

```

import java.util.*;

public class ArrayApp {

    public void printArray(char [] firstArray, String [] secondArray){
        for(int i=0;i<firstArray.length;i++){ //普通 for 循环遍历数组
            System.out.println(firstArray[i]);
        }
        for(int i=0;i<secondArray.length;i++){
            System.out.println(secondArray[i]);
        }
        for(char c:firstArray){ //for 循环增强，循环因子 c 顺序取出数组元素的内部
            System.out.println(c);
        }
        for(String b:secondArray){
            System.out.println(b);
        }
        String s=Arrays.toString(secondArray);
        System.out.println(s);
        System.out.println(firstArray); //输出数组 firstArray 的
        System.out.println(""+firstArray); //输出数组 firstArray 的引用地址
    }
}

```



普通高等教育“**十 三 五**”规划教材

Java 语言 程序设计教程

Java YUYAN CHENGXU SHEJI JIAOCHENG

胡 光 主 编
李效伟 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等教育“十三五”规划教材

Java 语言程序设计教程

胡 光 主 编

李效伟 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书从 Java 语言程序设计基础入手,以面向对象程序设计思想为主线,循序渐进地讲述 Java 程序设计理论与应用知识。全书内容主要包括 Java 语言的基础语法、面向对象程序设计思想与应用、Java 语言的常用类、异常处理机制、I/O 流、Java 语言的常用数据结构及其实现、GUI 程序设计、多线程技术和网络程序设计基础等。

本书内容丰富,强调理论与实践相结合。书中所有内容均遵守 Java SE 平台的程序设计规范,实践内容实用性强。书中正文力求简明和通俗易懂,章节结构清晰,既包含 Java 语言程序设计的基础知识,还包含深入的高级主题,每章都附有贴合正文内容的编程习题,便于读者复习和熟练掌握 Java 程序设计知识并应用到项目开发过程中。

本书适合作为计算机科学与技术 and 软件工程相关专业的 Java 编程基础教材,也适合 Java 语言程序设计的初学者和已经具备编程基础的技术人员使用。

图书在版编目(CIP)数据

Java 语言程序设计教程/胡光主编. —北京:中国铁道出版社,2018.1

普通高等教育“十三五”规划教材

ISBN 978-7-113-24096-7

I. ①J… II. ①胡… 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 325729 号

书 名: Java 语言程序设计教程

作 者: 胡 光 主编

策 划: 潘晨曦

读者热线: (010) 63550836

责任编辑: 李露露 徐盼欣

封面设计: 刘 颖

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.tdpress.com/51eds/>

印 刷: 三河市宏盛印务有限公司

版 次: 2018年1月第1版 2018年1月第1次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 21 字数: 512 千

书 号: ISBN 978-7-113-24096-7

定 价: 56.80 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

随着计算机与信息技术的不断发展,学习一门应用广泛的程序设计语言是适应 IT 编程行业发展的必选之路。2015 年 7 月 4 日,国务院印发了《国务院关于积极推进“互联网+”行动的指导意见》。从该指导意见可以看出,选择一种与互联网及其应用紧密联系、跨平台和移植性强的程序设计语言符合软件开发的发展要求,并能够推动软件开发领域的不断拓展。

Java 语言最初是基于互联网技术的应用而产生的,恰好适应了国家关于 IT 行业不断发展的战略需要。经过数年的发展,Java 语言已经成为一门成熟的面向对象的程序设计语言,在所有程序设计语言中占有举足轻重的地位。从面向对象程序设计思想的角度来说,Java 语言在网络、跨平台、多线程和安全性方面比其他程序设计语言更具有优势,更适合程序设计人员作为开发工具使用。从面向对象程序设计的应用角度来说,Java 语言将大部分算法实现进行了封装,实现比较容易,可以节省程序设计人员的开发时间,提高开发效率。

本教材以面向对象程序设计思想为主线,将所有内容都围绕“封装、继承与多态”展开,所有代码示例严格符合编程规范。Java 程序设计的基础知识主要以封装类中的数据属性、构造方法和功能方法作为主线进行讲述,侧重封装、继承与多态之间的相互关系。Java 程序设计应用中的 GUI 程序以三个顶层容器组件为主线进行讲述,侧重桌面程序的设计。Java 程序设计应用中的数据结构、输入输出、网络应用以典型案例为主线进行讲述,侧重完整项目的开发与设计。

全书共分 10 章,各章节之间既有相互独立性又有潜在的联系,保证了学习的连贯性。第 1 章对 Java 与面向对象思想之间的关系进行了简述。第 2 章介绍了基本数据类型、数据字段、操作符、表达式和语句流程控制等程序设计基础知识。第 3 章详细讲解了面向对象程序设计的特征,学习使用类、对象、接口、类的继承、多态进行程序设计。第 4 章在引入 Java 在线帮助文档的前提下,介绍常用的 Java 类及其应用。第 5 章讲述了 Java 语言的异常处理机制。第 6 章讲述了数据持久化中输入流和输出流封装的知识。第 7 章介绍了基于 Java 语言的常用数据结构的实现和 Java 集合框架,它为数据结构定义了一套有用的 API。第 8 章介绍了 Java GUI 程序设计和 Java Applet 的应用,侧重学习委托事件处理模型,使程序具有更好的响应和人机交互功能。第 9 章介绍了 Java 多线程技术,它实现了让程序任务并发执行的特征及安全性。第 10 章讲解了 Java 程序在网络方面的应用,使得网络上的不同主机能够相互通信。

