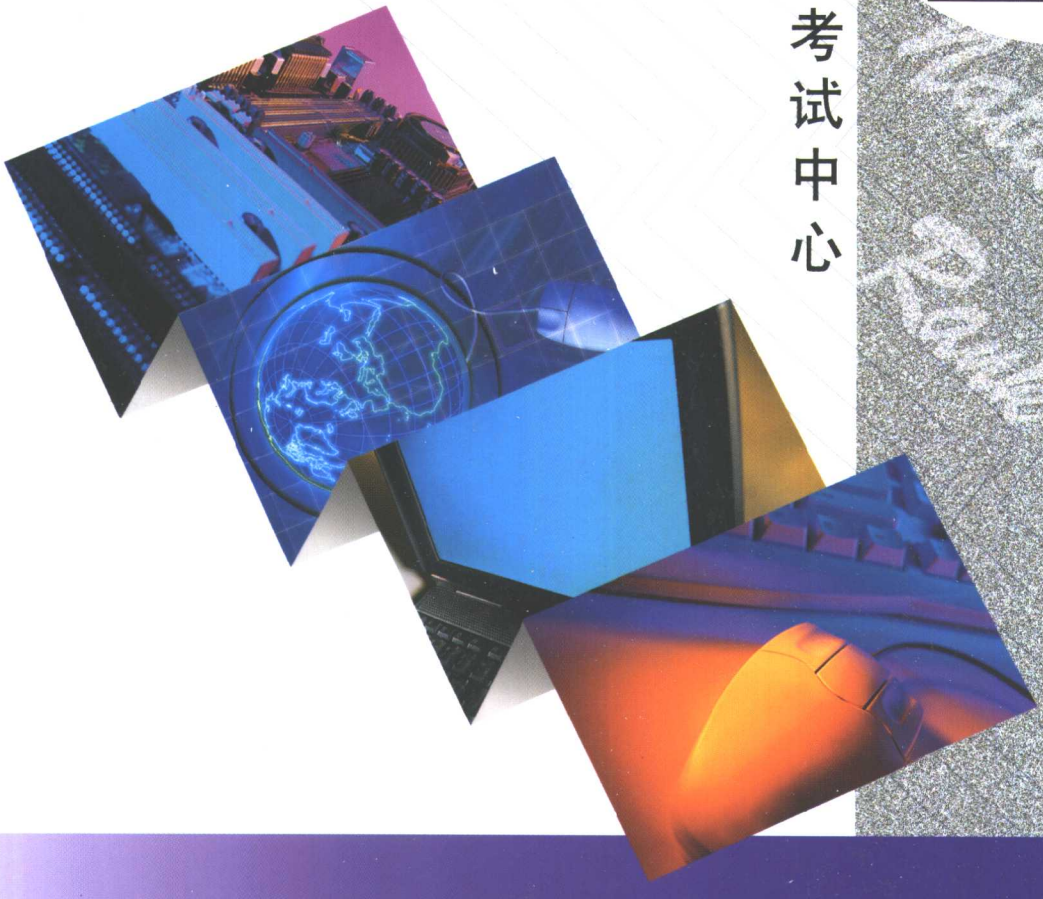
教育部考试中心

NCR Examination
全国计算机等级考试



全国计算机等级考试

二级教程

—Java语言程序设计

高等教育出版社

全国计算机等级考试

二级教程

——Java语言程序设计

教育部考试中心

高等教育出版社

内容提要

由国家教育部考试中心推出的计算机等级考试是一种客观、公正、科学的专门测试计算机应用人员的计算机知识与技能的全国性考试，它面向社会，服务于社会。

本书在教育部考试中心组织下、在全国计算机等级考试委员会指导下，由有关专家执笔编写而成。本书按照二级(Java语言程序设计)考试大纲的要求编写，内容包括：Java体系结构、基本数据类型、流程控制语句、类、数组和字符串操作、输入/输出及文件操作、图形用户界面编写、线程和串行化技术、Applet程序设计以及应用开发工具和安装使用等。本书是参加全国计算机等级考试二级Java语言程序设计的考生的良师益友，是教育部考试中心指定教材，也可作为欲学习Java编程的读者的参考书。

