
javac HelloWorldApplet.java
```

(4) 使用 JDK 提供的 Appletviewer 运行：

```
appletviewer HelloWorldApplet.html
```

Java Applet 程序必须创建一个 Applet 或 JApplet 的子类。Applet 程序中不需要有 main() 方法。

1.1.4 Java 程序的开发过程

Java 语言包括 3 种核心机制：Java 虚拟机、垃圾收集机制和代码安全检测。Java 程序的开发过程大致分为 3 个阶段。

(1) 编写 Java 源文件。将编辑好的源程序以扩展名“.java”保存起来，即保存成“*.java”文件。

(2) 编译 Java 源程序。使用 Java 编译器编译“*.java”源程序，从而得到字节码文件“*.class”。

(3) 运行 Java 程序。

Java 程序的开发流程如图 1.4 所示。

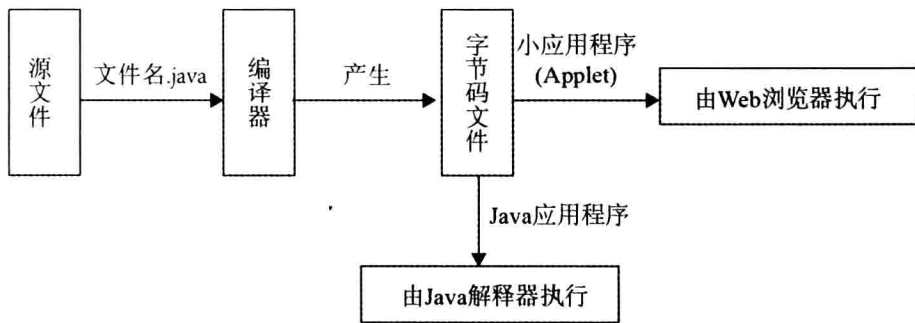


图 1.4 Java 程序的开发流程

从图 1.4 中可以看出，一个源文件首先应保存为扩展名为.java 的文件，通过编译器产生字节码文件。

如果编写的是小应用程序(Java Applet)，可以直接由浏览器解释运行，对于 Java 应用程序，就会由 Java 解释器来执行。

在本书中，我们只有在后面才会接触到小应用程序，前半部分还是以应用程序为主，也就是说，我们使用的是 Java 解释器。

1.2 Java 的开发环境

1.2.1 JDK 的安装

1. 什么是 JDK

JDK 的全名是 Java Development Kit，是 Sun 公司开发的提供给程序设计人员用于开发 Java 程序的一组工具，目前的版本是 1.7。JDK 是整个 Java 的核心，包括了 Java 运行环境、Java 工具和 Java 基础类库。了解 JDK 是学好 Java 的第一步。

对于 JDK 内所提供的工具，本书中要用到的有 `javac.exe` 和 `java.exe` 两个程序。

`javac.exe` 是 Java 语言的编译器，用于将 Java 程序编译成字节码文件，`java.exe` 则是 Java 语言的解释器，用于执行 Java 程序的字节码。

2. JDK 的下载