在教材内容方面,由浅入深进行讲述,如果读者把此书作为入门教材,建议熟练学习前 8 章的内容,培养描述实际问题的程序化解决方案的关键技能,并通过基本算法和数据结构将方案转

变成程序。如果读者已经具有一定的编程基础，建议重点学习第9章和第10章，使用面向对象程序设计的方法开发多线程并发任务程序或网络程序，提高程序设计的项目应用能力。在实践方面，本教材并没有局限于讲述某一种具体的程序设计集成开发环境，让读者能够按照自己的实际要求进行编程，提高了教材使用的通用性。教材每一章都安排了适量的练习题，给读者增加自主性思考和实践练习的机会。这些练习题不再使用实验指导书性质的基于现有代码程序设计，而是要求读者根据题目文字描述，自行分析和设计。练习题既有简单的程序设计题，也有复杂的算法实现题，还有完整的项目实践题，让读者循序渐进地学习 Java 语言程序设计。

Java 语言程序设计的实践练习过程中，提供了 Java SE 各种版本的在线帮助文档，即 Java SE API。Java SE API 可以帮助读者理解教材中的实践内容，提高程序设计的效率。本教材并没有简单地套用 Java SE API 中的内容，而是在该文档的基础上增加程序设计原理、经验和技巧，力求比在线帮助文档更加准确、简明和便于理解。

与其他同类教材相比，本教材的主要特色为：

(1) 更加强化面向对象程序设计中的封装、继承和多态特征在程序中的体现，所有的例题和练习题都力求用严格的编程规范实现。

(2) 在程序设计基础知识学习的前提下，重点提升编程能力，重视逻辑思维和计算思维的培养。

(3) 重视理论与实践相结合，突出程序设计动手能力的培养的特点。摒弃了传统实验指导书提供部分代码的问题，重点培养完整程序编写的能力。每章后的习题都需要设计编程思想和完成完整的程序代码，提高程序设计的熟练度。

(4) 重视扩展内容深度和广度，既可以把本教材作为 Java 语言程序设计的基础教材，也可以作为深入学习 Java 语言程序设计的工具书。

本教材由烟台大学计算机与控制工程学院胡光任主编，山东女子学院信息技术学院李效伟任副主编。李效伟编写了第1章和第2章，胡光编写了其余章节。在编写过程中许多领导和老师提出了宝贵的意见和建议，也得到了许多朋友的帮助和支持，在这里一并表示感谢。

本教材适合作为计算机科学与技术 and 软件工程相关专业的程序设计基础教材，也可以作为需要学习 Java 程序设计基础和进阶的工程技术人员自学参考书。

由于编者的水平有限，不足和疏漏之处在所难免，欢迎广大技术专家和读者指正。

编 者

2017 年 11 月

第 1 章 绪论	1	2.2.5 赋值操作符和赋值 表达式	24
1.1 Java 概述	1	2.2.6 条件操作符和条件 表达式	25
1.1.1 Java 语言的产生与发展	1	2.2.7 其他操作符和相关 表达式	25
1.1.2 Java 语言的特点	1	2.2.8 操作符的优先级和复杂 表达式	26
1.1.3 Java 语言的开发与运行 平台	3	2.3 控制流语句	27
1.2 Java SE 开发环境	3	2.3.1 顺序结构语句	27
1.2.1 Java SE 的下载、安装与 环境变量的配置	3	2.3.2 条件控制语句	27
1.2.2 Java 语言程序设计工具	4	2.3.3 循环控制语句	33
1.3 Java 语言的程序设计过程	5	2.3.4 跳转语句	38
1.3.1 Java 应用程序的设计 过程	5	2.4 递归	40
1.3.2 Java 小程序的设计过程	8	习题	41
1.3.3 Java 语言程序注释	10	第 3 章 面向对象程序设计	43
习题	11	3.1 概述	43
第 2 章 程序设计基础	12	3.1.1 面向对象程序设计的基本 概念	43
2.1 标识符、关键字和数据类型	12	3.1.2 面向对象程序设计的 特点	45
2.1.1 标识符和关键字	12	3.2 类和对象	47
2.1.2 基本数据类型	13	3.2.1 类	47
2.1.3 常量和变量	14	3.2.2 对象	54
2.1.4 基本数据类型的转换	17	3.2.3 包	60
2.2 操作符和表达式	19	3.3 继承	64
2.2.1 算术操作符和算术 表达式	19	3.3.1 概述	64
2.2.2 关系操作符和关系 表达式	20	3.3.2 抽象类和最终类	74
2.2.3 逻辑操作符和逻辑 表达式	20	3.3.3 对象的引用转型	76
2.2.4 位操作符和位操作 表达式	21	3.4 接口	81
		3.4.1 概述	81
		3.4.2 接口的回调	85