图书在版编目(CIP)数据

全国计算机等级考试二级教程——JAVA语言程序设计/教育部考试中心编. —北京：高等教育出版社，2004.4

ISBN 7-04-015260-6

Ⅰ.全… Ⅱ.教… Ⅲ.①电子计算机-水平考试-教材 ②JAVA语言-程序设计-水平考试-教材 Ⅳ.TP3

中国版本图书馆CIP数据核字(2004)第016651号

策划编辑 肖子东　**责任编辑** 古　锋　**封面设计** 于文燕
版式设计 马静如　**责任校对** 殷　然　**责任印制** 韩　刚

出版发行	高等教育出版社	购书热线	010-64054588
社　　址	北京市西城区德外大街4号	免费咨询	800-810-0598
邮政编码	100011	网　　址	http://www.hep.edu.cn
总　　机	010-82028899		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	高等教育出版社印刷厂		
开　　本	880×1230　1/16	版　　次	2004年4月第1版
印　　张	19.25	印　　次	2004年4月第1次印刷
字　　数	590 000	定　　价	32.40元

大力推行全国计算机等级考试 为发展知识经济、信息产业和培养计算机 专门人才作出贡献

（代　序）

中国科学院院士　北京大学信息与工程科学学部主任
全国计算机等级考试委员会主任委员
杨芙清

当今，人类正在步入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。科教是经济发展的基础，知识是人类创新的源泉。基础研究的科学发现、应用研究的原理探索和开发研究的技术发明，三者之间的联系愈来愈紧密，转换周期日趋缩短。世界各国的竞争已成为以经济为基础、以科技（特别是高科技）为先导的综合国力的竞争。

在高科技中，信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的观念、生活和社会的结构，是当代发展知识经济的支柱之一。

在信息科学技术中，微电子是基础，计算机硬件及通信设施是载体，计算机软件是核心。软件是人类知识的固化，是知识经济的基本表征，软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代，软件是信息化的核心，国民经济和国防建设、社会发展、人民生活都离不开软件，软件无处不在。软件产业是增长最快的朝阳产业，是具有高额附加值、高投入/高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程，促进从注重量的增长向注重质的提高方向发展，是典型的知识型产业。软件产业是关系到国家经济安全和文化安全，体现国家综合实力，决定21世纪国际竞争地位的战略性产业。

为了适应知识经济发展的需要，大力推动信息产业的发展，需要在全民中普及计算机的基本知识，广开渠道，培养和造就一批又一批能熟练运用计算机和软件技术的各行各业的专门人才。

1994年，原国家教委（现教育部）推出了全国计算机等级考试，它是一种重视应试人员对计算机和软件的实际掌握能力的考试。它不限制报考人员的学历背景，任何年龄段的人员都可以报考。这就为培养各行各业计算机的应用人才，开辟了一条广阔的道路。

1994年是推出计算机等级考试的第一年，当年参加考试的有1万余人；到2003年，报考人数已达251万余人。截止至2003年底，全国计算机等级考试共开考18次，考生人数累计超过1 050万人，其中，有350多万考生获得了不同级别的证书。

事实说明，鼓励社会各阶层的人士通过各种途径掌握计算机应用技术，并运用等级考试对他们的才干予以认真的、有权威性的认证，是一种较好的人才培养的有效途径，是比较符合我

国具体情况的。等级考试也为用人部门录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看，不论是管理人员还是应试人员，对该项考试的内容和形式都给予了充分肯定的评价。

计算机等级考试所取得的良好效果，也同全国各有关单位专家们在等级考试的大纲编写、试题设计、阅卷评分及效果分析等等多项工作中所付出的大量心血和辛勤的劳动密切相关，他们为这项工作的顺利开展作出了重要的贡献。

计算机与软件技术是一项日新月异的高新技术，计算机等级考试的考试内容和考核形式也将跟随新技术的发展不断创新，需要及时推出新的考试科目，及时修订旧科目的考试大纲、教材，对考试命题以及上机考试系统进行改革和完善，从而使等级考试更能反映当前的应用实际，使培养计算机应用人才的基础工作更健康地向前发展。本书的出版正是为了满足新时期新技术发展的需要，满足社会主义市场经济人才培养的需要。

从面临知识经济的机遇与挑战这样一个社会大环境的背景出发，考察全国计算机等级考试，就会看到，这一举措是符合知识经济和发展信息产业的方向的，是值得大力推行的。

我们相信，在21世纪知识经济和加快发展信息产业的形式下，在教育部考试中心的精心组织领导下，在全国各有关专家们的大力配合下，全国计算机等级考试一定会以更新的面貌出现，从而为我国培养计算机应用专门人才的宏大事业作出更多的贡献。

2003年12月

前　言

本书是根据教育部考试中心制定的《全国计算机等级考试考试大纲(2004年版)》中对Java语言的要求编写的。本书以最新JDK1.4.2版本为基础,介绍Java语言编程基础知识以及应用开发技术。本书中的实例均通过编译并可直接在JDK1.4.2上运行,每章包括练习题,供考生练习使用,是考生复习备考的必备教材。