JDK 可以从 Oracle 网站免费下载得到，下面以本书要用到的 JDK 7 版本为例，介绍下载过程。

在浏览器中键入如下网址：

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
进入网页找到下载链接，下载对应平台的合适 JDK，如图 1.5 所示。

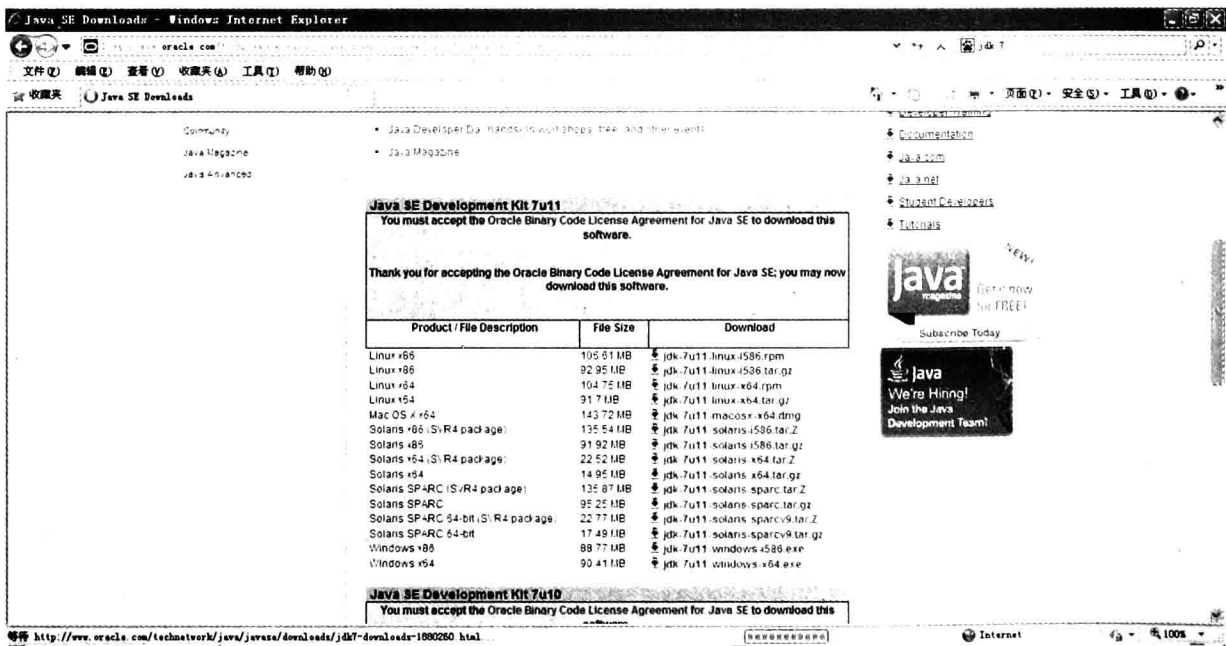


图 1.5 JDK 下载网页

3. JDK 的安装

完成 JDK 安装文件的下载后，双击运行该文件进行安装。

(1) 进入安装向导，如图 1.6 所示。

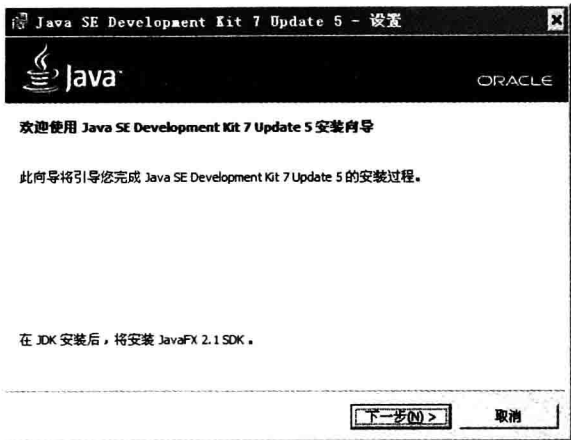


图 1.6 JDK 安装向导

(2) 单击“下一步”按钮，更改安装路径，选择安装所有组件，如图 1.7 所示。

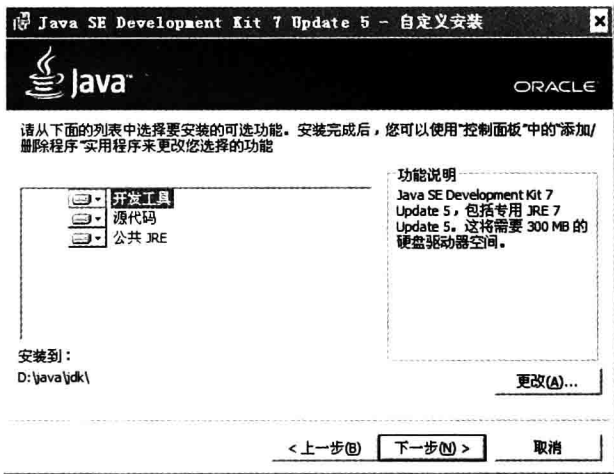


图 1.7 选择 JDK 安装路径和组件

(3) 单击“下一步”按钮，开始安装。出现安装进度界面，如图 1.8 所示。

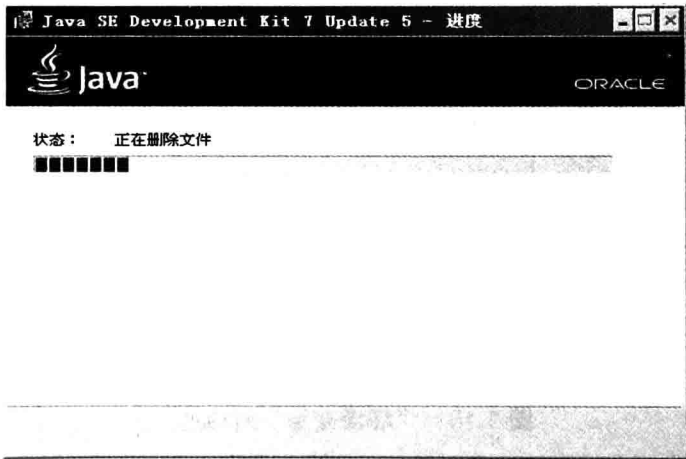


图 1.8 JDK 安装进度

(4) 更改 JRE 安装路径。如图 1.9 所示。

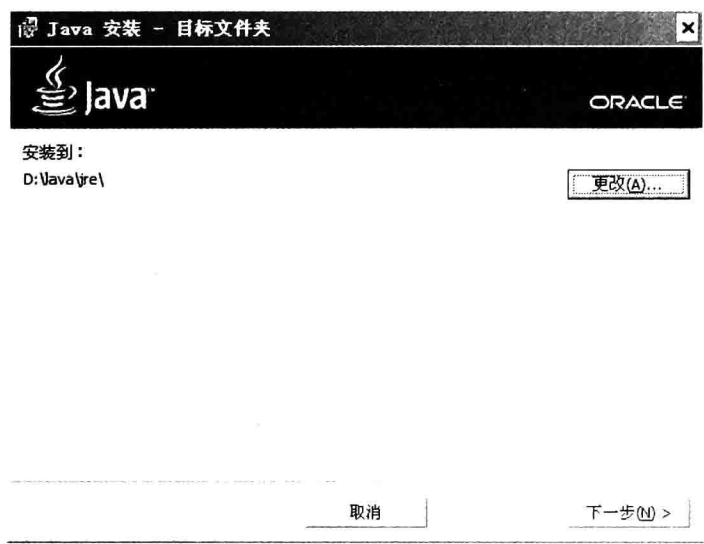


图 1.9 修改 JRE 安装路径

(5) 单击“下一步”按钮，完成 JRE 的安装。

1.2.2 JDK 的配置

JDK 环境变量的设置主要是设置变量 Path 和 Classpath。设置 Path 变量的目的是在任何路径下都可以运行 JDK 的命令，设置 Classpath 变量的目的是让应用程序可以找到所需要的类包。下面以 Windows XP 为例，来说明 Path 和 Classpath 变量的设置过程。

1. Path 变量的设置

- (1) 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。
- (2) 在弹出的“系统属性”对话框中选择“高级”选项卡，单击其中的“环境变量”按钮，出现“环境变量”对话框，如图 1.10 所示。

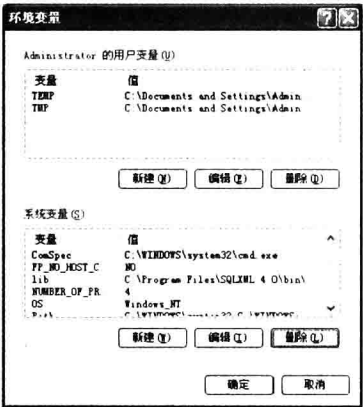


图 1.10 “环境变量”对话框