3.5 嵌套类和匿名类	87	第 6 章 输入流和输出流	151
3.5.1 嵌套类	87	6.1 概述	151
3.5.2 匿名类	89	6.1.1 流	151
3.6 Java 面向对象思想进阶	92	6.1.2 字节流	152
3.6.1 反射	92	6.1.3 字符流	153
3.6.2 注解	93	6.2 文件输入流和输出流	154
3.6.3 Java 应用程序常用工具	94	6.2.1 文件类	154
习题	96	6.2.2 文件输入字节流和输出 字节流	158
第 4 章 常用类	98	6.2.3 文件字符输入流和字符 输出流	165
4.1 字符串	98	6.3 标准输出流和标准输入流	168
4.1.1 String 类	98	6.3.1 标准输出流	168
4.1.2 StringBuffer 类和 StringBuilder 类	103	6.3.2 格式化输出	169
4.1.3 正则表达式	104	6.3.3 标准输入流	171
4.2 数组	111	6.3.4 格式化输入	171
4.2.1 一维数组	111	6.3.5 标准输入流和标准输出流 的重定向	173
4.2.2 多维数组	121	6.4 其他输入流和输出流	175
4.3 数据类型类、日期和时间类	124	6.4.1 RandomAccessFile 类	175
4.3.1 数据类型类	124	6.4.2 字节流和字符流的综合	176
4.3.2 System 类	127	6.4.3 对象流	178
4.3.3 Math 类	128	习题	180
4.3.4 日期和时间类	129	第 7 章 常用数据结构	182
4.4 泛型	133	7.1 集合	182
4.4.1 泛型类	134	7.1.1 Collection<E>接口	183
4.4.2 泛型接口	135	7.1.2 Collections 类	184
4.5 枚举	136	7.2 链表	185
4.5.1 枚举的创建	136	7.2.1 链表接口及常用实现类	186
4.5.2 枚举的使用	137	7.2.2 链表类 LinkedList<E>	187
习题	138	7.2.3 链表类 ArrayList<E>	192
第 5 章 异常处理	140	7.3 堆栈 Stack<E>	194
5.1 概述	140	7.3.1 向量 Vector<E>	194
5.1.1 程序中的问题	140	7.3.2 堆栈 Stack<E>	195
5.1.2 异常类	141	7.4 队列	196
5.2 异常处理概述	144	7.5 集合 Set<E>	199
5.3 自定义异常	147	7.5.1 HashSet<E>类	199
5.4 断言	149	7.5.2 TreeSet<E>类	201
习题	150		

7.6 映射 Map<K,V>	202	9.1.1 基本概念	281
7.6.1 HashMap<K,V>类	202	9.1.2 线程的执行	282
7.6.2 TreeMap<K,V>类	204	9.2 线程	283
习题	206	9.2.1 线程的创建	283
第 8 章 图形用户界面	208	9.2.2 线程的功能与实现	288
8.1 概述	208	9.2.3 线程的状态和生命 周期	290
8.2 容器类	210	9.2.4 线程的调度和优先级	292
8.2.1 窗口框架	211	9.2.5 线程组	293
8.2.2 面板	213	9.2.6 线程的动画应用	294
8.2.3 其他容器	214	9.3 多线程同步机制	296
8.3 原子组件与布局	219	9.3.1 概述	296
8.3.1 原子组件	219	9.3.2 多线程同步的调度	302
8.3.2 布局管理器	223	习题	306
8.4 事件处理	229	第 10 章 网络程序设计	308
8.4.1 委托事件处理模型	230	10.1 概述	308
8.4.2 动作事件	231	10.1.1 TCP/IP	308
8.4.3 选项事件	233	10.1.2 IP 地址与通信端口	309
8.4.4 文本插入符事件	236	10.1.3 URL 的基本概念	310
8.4.5 窗口事件	238	10.1.4 Java Web 开发	310
8.4.6 鼠标事件	240	10.2 基于 TCP 的 Java 网络程序 设计	313
8.4.7 焦点事件和键盘事件	243	10.2.1 Socket 类和 ServerSocket 类	313
8.4.8 系统托盘图标支持	246	10.2.2 基于 TCP 的 Java 网络 程序设计过程	314
8.4.9 GUI 程序设计过程	247	10.3 基于 UDP 的 Java 网络程序 设计	322
8.5 对话框	248	10.3.1 DatagramPacket 类和 DatagramSocket 类	323
8.5.1 标准对话框	248	10.3.2 基于 UDP 的 Java 网络 程序设计过程	324
8.5.2 自定义对话框	258	习题	326
8.6 Java Applet	259	参考文献	327
8.6.1 概述	259		
8.6.2 Applet 工作原理	262		
8.6.3 Java Applet 的用户界面 ..	268		
8.6.4 Java Applet 应用	273		
习题	278		
第 9 章 多线程	281		
9.1 概述	281		