本书由清华大学计算机系柳西玲教授主编,参加编写的作者有:第1、2、6、12章由柳西玲编写;第3、4、9章由许斌编写;第8、10、11章由朗波编写;第5、7章由金铁鹰编写。全书经南开大学辛运帏教授审阅。由于编写时间仓促,难免有疏误之处,请读者提出宝贵意见,以便修订时改进。

编者

2003年12月

目　录

第 1 章　Java 语言概论

Java 语言是 1995 年 5 月由 Sun 公司在 SunWorld 大会上发布的，从此，这一新一代的网络计算机语言受到广泛青睐，很快兴起了 Java 的热潮。仅仅短短的几年，它为 Web 提供了简便而功能强大的 API 接口，为 Web 提供了动态内容的交互页面技术，使 Web 网页翻开了新的一页，使互联网从以通信为主的功能扩展到网络应用系统服务。这种突破性的技术，对企业产生了革命性的影响，也显示了 Java 语言的强大生命力。

1.1　Java 语言简介

1.1.1　Java 语言的由来

1991 年，Sun 公司的 Jame Gosling、Bill Joe 等人，为电视、控制烤面包机等家用电器的交互操作开发了一个 Oak(一种橡树名字)软件，它是 Java 的前身。当时，Oak 并没有引起人们的注意，直到 1994 年，随着互联网和 3W 的飞速发展，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的支持，得以研制和发展。为了促销和法律的原因，1995 年 Oak 更名为 Java。Java 的得名还有段小插曲，一天，Java 小组成员正在喝 Java 咖啡时，议论给新语言取名字的问题，有人提议用 Java(Java 是印度尼西亚盛产咖啡的一个岛屿)，这个提议得到其他成员的赞同，于是就采用了 Java 来命名此新语言。很快 Java 被工业界认可，许多大公司如 IBM、Microsoft、DEC 等购买了 Java 的使用权，并被美国著名杂志 PC Magazine 评为 1995 年十大优秀科技产品。从此，开始了 Java 应用的新篇章。

Java 的诞生是对传统计算机模式的挑战，对计算机软件开发和软件产业都产生了深远影响：

(1) 软件 4A 目标要求软件能达到任何人(Any One)在任何地方(Any Where)在任何时间(Any Time)对任何电子设备(Any Device)都能应用。这样能满足软件在异构平台上互操作、具有可伸缩性和重用性并可即插即用等分布式计算模式的需求。

(2) 基于构件开发方法的崛起，引出了 CORBA 国际标准软件体系结构和多层应用体系框架。在此基础上形成了 Java 2 平台和.NET 平台两大派系，推动了整个 IT 业的发展。

(3) 对软件产业和工业企业产生了影响深远，软件从以开发为中心转向以服务为中心。中间件提供商、构件提供商、服务器软件提供商以及咨询服务商出现。企业必须重塑自我，B2B 的电子商务将带动整个新经济市场，使企业获得新的价值、新的增长、新的商机、新的管理。

(4) 对软件开发方法带来新的革命，重视使用第三方构件集成，利用平台的基础设施服务，实现开发各阶段的重用技术，重视开发团队的组织和文化理念，协作、创造、责任、诚信是人才的基本素质。

总之，目前已看到 Java 对信息时代的重要作用，未来还会不断发展，Java 在应用方面将会有更广阔的前景。

1.1.2　Java 语言的特点及优势

Java 是一个网络编程语言，它避免了许多其他编程语言的缺点，更好地利用了当前软件新技术，是一种新概念。首先，它作为编程语言，简单易学，利用了面向对象的技术基础，但又独立于硬件结构，具有可移植

性、健壮性、安全性、高性能。其次，它围绕网络应用开发，最大限度地利用网络资源。它的小应用程序(Applet)在网络上的传输不受计算机CPU和环境限制。第三，Java提供了丰富的类库，为编程人员提供快速和标准的应用接口，提高了应用软件的生产率。下面将详细地讨论Java语言的特点。

1. 简单易学

Java由Oak发展而来，简单易学。其简单性首先表现在自身系统的精炼，它的基本解释程序和类库支持只占40 KB空间，附加的基本标准类库和多线程支持也只占175 KB的空间，力图用最小的系统完成尽可能多的功能。对硬件环境的要求不高，在很普通的计算机上就能运行。其次，它只要求理解一些基本概念就能编程，无需任何编程基础就可学会。它的基本语法与C++类似，但删除了C++中许多不常用而又难以理解的内容，如运算符重载、多重继承、指针操作等，大大降低了编程学习的难度。