(3) 在“系统变量”列表框内找到 Path 变量。若没有 Path 变量，则单击“新建”按钮，创建 Path 变量。单击“编辑”按钮或双击 Path 变量，弹出如图 1.11 所示的“编辑系统变量”

对话框。在“变量值”文本框的尾部添加“D:\Java\jdk\bin”，这里的“D:\Java\jdk”为 JDK 的安装路径，单击“确定”按钮结束。

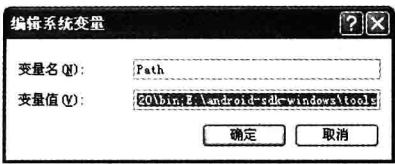


图 1.11 “编辑 Path 系统变量”对话框

2. Classpath 变量的设置

(1) 根据 Path 变量的设置步骤，打开“环境变量”对话框，在“系统变量”列表框中找到 Classpath 变量。如果不存在 Classpath 变量，单击“新建”按钮创建 Classpath 变量。然后选择 Classpath 项，单击“编辑”按钮或者双击 Classpath 变量，出现“编辑系统变量”对话框，如图 1.12 所示。

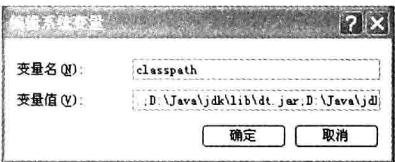


图 1.12 “编辑系统变量”对话框

(2) 在“变量值”文本框中添加“.;D:\Java\jdk\lib\dt.jar;D:\Java\jdk\lib\tools.jar”，单击“确定”按钮结束。

3. 环境变量的设置和测试

进入控制台界面，在控制台界面下输入 javac 和 java 指令，按 Enter 键，若能成功显示 javac 和 java 指令的用法及所包含参数的类型和含义，则说明 Path 变量和 Classpath 变量配置成功。

1.2.3 JDK 的使用

本小节主要介绍 JDK 开发包中的主要工具及其使用方法，属于技术细节问题，读者可以先跳过这里的大部分细节内容，当要用到这些工具的时候再查阅相关部分。

1. Java 编译器(javac.exe)

顾名思义，Java 编译器的作用就是将 Java 源代码(.java 文件)转换为字节码(.class 文件)。其实，Java 编译器本身就是一个用 Java 语言编写的应用程序。

Java 编译器可以同时编译多个源程序，输出结果为类文件(字节码文件)。对于源程序中每个类的定义，它都会产生一个独立的类文件。

Java 编译器的使用格式如下：

```
javac [options] source_files
```