第 1 章 | 绪 论

随着计算机科学技术的发展,计算机硬件和软件技术已经渗入经济产业的各个领域。相对于计算机硬件而言,计算机软件是发挥计算机潜力的重要技术工具。要实现计算机硬件的潜力,就必须对计算机软件进行程序设计和开发。为了使程序开发更加方便,计算机程序设计语言的发展方向也朝着编程人员的思维习惯方面靠近。正是由于 Java 语言非常符合这种发展趋势,目前它已经成为应用非常广泛的程序设计语言之一。

1.1 Java 概述

与现实世界中的其他事物一样,计算机程序设计语言的发展也是不断变化的。目前,计算机程序设计行业中主要有面向过程程序设计和面向对象程序设计两大分支。在面向过程程序设计思想中,程序由在数据上运行的过程和功能的集合组成,许多传统程序设计语言(Fortran、Pascal 和 C 等)体现的都是面向过程思想。在面向对象程序设计思想中,程序被看作“对象”的集合,数据和相关操作被封装在某个单元里,面向对象程序设计语言有 Smalltalk、C++、Python 和 Java 等。

1.1.1 Java 语言的产生与发展

1991 年,Sun 公司设计了一种能够适用于编写嵌入式系统电子设备微处理器的程序设计语言,由 James Gosling 负责的编程团队对其进行了开发。与此同时,随着 Internet 的迅速普及,该语言被重新设计和编写,并于 1995 年 Sun World 95 大会上正式公布为 Java 语言。Java 语言的产生,无论在理论计算领域、商业计算领域还是网络计算领域都开辟了计算机程序设计的新纪元,James Gosling 也被业内称为 Java 语言之父。

从 Java 语言产生以后,它一直处于计算机程序设计语言的主流位置。1995 年 Java 语言被美国的 *PC Magazine* 杂志评为 1995 年十大科技优秀产品,许多计算机公司(如 IBM 公司和 Apple 公司等)都开始支持和开发 Java 软件产品。2009 年,Oracle 公司收购了 Sun 公司,Java 也归为 Oracle 公司所有。目前,Java 语言的开发领域遍布计算机行业的各个方面,取得了显著的业绩。

1.1.2 Java 语言的特点

Internet 和 Web 应用的发展推动了 Java 语言的不断前进,它能够在异构、分布式网络中的不同平台上开发出灵活的应用程序。

1. 简单性

从面向对象程序设计的角度来说,Java 语言与其他程序设计语言有相同的特征,但是去掉了

类似指针、多重继承等难以理解的特性,语法比较简单。Java 语言在编译和运行方面通过垃圾自动回收机制简化了程序内存管理,统一了各种数据类型在不同操作系统平台上所占用的内存大小。

2. 面向对象性

Java 语言在产生之初就致力于建立一种模拟人类思维来解决实际问题的模型,这种模型将重点放在对象和对象的实现上,非常符合人们的思维习惯,容易扩充和维护。

3. 网络特性

Java 语言具有处理网络协议的特点,能够通过 URL 打开和访问网络上的对象,其便捷程度就好像访问本地文件一样。目前,网络技术的不断发展促进了 Java 语言程序设计平台的扩展,Java 语言广泛应用到网络、移动端和嵌入式设备等方面。

4. 健壮性

Java 语言在编译和解释执行过程中都会进行严格的检查,以减少错误的发生,其中垃圾自动回收机制和异常处理机制在很大程度上保证了程序的健壮性。

5. 安全性

Java 语言设计的程序可以适用于网络和分布式系统,这就需要 Java 程序必须符合网络安全协议。在解释执行程序过程中,Java 虚拟机会针对程序的安全性进行检测,它不会访问或修改不允许访问的内存或文件。

6. 平台无关性

Java 编译器会对 Java 程序生成一个体系结构中立的目標文件格式,这是一种编译成功的代码,只要有 Java 运行时系统,就可以在不同处理器上运行同一个 Java 程序,而不需要做任何修改。Java 编译器通过生成与特定的计算机体系结构无关的字节码指令来实现这一特性。这些字节码不仅可以很容易地在任何计算机上运行,而且可以迅速地翻泽成本地计算机代码。

7. 可移植性

Java 语言在基本数据类型和引用数据类型的定义上以字节码的方式进行存储,这消除了二进制数据顺序的困扰,增强了字节码数据可移植的能力。

8. 解释型特性

Java 语言是一种解释型程序设计语言,解释器可以在任何移植了解释器的计算机上执行 Java 字节码,开发过程更加快捷。随着编译和解释能力的提高,目前,使用即时编译器可以将字节码翻泽成机器码。

9. 多线程性

多线程机制可以带来更好的交互响应和实时行为,Java 语言多线程程序设计可以利用多个处理器并发执行多任务,提高程序运行效率。

10. 动态性

Java 语言能够适应不断发展的环境,它的类库可以自由添加新方法和实例变量,而对客户端却没有任何影响,这种动态特性提高了 Java 语言封装新特性的发展速度。

1.1.3 Java 语言的开发与运行平台

计算机平台是程序在其中运行的硬件或软件环境，如 Microsoft Windows、Linux、Solaris OS 和 Mac OS 等。大多数计算机平台都可以被描述为操作系统和底层硬件的组合，而 Java 语言的开发与运行平台则是一种运行在其他基于硬件平台之上的纯软件平台。Java 平台由 Java 虚拟机 (Java Virtual Machine, JVM) 和应用程序编程接口 (Application Programming Interface, API) 两部分组成，如图 1-1 所示。