2. 利用面向对象技术

过去的高级语言大多数是面向过程的语言，这类语言通过数据结构和算法去描述客观世界。它的致命缺点是问题的解决与程序是一对一的关系，问题如有任何一点变更，程序也随之要修改。因此，可维护性和重用性很差。而面向对象的语言将客观世界看成由各种对象组成，对象客观实体可定义成大的结构——类(class)，每个类有自己的特性(属性)和操作(方法)。面向对象技术使复杂的问题可以分解简化，大大地提高了软件重用性，能适应变化快速的需求。与其他面向对象语言相比，Java利用面向对象技术更彻底。它把所有的Java应用和Applet都看做对象，按类封装。虽然Java的简单类型不是对象，但它提供了封装对象，使这些简单类型可以作为类来实现。其封装性实现了模块化和信息隐藏，继承性实现了代码重用，让用户可以自由地定义数据类型，建立自己的类库。

3. 分布式计算

Java提供的类库支持TCP/IP协议，应用程序可通过URL地址，在访问网络上任何地方的对象时，如同访问本地文件一样简单。

4. 健壮性(也称鲁棒性)

Java语言在编译和运行时都有比较严格的检查，过去许多编程语言的出错统计资料说明，最容易出错的是数据类型的不匹配和内存地址计算出错。Java在这方面有独特的措施。首先，Java是一种强类型语言，即它在编译和连接时进行大量的类型检查，防止不匹配数据类型的发生，对非法数据类型都将在编译和解释时指出。其次，Java语言不允许使用指针访问内存，更不允许使用指针数组访问内存。另外Java设计有自动收集垃圾功能，这不仅防止了内存地址计算出错的问题，也省去了编程时对内存进行分配的烦恼。

5. 安全性

面向网络、分布式环境的Java语言，对非法侵入的防范是至关重要的，Java语言必须提供充分的安全保障，它在运行程序时，有严格的访问权限检查。对字节代码执行前要检查，不允许使用指针，可防止对内存的非法入侵，它是目前安全性最佳的语言。

6. 跨平台(即体系结构中立)

Java解释器采用生成与体系结构无关的字节代码指令的技术，只需安装Java运行系统，就可保证Java程序可以在网络的任何地方运行。这对Web应用是很重要的，也是必需的，即同一个Java程序可在不同的处理器上运行，大大地降低了软件维护成本，提高了软件生产效率。

7. 可移植性

跨平台的特点，保证了软件的可移植性。Java的类库也具有可移植性。另外，Java本身的编译器也用Java语言编写，运行系统的虚拟机用C语言实现，这样，Java系统本身也具有可移植性。

8. 解释执行

Java语言用字节码进行解释执行。字节本身带有许多编译时产生的信息，使它的连接过程更简单。

9. 高性能

60年代，由于硬件条件限制，给人们留下解释器很慢的印象，另外，在Java刚问世时JVM(Java虚拟机)速度也不大理想，因此，至今在有些人的认知中也还存在误区，认为Java的缺点是性能慢。实际上，大量Ja-

va应用系统的成功案例，足以证明高性能已成为它的一大特点。字节码的设计很容易直接转换成一些特定CPU的机器码，其性能高到使人不感觉到与编译有什么差别。而且编译时还可对字节码优化，生成高质量的代码。另外，它的多线程技术也是性能提高的重要因素。

10．多线程

多线程是Java程序的并发机制，它能同步共享数据、处理不同的事件。网络连接需要时间，如在网上采用事件循环机制会造成网上瓶颈。因此，多线程技术使网上实时交互很容易实现，为解决网上大数量的客户访问提供技术基础。

11．动态性

Web应用系统中最大的挑战是快速变化的需求，即软件的演化性。软件可扩充性、可伸缩性强，才能适应不断发展变化的环境。Java语言带动了软件构件化方法，它本身的类库就可以自由地增加新方法或实例。Java通过接口支持多重继承，使类继承具有更灵活的扩展性。它还可以随时插入构件和数据库。

12．Applet的特点

Applet是Java的一类特殊应用程序，它嵌入HTML中，随主页发布到互联网上。利用它可以实现多媒体的用户界面或复杂的计算。Applet要求在支持Java的浏览器上运行，它使互联网上的信息能很容易地实现动态性和交互性。Java类库提供的Applet类是所有Applet程序的根，Applet类为编程准备好程序框架，编写时只需填入相应的方法实现，无需考虑窗口创建、事件处理等问题，大大简化了编程的复杂性。

1.1.3 Java语言实现的机制

1．Java虚拟机

Java语言的执行模式是半编译和半解释型。Java编写好的程序首先由编译器转换为标准字节代码，然后由Java虚拟机去解释执行。字节代码也是一种二进制文件，但不能直接在操作系统上运行，它可看做虚拟机的机器码。虚拟机把字节代码程序与各操作系统和硬件分开，使Java程序独立于平台。虚拟机可以用软件实现，也可用硬件实现，但在无线技术中，都用硬件实现。

Java程序的下载和执行步骤如图1－1所示：

(1) 程序经编译器得到字节代码；

(2) 浏览器与服务器连接，要求下载字节码文件；

(3) 服务器将字节代码文件传给客户机；

(4) 客户机上的解释器执行字节代码文件；

(5) 在浏览器上显示并交互。

虚拟机(VM)的执行过程有3个特点：

(1) 多线程；

(2) 动态连接；

(3) 异常处理。

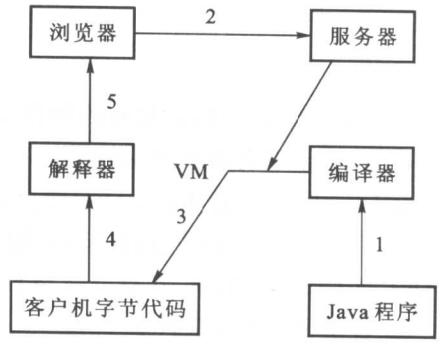


图1－1 Java程序的下载和执行

2．垃圾回收机制

在Java语言中，所有事物都封装在类中，需要时创建类的实例(对象)来处理，这种动态的实例都存储在内存堆中。为了充分利用资源，Java有一个系统级的线程，对内存的使用进行跟踪，使程序员从繁忙的内存管理中解放出来。该线程可以在系统空闲时对不用的内存进行回收。

3．代码安全检测

Java程序的安全性体现在多个层次上，在编译层，有语法检查；在解释层，有字节码校验器、测试代码段格式和规则检查，访问权限和类型转换合法性检查，操作数堆栈的上溢或下溢，代码参数类型合法性等；在平台层上，通过配置策略，可设定访问资源域，而无需区分本地或远程。

1.2 Java 的体系结构

前面已说到 Java 以 JVM 为基础，实际上，许多 JVM 也是由软件实现的。Java 的体系结构如图 1－2 所示。最下层是移植接口，由适配器和 Java OS 组成，保证 Java 体系结构可以跨平台。虚拟机的上层是 Java 基本类和基本 API，它们都具有可扩展性。最上一层是 Java 应用程序和 Applet 小程序。

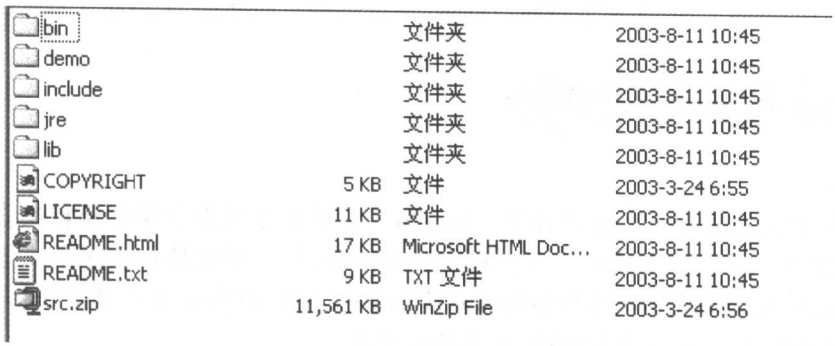
Java 的产品主流操作系统平台是 Solaris、Windows 和 Macintosh。目前最新的 JDK 是 j2sdk1.4.2 版，可到 SUN 公司网站下载。本书第 11 章将叙述如何具体下载，全书所有例子都用 j2sdk1.4.2 版编写。

<table>
<tr><td colspan="2">Java 应用程序和 Applet</td></tr>
<tr><td>Java 基本 API</td><td>Java 标准扩展 API</td></tr>
<tr><td>Java 基本类</td><td>Java 标准扩展类</td></tr>
<tr><td colspan="2">Java 虚拟机</td></tr>
<tr><td colspan="2">移植接口</td></tr>
</table>

图 1－2 Java 的体系结构

1.2.1 JDK 目录结构

以 j2sdk1.4.2 版为例，解压后得目录结构如图 1－3 所示。



bin		文件夹	2003-8-11 10:45
demo		文件夹	2003-8-11 10:45
include		文件夹	2003-8-11 10:45
jre		文件夹	2003-8-11 10:45
lib		文件夹	2003-8-11 10:45
COPYRIGHT	5 KB	文件	2003-3-24 6:55
LICENSE	11 KB	文件	2003-8-11 10:45
README.html	17 KB	Microsoft HTML Doc...	2003-8-11 10:45
README.txt	9 KB	TXT 文件	2003-8-11 10:45
src.zip	11,561 KB	WinZip File	2003-3-24 6:56