Java 语言的应用领域非常广泛，与之相对应，Java 语言的程序设计与运行平台主要有如下三个版本。

(1) Java SE (Java Standard Edition): 称为 Java 标准版本或 Java 标准平台。Java SE 提供了标准的 Java 开发工具包 JDK (Java Development Kit) 来开发 Java 桌面应用程序、某些服务器应用程序和 Java Applet 小程序等。JDK 是整个 Java 的核心，包括了 Java 运行时环境 JRE (Java Runtime Environment)、Java 工具和 Java 基础类库。

Java SE 从早期的 J2SE 1.1 版本开始一直持续更新。2004 年 9 月，J2SE 1.5 版本发布时更名为 Java SE 5.0，目前已经更新至 Java SE 8.0 系列版本，并且不断更新版本。

(2) Java EE (Java Enterprise Edition): 称为 Java 企业版本或 Java 企业平台。Java EE 不仅提供了标准的 JDK，而且增加了 Java EE 标准开发工具包 SDK (Standard Development Kit) 来开发企业级的服务应用程序。目前该版本已经更新至 Java EE 6.0 系列，并且不断更新版本。

(3) Java ME (Java Micro Edition): 称为 Java 微型版本或 Java 微型平台。Java ME 是一种微型的 Java 开发和运行时环境，主要用于各种小型或微型系统设备的产品程序开发。针对不同的系统设备，Java ME 提供了相应的 SDK 套件进行程序开发，如移动设备、嵌入式系统、Java TV 及 Java Card 等。同时，每种 SDK 套件都有相应的更新版本。

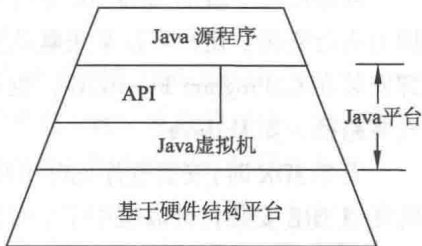


图 1-1 Java 源程序、Java 平台和底层硬件的关系图

1.2 Java SE 开发环境

Java SE 是 Java 程序程序设计的基础版本，其他 Java 程序的开发都是基于 Java SE 相关环境。因此，本书主要在 Windows 操作系统中应用 Java SE 平台进行程序开发。

1.2.1 Java SE 的下载、安装与环境变量的配置

1. Java SE 的下载

如前所述，Java SE 的版本总是不断更新，其目的是不断增加新的 Java 特性。相比以前的版本，Java SE 5.0 版本是 Java 特性变化最大的一次，所以 Java 程序开发都是以 Java SE 5.0 以后的版本为主。随着新特性的不断出现，Oracle 公司会及时发布 Java SE 的最新版本，可以登录网站 <http://www.oracle.com/technetwork/java/javase/downloads> 免费下载。Java SE 开发包下载界面如图 1-2 所示。

2. Java SE 的安装

双击已经下载的 Java SE 平台可执行文件即可进行安装。用户可以采用默认安装路径,即安装在 C:\Program Files\Java,也可以自定义安装路径,如 D:\Java。

安装 JDK 时,安装程序同时也将 Java 运行时环境 JRE 安装在计算机中了。如果用户不需要开发 Java 程序,而只想运行别人的 Java 程序,可以只安装 JRE(可以登录网站 <http://www.oracle.com/technetwork/java/javase/downloads> 免费下载)。

Java SE Development Kit 8u144		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	jdk-8u144-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u144-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.65 MB	jdk-8u144-linux-i586.rpm
Linux x86	179.44 MB	jdk-8u144-linux-i586.tar.gz
Linux x64	162.1 MB	jdk-8u144-linux-x64.rpm
Linux x64	176.92 MB	jdk-8u144-linux-x64.tar.gz
Mac OS X	226.6 MB	jdk-8u144-macosx-x64.dmg
Solaris SPARC 64-bit	139.87 MB	jdk-8u144-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.18 MB	jdk-8u144-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u144-solaris-x64.tar.Z
Solaris x64	96.99 MB	jdk-8u144-solaris-x64.tar.gz
Windows x86	190.94 MB	jdk-8u144-windows-i586.exe
Windows x64	197.78 MB	jdk-8u144-windows-x64.exe

图 1-2 Java SE 开发包下载界面

3. Java SE 帮助文档的下载

针对 Java SE 7.0 及以后版本的使用,Oracle 公司提供了相应的帮助文档。该文档可以登录 <http://www.oracle.com/technetwork/java/javase/documentation/> 免费下载,下载后进行解压缩即可。Java SE 帮助文档都是以 Web 页的形式提供浏览,用户可以单击相应链接查看帮助文档。