图 1－3 JDK 目录结构

其中：

bin 目录下有编译器、解释器和许多工具(如服务器工具、IDL、package 工具和 jdb 等)。

demo 目录下有各种演示例子。

include 目录下是 Win32 子目录，都是本地方法文件。

jre 目录是 Java 程序运行用环境的根目录，它下面有 bin 子目录，包括平台所用工具和库的可执行文件和 DLL 文件；lib 子目录，包括 java 运行环境的代码库、属性设置和资源文件，默认安装目录，安全管理。

lib 目录下都是库文件。

src.zip 是源码压缩文件。

1.2.2 Java 的 API 结构

Java 是面向对象语言，对象是客观事物的实体，对象与实体是一一对应的，它是很具体的概念。在面向对象语言中，对象是程序的基本单位。Java 语言以类为程序的基本单位，类(Class)是具有某些共同特性实体的集合，是一种抽象的概念，它实质上是一种对象类型的定义，即对具有相同行为对象的一种抽象，说明该类型对象的性质。在 Java 语言中，类是具有某种功能的基本模块的描述，它所提供的标准类库，为编程所需的底层模块提供了常用的方法和接口，并将它们分类封装成包，每个包又包括子包，形成树结构的类层次。类库主要包括核心 java 包、javax 和 org 扩展包。

1. Java 核心包

Java 核心包包括：

(1) java.lang 包：封装所有编程应用的基本类，如 Object、Class、String、System、Integer、Thread 等。而 Object 是所有类的根，它所包含的属性和方法被所有类继承，它的方法有：

- protected Object clone() throws CloneNotSupportedException，该方法生成当前对象的一个拷贝，并将它返回。该对象类型是 Object，它要求使用该方法的类都必须实现 cloneable 接口，否则视为异常。
- public final Class getClass()，该方法返回当前对象的运行对象，该对象类型是 Class 类。
- public final void notify()，该方法唤醒等待对象监视器的多个线程中的一个，是用于多线程技术的方法。
- public final void notifyAll()，该方法唤醒所有等待监视器的线程。
- public final void wait(long timeout)throws InterruptedException，该方法让当前线程放弃对象的锁定。形参 timeout 为超时，如当前线程在等待中被中断是异常。
- public final void wait()throws InterruptedException，该方法与 wait(0)作用相同。
- public int hashCode()，该方法返回一个 hash code 值。
- public boolean equals(Object obj)，如当前对象与形参对象相同则返回 true，否则返回 false，所有类型对象的比较都用此方法。
- public String toString()，返回该对象信息的字符串，该字符串与对象相同类型。
- protected void finalize()throws Throwable，该方法把对象从内存中清除，由垃圾收集器自动调用。
- public final void wait(long timeout,int nanos)throws InterruptedException，该方法的形参 nanos 是十亿分之一秒的时间，超时计算更精确。

Class 类是由编译器自动生成对象的一个特殊类，它伴随每个类。这个 Class 对象包含所属类的所有信息，可通过 Class 类的方法访问这些信息。Class 的方法有：

- public static Class forName(String className)throws ClassNotFoundException，该方法返回形参所指类的 Class 对象，形参是一个类名。
- public String getName()，该方法返回 Class 对象的名字。
- public Class getSuperclass()，返回一个数组，其成员是 Class 类的实例。
- public ClassLoader getClassLoader()，对类加载内存。
- public Class getComponentType()，返回数组成员的类型。如对象不是数组，返回 null(空值)。
- public int getModifiers()，返回类/接口的修饰符，如 public，final 等。
- public Class class getDeclaringClass()，如当前对象是另一个类的成员，则返回该类的 Class 对象，否则为空。
- public Class[ ]getSuperclass()，返回的是一个数组，数组成员是 Class 对象。
- public Package getPackage()，返回当前类所在包。
- public Class[ ] getInterfaces()，返回接口。
- public Object[ ] getSigners()，返回信号。
- public Field[ ] getFields()throws SecurityException，正常返回项，出错抛出异常。
- public Method[ ] getMethods()throws SecurityException，正常返回方法结果，出错抛出异常。
- public Constructor[ ] getConstructors()throws SecurityException，正常返回结构文件名，出错抛出异常。

System 类是一个特殊类，它是一个 final 类，所有的方法都用类变量来调用，即对 System 类不可能实例化，它主要提供了标准输入/输出和系统环境信息的访问、设置。它的属性有：

- public static final InputStream in，标准输入。
- public static final PrintStream out，标准输出。
- public static final PrintStream err，标准错误输出。