4. Java SE 环境变量的配置

Java SE 安装成功后,JDK 平台提供的 Java 程序命令并不是在任何路径下都可以运行,Java 类库并不能随时加载。为了能在任何路径下使用 Java 命令和随时加载 Java 类库,应该设置环境变量。

(1) 环境变量 Path 值的配置。在 Windows 7 及以后版本的 Windows 操作系统中,右击“计算机”图标,在弹出的快捷菜单中选择“属性”命令,在弹出的对话框中选择“高级系统设置”选项,在弹出的“系统属性”对话框中选择“高级”选项卡。单击“环境变量”按钮,在“系统变量”中选中 Path,单击“编辑”按钮,在“变量值”中的最后位置输入一个分号后,如果用户采用默认安装路径则接着输入内容 C:\Program Files\Java\jdk1.6.0_22\bin,如果用户采用自定义安装路径(如 D:\Java),则接着输入内容 D:\Java\jdk\bin。

(2) 环境变量 ClassPath 值的配置。在 Windows 7 及以后版本的 Windows 操作系统中,右击“计算机”图标,在弹出的快捷菜单中选择“属性”命令,在弹出的对话框中选择“高级系统设置”选项,在弹出的“系统属性”对话框中选择“高级”选项卡。单击“环境变量”按钮,在“系统变量”中选中 ClassPath,单击“编辑”按钮,在变量值中的最后位置输入一个分号后,如果用户采用默认安装路径则接着输入内容 C:\Program Files\Java\jdk1.6.0_22\jre\lib\rt.jar;.,如果用户采用自定义安装路径(如 D:\Java),则接着添加内容 D:\Java\jdk1.6.0_22\jre\lib\rt.jar;。其中,“;”为两个路径的分隔符,后面的“.”表示当前路径,一定不能缺少。如果在“系统变量”中没有 ClassPath,则单击“新建”按钮,在弹出的对话框中,输入“变量名”的值:ClassPath。

此时,就可以编写 Java 源代码,并进行编译和运行 Java 程序了。

1.2.2 Java 语言程序设计工具

设计 Java 语言程序首先是编写 Java 源代码,而选择适当的编辑器是提高程序设计效率的第一步。目前 Java 语言程序设计的工具比较多,主要有文本编辑器和集成开发环境两大类。

1. 文本编辑器

对于开发 Java 程序的初学者,通常采用无格式的文本编辑器。这种无格式的文本编辑器可以

让编程者发现常见的编程错误，以巩固编程者的程序开发基础，提高 Java 编程水平。比较常用的无格式 Java 文本编辑器有 Windows 记事本、EditPlus 及 UltraEdit 等。

2. 集成开发环境

对于具有一定程序设计经验的 Java 编程者，通常采用集成开发环境（Integrated Development Environment, IDE）。这种 IDE 将各种开发过程中所需的工具集成到一起，从而使程序设计需要的整合过程更加方便，同时可以随时增加相应的开发插件，以扩展 IDE 的开发功能和提高 IDE 的开发效率。比较常用的 Java 程序设计 IDE 有 Eclipse、MyEclipse 或 NetBeans 等。其中，Eclipse IDE 可以登录到网站 <http://www.eclipse.org/downloads/> 免费下载，NetBeans IDE 可以登录到网站 <http://netbeans.org/downloads/index.html> 免费下载。

1.3 Java 语言的程序设计过程

Java 语言的程序设计主要有 Java 应用程序（Java Application）和 Java 小程序（Java Applet）两种类型。

Java Application 一般是可以独立运行的计算机应用程序。Java Applet 是一种用 Java 语言开发，专门用于嵌入网页（HTML）文件中，并在浏览器中运行的程序。

目前，Java 语言还可以开发兼有 Java Application 和 Java Applet 特征的程序，该种程序既可以像 Java Application 一样独立运行，也可以嵌入到网页文件中，在浏览器中运行。

1.3.1 Java 应用程序的设计过程

由 Java 平台的特点可以知道，Java 应用程序需要经过“编辑—编译—运行”三个步骤才能完成正确的 Java 程序设计，如图 1-3 所示。



图 1-3 Java Application 的开发过程

1. 编辑源文件

程序设计的第一步就是编辑源文件，也称编辑源代码。本例以 Windows 操作系统中的记事本作为文本编辑器编辑源代码。

（1）在记事本中输入源代码。在 Windows 操作系统中，选择“开始”→“所有程序”→“附件”→“记事本”命令，打开记事本编辑器，在新文档中输入 Java 代码，源代码如下：

```
/**
 * 定义 HelloJava 类，功能是输出文本内容
 */
public class HelloJava {
    /**
     * 定义 HelloJava 类的方法，功能为输出 Welcome to Java World!
     * 方法名为 print，返回类型为 void，没有参数
     */
    public void print(){
```

//下面一行是输出内容的语句

```
System.out.println("Welcome to Java World!");
```

Java 语言程序设计时,除了字符串中的内容(即双引号中的内容,不包括双引号)外,所有源代码中所涉及的符号都必须在英文输入状态下进行输入。另外,Java 语言严格区分字母大小写,所以必须保持字母大小写的一致性,如 HelloWorld 与 helloWorld 是不同的。