有关环境信息的方法有：

- public static Properties getProperties(argument)，返回系统环境信息。
- public static String setProperty(String keg,String value)，设置系统变量的值。
- public static long currentTimeMillis()，返回系统时间(毫秒)。
- public static void exit(int status)，强制关闭 JVM 并把状态 status 传给操作系统，status 非零时，是非正常退出。
- public static void gc()，运行垃圾收集器。
- public static void setIn(InputStream in)，设置输入信息。
- public static void setOut(PrintStream out)，设置输出信息。
- public static void setErr(PrintStream err)，设置错误信息。

(2) java.awt 包：封装抽象窗口工具包，提供构建和管理用户图形界面功能。

(3) java.applet 包：为 Applet 提供执行需要的所有类，主要是访问 Applet 内容的通信类。

(4) java.io 包：提供程序输入/输出文件操作的类。

(5) java.net 包：提供程序执行网络通信应用及 URL 处理的类。

(6) java.rmi 包：提供程序远程方法调用所需的类。

(7) java.math 包：提供程序常用的整数算术以及十进制算术的基本方法类。

(8) java.util 包：提供实用程序类和集合类，如系统特性定义和使用、日期函数类、集合 Collection、Map、List、Arrays 等常用工具类。

(9) java.sql 包：提供访问和处理标准数据源数据的类。

(10) java.security 包：提供网络安全操作类。

(11) java.text 包：提供所有处理文本、日期、数字以及非自然语言的消息操作的类。

(12) java.bean 包：提供开发编写 java bean 所需的类。

2. 扩展包 Javax

扩展包 Javax 有：

(1) javax.naming 包：提供命名服务所需的类和接口。

(2) javax.swing 包：提供构建和管理应用程序的图形界面的轻量级的构件。

(3) javax.rmi 包：提供远程方法调用的应用程序接口。

(4) javax.transaction 包：提供事务处理所需的基本类。

(5) javax.sound 包：提供多媒体声音要求的 MIDI 输入/输出以及合成操作的基本类。

(6) javax.accessibility 包：提供用户界面构件之间相互访问机制的基本类。

3. Org 包

Org 包是一些有关国际组织的标准。

4. Java 语言的 API 文档

Java 语言的 API 文档是编程的好工具，当需要查询类库时，可到网站 http://java.sun.com/j2se/1.4.2/docs/下载 j2sdk-1_4_2-doc 后，找 docs 文件夹中的 api 文件夹，打开 index.html，进入选择包的界面，选任意包后，再从包中选类的超链接，就可看到该类的组成：

(1) 包名。

(2) 类名。

(3) 继承结构。

(4) 类的实现接口。

(5) 类定义和说明。

(6) 类的成员变量列表。

(7) 构造方法列表。

(8) 方法返回类型及方法列表。

(9) 方法详细列表及描述。

为了查找方便,可把 JDK1.4.2 \ docs \ api \ index.html 存入浏览器中,可随时查阅。

1.2.3 Java程序结构

Java 程序包括源代码(.java 文件)、由编译器生成的类(.class 文件)、由归档工具 jar 生成的.jar 文件、对象状态序列化.ser 文件。由于只有源代码需要开发者编写,这里也就只讨论源代码的结构:

(1) package 语句,0 或 1 个,指定源文件存入所指定的包中,该语句必须在文件之首,如没有此语句,源文件存入当前目录下。

(2) import 语句,0 或多个,必须在所有类定义之前引入标准类。

(3) public classDefinition,0 或 1 个,指定应用程序类名,也是源文件名。

(4) classDefinition,0 或多个,类定义。

(5) interfaceDefinition,0 或多个,接口定义。

提示:Java 是区分大小写的。源文件名与程序类名必须相同,其扩展名为.java,源文件中最多只能有一个 public 类,其他类的个数不限。

1.2.4 Java程序编写及运行的过程

Java 有 2 类应用程序,Java Application 和 Java Applet。前者是独立的应用程序,而后者嵌入 HTML 在浏览器中执行。下面针对这两类应用程序的编程和执行环境进行讨论。

1. 编写和运行 Java Application 程序

Java Application 应用程序的编写和执行分 3 步进行:

(1) 编写源代码,首先要选一个无格式的文本编辑器,如 Windows 的记事本、UltraEdit 等。千万不要用 Word 这类带格式的文本编辑器,因为它隐藏有许多 Java 解释器不能识别的格式信息。其次,创建一个文件夹,如 D: \ javaProgram 用来存放编写好的 Java 程序。然后可打开编辑器编写程序,写完后以扩展名.java 存入新建文件夹 D: \ javaProgram 中。