(2)将源代码保存到名为 HelloJava.java 的文件中,该文件称为源文件。在记事本编辑器中,选择“文件”菜单中的“另存为”命令,在弹出的“另存为”对话框中,指定要保存文件的文件夹(路径),如 D:\Java(如果指定文件夹不存在,可以单击“新建文件夹”按钮,建立指定文件夹即可)。在“文件名”文本框中,输入 HelloJava.java(注意,HelloJava.java 与 HelloJava.Java 是不同的),其他选项保持不变,如图 1-4 所示。单击“保存”按钮,退出记事本编辑器,完成源代码的保存过程。

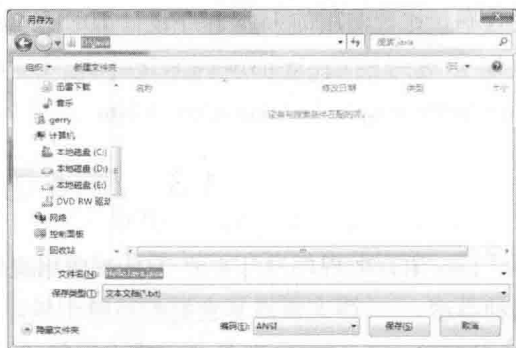


图 1-4 保存 HelloWorld.java 源文件示意图

(3)一个 Java 应用程序源文件中可以由若干个类(由 class 进行标识)组成。本例中只有一个 class HelloWorld,所以只有一个类。如果一个源文件中有多类(即有多个 class 标识的类),那么只能有一个类是 public 类(如 public class HelloWorld),而且源文件的文件名必须与这个类的名字完全相同,扩展名为.java。如果一个源文件中有多类(即有多个 class 标识的类),但是所有的类都不是 public 类,那么源文件的名字只要和某个类的名字相同,并且扩展名为.java 即可。

(4)编辑 Java 应用程序的主类源代码。如前所述,Java 应用程序是独立运行的计算机应用程序,所以必须存在一个程序执行入口。Java 语言规定,一个 Java 应用程序必须存在一个类含有 public static void main(String[] args){...}方法,这个类称为 Java 主类,public static void main(String[] args){...}方法是程序开始执行的位置,称为程序执行入口。HelloWorld 类是不能被 Java 语言解释运行的,要想实现其功能,必须将其一个对象放入主类的 main()方法中。按照上述方法,编辑和保存 Java 主类源文件 TestHelloWorld.java,源代码如下:

```
public class TestHelloJava {
    public static void main(String[] args){
        HelloJava helloJava=new HelloJava(); //表示生成 HelloJava 类的一个对象
        helloJava.print(); //实现对象的功能
    }
}
```

2. 编译源文件

从图 1-2 可以知道,当源文件编辑完成后,需要使用 Java 编译器(即 javac.exe 命令)对源文件进行编译。在编译时可以同时检验源代码的书写规则是否正确及是否导入程序所需要的 Java 类库等,如果编译失败,可以看到源代码的错误提示,如果编译成功,则在相应位置自动生成对应的字节码文件。字节码文件的文件主名与对应类的名字相同,扩展名为.class。

编译源文件的操作步骤如下:

(1) 在 Windows 7 及以后版本的 Windows 操作系统中,选择“开始”→“所有程序”→“附件”→“运行”命令,在弹出的“运行”对话框中输入 cmd 命令,单击“确定”按钮,如图 1-5 所示。

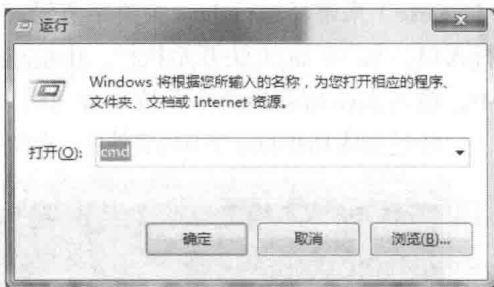


图 1-5 “运行”对话框

(2) 当出现命令提示符窗口时,显示的是 Windows 操作系统的默认路径,如图 1-6 (a) 所示。要编译源文件,应该将当前路径改变为文件所在的路径。例如,如果源文件路径在 C 盘上的 program 目录下,那么在命令提示符中输入以下命令并按 Enter 键: cd c:\program, 如图 1-6 (b) 所示。



(a) 命令提示符窗口



(b) 改变路径的命令提示符

图 1-6 命令提示符窗口

如果要改变为另一个驱动器上的路径(如源文件在 D 盘 java 目录下),应该在命令提示符中先输入以下命令并按 Enter 键: D:。

如前所述,HelloJava.java 和 TestHelloJava.java 文件都保存在 D 盘 java 目录下,因此需要将当前路径改变为 D 盘的 java 目录。如果输入 dir 命令,会列出目录下的所有文件和文件夹,如图 1-7 所示。

(3) 使用 Java 语言编译器(javac.exe)对源文件进行编译。在命令提示符窗口中,输入 Java 编译命令并按 Enter 键: javac HelloJava.java。