(2) 编译源代码,只要下载了 j2sdk1.4.2,它已包含有编译器 javac.exe 对 Java 程序进行编译,需要进入 MD-DOS 方式,在 DOS 提示符下输入命令:cd D: \ javaProgram, 设置运行目录,再输入编译命令: javac 源文件全名(带扩展名.java),如没有语法错误,在文件夹 D: \ javaProgram 中出现一个二进制字节码文件:源文件名.class,它由编译器自动生成。如源代码有语法错误,给出错误报告,按行指出错误,编者按报告改正错误后,重复上面编译命令,直至编译成功。

(3) 解释执行,利用 j2sdk1.4.2 的解释器 java.exe 执行。仍在 DOS 方式下,输入命令:java 源文件名(不带.java 扩展名),如执行成功,显示结果,如执行有错,显示错误报告,设法排错直至获得正确结果。

2. 编写和运行 Java Applet 应用程序

Java Applet 应用程序的编写和执行共分 4 步进行:

(1) 编写源代码,这步与 Java Application 应用程序相同,编辑一个源文件存入指定文件夹中。注意,该程序不含 main 方法。

(2) 编写 HTML 文件调用该小程序,以.html 为扩展名存入相同文件夹。

(3) 编译过程,与 java Application 应用程序相同,编译应用程序的 java 部分。

(4) 解释执行,同样在 DOS 方式下,输入命令: appletviewer *filename*.html (这里的 *filename* 不要求与 java 文件同名)。如无错误,显示结果,如有出错报告,排错后,重复上面解释执行。

3. JDK 工具

上面编写执行 java 程序的过程中用到了一些工具，SUN 公司免费提供了一套 JDK 工具，它主要包括：

(1) javac.exe：Java 编译器，能将源代码编译成字节码，以.class 扩展名存入 java 工作目录中，它的命令格式为：javac [选项] 文件名(全名)。

其中[选项] - g　输出文件中加行号及局部变量信息，为调试程序时用。

- o　对类文件进行优化编译。
- nowarn　关闭警告信息，只显示错误信息。
- verbose　显示源代码文件和字节码文件的相关信息。
- 路径/目录。

(2) java.exe：Java 解释器，执行字节码程序。该程序是类名所指的类，必须是一个完整定义的名字，必须包括该类所在包的包名，而类名和包名之间的分隔符是“.”，如类不在任何包中，类名是单独的。执行命令格式为：java [选项] 类名 [程序参数]。

其中[选项] - cs　检查目标文件是否过时，如已过时将自动从源文件重新编译。

- D 属性名=值　定义属性名。
- debug　将程序连接到调试器。
- ms 分配内存初值　解释器启动时分配给堆的内存大小。
- mx 分配最大内存值　解释器为对象和数组，动态分配堆最大内存值，默认值为 16MB。
- noverify　不进行字节代码验证。
- verify　进行字节代码验证，也是默认状态。
- noasyncgc　关闭异步垃圾收集器。
- oss 栈尺寸　设置每个线程栈的尺寸，默认值为 400KB。
- ss 栈尺寸　设置每个线程本地栈的尺寸，默认值为 128KB。
- v　每装载一个类，打印一条信息。
- verbosegc　无用单元收集器每释放一次内存，打印一条信息。
- 路径/目录。

(3) javadoc.exe：Java 文档生成器，对 Java 源文件和包以 MML 格式产生 AP 文档。如给出包名，它按类路径相关的对应包目录下找出所有.java 源文件，为每个类生成一个 HTML 文档并生成该包中所有类的 HTML 文档索引。默认时，这些 HTML 文件存入当前目录下。执行命令的格式：javadoc[选项]包名或 javadoc[选项]文件名。

其中[选项] - verbose　显示行为信息。

- 路径/目录　指明包的路径/HTML 文件所存目录。

(4) javap.exe：Java 类分解器，对.class 文件提供字节代码的反汇编，并打印。默认时，打印类的公共域、方法、构造函数、静态初值。执行命令格式：　javap[选项]类名。

其中[选项]　-c　打印给出每个类中方法的 JVM 指令。

- -classpath　指明类路径。
- -l　在原来打印信息上增加私有及保护方法和变量。

(5) jdb.exe：Java 调试器，如编译器返回程序代码错误，可用它对程序进行调试，它是解释器的拷贝，类调试器。执行命令格式：jdb[解释器选项]类名或 jdb[- host 主机名] - password。

它包含的调试命令有：

!!　重复上一次输入的命令。

catch [异常类]　在激活异常时，自动产生一个断点。

clear[类名:行号]　清除所指类的行断点。

cont　继续指向行。