图 1-7 改变路径至 Java 源文件所在目录

如果编译成功,则在相应位置自动生成对应的字节码文件 HelloJava.class。字节码文件的文件主名与对应类的名字相同,扩展名为.class。如果编译失败,则可以看到源代码的错误提示,然后返回源文件修改错误,再进行编译,直到编译成功。同时将主类 TestHelloJava.java 文件编译生成相应的字节码文件 TestHelloJava.class,如图 1-8 所示。

3. 解释运行

当生成字节码文件后,就可以解释运行 Java 应用程序了。Java 应用程序使用 Java 解释器

(java.exe) 来解释运行 Java 主类字节码。Java 应用程序总是把 Java 主类的 main() 方法作为程序执行入口, 从 main() 方法开始执行。由前述可知, 本例的主类是 TestHelloJava, 在命令提示符窗口中, 输入 Java 解释器命令并按 Enter 键: java TestHelloJava。

经过加载 HelloJava 字节码文件后, 在命令提示符窗口中就可以看到程序的运行结果, 如图 1-9 所示。



图 1-8 Java 应用程序编译窗口

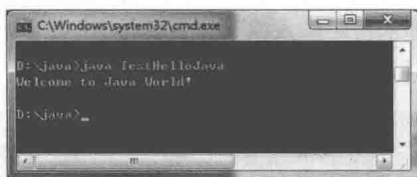


图 1-9 程序在屏幕上 (命令提示符窗口) 输出程序结果

1.3.2 Java 小程序的设计过程

与 Java 应用程序的设计过程相类似, Java 小程序也需要经过“编辑—编译—运行”三个步骤才能完成正确的 Java 程序开发, 如图 1-10 所示。

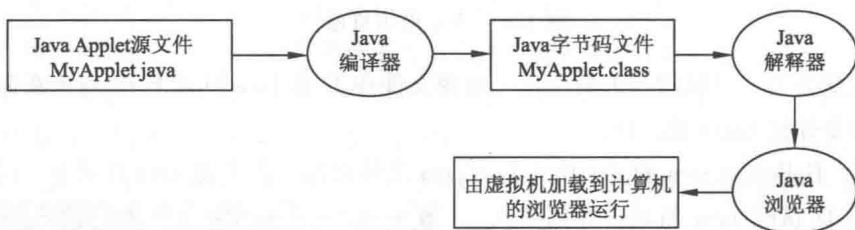


图 1-10 Java Applet 的开发过程

1. 编辑源文件

和 Java 应用程序的编辑源文件过程相同, 打开记事本编辑器, 在新文档中输入 Java 代码, 源代码如下:

```

import javax.swing.*;
import java.awt.*;
import java.applet.*;
public class HelloApplet extends JApplet {
    public void init() { //Applet 初始化方法
    }
    public void paint(Graphics g){ //Applet 显示内容方法
        g.setColor(Color.red);
        g.drawString("Welcome to Applet!",10,20);
    }
}
  
```

2. 保存源文件

将源代码保存到名为 HelloApplet.java 的文件中, 本例将该文件保存在 D 盘 java 目录下。

3. Java 小程序的类

Java 小程序可以由若干个类(由单词 class 进行标识)组成。与 Java 应用程序不同的是, 一个 Java 小程序不再需要包含了 main()方法的程序运行入口的主类, 而必须要包含一个并且只有一个类扩展了 Applet(或者 JApplet)类, 即它是 Applet(或者 JApplet)的子类, 同时它也是 public 类(即 public class HelloApplet extends JApplet)。

4. 编辑 Java 小程序的网页文件源代码

Java 小程序是嵌入在网页中运行的 Java 程序, 因此除了编辑 Java 小程序源文件外, 还需要编辑一个网页文件, 即 HTML 文件, 由它来加载 Applet 小程序。在记事本编辑器中, 按照网页文件的编辑方法, 输入 HTML 文件的内容, 源代码如下:

```
<!-- TestApplet.html 作者: gerry-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <!--下面的 Applet 标记就是用来加载 Java 小程序 HelloApplet 的源代码-->
    <applet code="HelloApplet.class" height="40" width="200">
    </applet>
  </body>
</html>
```

将源代码保存到名为 TestApplet.html 的文件中, 该文件必须与 HelloApplet.class 文件处于同一个路径下(本例保存在 D 盘 java 目录下), 否则在 Applet 标记中需要增加 codebase 属性以确定其所加载的 class 文件的路径。

5. 编译源文件

与 Java 应用程序相同, 当源文件编辑完成后, 需要使用 Java 编译器(即 javac.exe 命令)对源文件进行编译, 如图 1-11 所示。编译成功, 则在相应位置自动生成对应的字节码文件 HelloApplet.class。

6. 解释运行

当 Java 小程序编译成功后, 浏览 HTML 文件时就可以加载并看到其运行结果了。可以使用浏览器浏览 HTML 文件(出于安全性考虑, 某些浏览器可能会出现阻止 Applet 加载的提示, 用户只需设置为允许阻止的内容显示即可), 也可以使用 Java 小程序查看器(appletviewer.exe)查看 HTML 文件, 如图 1-12 所示。