

## 内 容 提 要

本书是全国信息化计算机应用技术资格认证（CCAT）项目的指定教材，属于操作员级认证体系。CCAT 资格认证项目设立的目的除了培养学生掌握相应专业的理论知识，注重学员动手能力、创新能力的训练外，还注重培养和提高学员的企业管理能力，为社会和企业培养既懂技术、又懂管理的复合型人才，以改变人才培养中存在的重理论轻实践、重文凭轻能力的缺陷。

本书以运用 JBuilder X 开发 Java 应用程序为主线，详细介绍了 JBuilderX 的各项功能及使用技巧。

本书共分为三部分，分别介绍 Java 基础知识、JBuilder X 的使用方法、Java 的数据库和网络编程等，其中第 1 章到第 6 章主要介绍 Java 基础知识，第 7 章和第 9 章主要介绍 JBuilder X 的使用方法，第 8 章介绍图形界面编程，第 10 章和第 11 章主要介绍数据库和网络编程。

本书也可作为高等院校、高等职业院校信息与计算机相关专业的教材，也适合作为 Java 学习者的自学参考书。

版权专有 侵权必究

---

### 图书在版编目（CIP）数据

Java 程序设计标准教程/胡剑锋，盛鸿宇主编；全国信息化计算机应用技术资格认证管理中心组编. —北京：北京理工大学出版社，2007. 1

全国信息化计算机应用技术资格认证指定教材

ISBN 978 - 7 - 5640 - 0820 - 8

I. J… II. ①胡… ②盛… ③全… III. Java 语言 - 程序设计 - 资格考核 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字（2006）第 147733 号

---

出版发行/北京理工大学出版社

社 址/北京市海淀区中关村南大街 5 号

邮 编/100081

电 话/(010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址/http://www.bitpress.com.cn

经 销/全国各地新华书店

印 刷/北京国马印刷厂

开 本/787 毫米×1092 毫米 1/16

印 张/15.25

字 数/321 千字

版 次/2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷

印 数/1~4000 册

定 价/28.00 元

责任校对/张 宏

责任印制/李绍英

# 目 录

<b>第 1 章 概述</b> .....	1
1.1 Java 的诞生与发展规律.....	1
1.2 Java 语言的特点.....	5
1.3 与 C 和 C++语言的异同.....	7
1.4 Java 平台 .....	9
1.5 Java 程序种类.....	10
1.6 Java 应用程序开发.....	11
习题.....	15
<b>第 2 章 数据类型、运算符与表达式</b> .....	16
2.1 变量和数据类型.....	16
2.2 运算符和表达式.....	23
习题.....	27
<b>第 3 章 Java 数组、字符串和流程控制</b> .....	29
3.1 Java 的数组 .....	29
3.2 字符串.....	32
3.3 Java 的程序结构.....	35
习题.....	43
<b>第 4 章 面向对象的基本概念</b> .....	46
4.1 类和对象.....	46
4.2 类的定义和实现.....	49
4.3 类的继承和派生.....	60
4.4 包.....	62
4.5 方法.....	67
习题.....	68
<b>第 5 章 异常处理和多线程</b> .....	70
5.1 异常与异常类.....	70
5.2 异常的处理.....	72
5.3 多线程.....	77
习题.....	84
<b>第 6 章 接口和输入输出</b> .....	85
6.1 接口.....	85

6.2 输入与输出.....	88
习题.....	92
<b>第 7 章 JBuilder X 开发环境</b> .....	93
7.1 JBuilder 的历史.....	93
7.2 JBuilder 的特性.....	93
7.3 JBuilder X 10.0 的安装.....	98
7.4 JBuilder X 的主界面.....	102
7.5 主菜单.....	105
7.6 工具栏.....	121
7.7 创建应用程序举例.....	123
习题.....	130
<b>第 8 章 Java 图形用户界面编程</b> .....	131
8.1 Java 图形用户界面编程知识.....	131
8.2 图形用户界面组件编程实例.....	131
习题.....	166
<b>第 9 章 布局管理器</b> .....	167
9.1 布局管理器.....	167
9.2 BorderLayout 布局管理器.....	167
9.3 FlowLayout 布局管理器.....	170
9.4 XYLayout 布局管理器.....	172
9.5 GridBagLayout 布局管理器.....	174
9.6 PanelLayout 布局管理器.....	177
9.7 CardLayout 布局管理器.....	179
9.8 其他布局管理器.....	180
习题.....	180
<b>第 10 章 网络编程</b> .....	181
10.1 URL 类的应用.....	181
10.2 Socket 编程.....	188
习题.....	201
<b>第 11 章 JBuilder 数据库编程</b> .....	202
11.1 数据库概述.....	202
11.2 JDBC 简介.....	208
11.3 JDBC 数据库应用程序.....	210
习题.....	224
<b>参考文献</b> .....	225

# 前 言

为贯彻中共中央、国务院《关于进一步加强人才工作的决定》，培养高层次、高技能和复合型的社会急需人才，全国信息化计算机应用技术资格认证管理中心受人事部中国高级公务员培训中心和教育部全国高等学校计算机教育研究会的委托，组织编写了全国信息化计算机应用技术资格认证（简称“CCAT 资格认证”）项目的指定教材。CCAT 资格认证项目是全国性的 IT 培训认证项目，其主要特色是为社会培养动手能力和管理能力兼备的人才。该培训认证与在国际上享有盛誉的瑞士管理论坛（Swiss Management Forum，简称“SMF”）已实现了国际互认。本书属于 CCAT 资格认证项目中操作员级认证体系。

Java 语言是 Sun 公司推出的一种面向对象的、多线程的、交互式的编程语言，它功能强大，表达能力强，应用广泛，是当前最为流行的编程语言。Java 语言的平台无关特性非常适用于网络和分布式应用，因此一经推出就备受青睐；其卓越的设计思想也使 Java 成为国际互联网中的“世界语”，并将网络的发展带入了一个新纪元。

本书共 11 章，循序渐进地介绍了 Java 语言的各种知识，包括：Java 的基本语法、面向对象基础、异常处理和多线程、接口和输入/输出、图形用户界面、网络编程、数据库编程、JBuilder X 基础、JBuilder X 布局管理器。在本书中，配有例题和练习，以便于读者能更好地学习和掌握 Java 的技能和操作。

本书具有以下特色：

1. 理论与实践结合。结合书中的例题对相关的理论知识进行系统地介绍。
2. 介绍 JBuilder X 这个 Java 集成开发环境。
3. 本书讲解力求简练、准确，强调知识的层次性和连贯性，例题和练习丰富实用，注重学生能力的培养。
4. 本书内容翔实、通俗易懂、图文并茂，以实例和图形界面为主，介绍了 Java 程序设计的各种方法和技巧。
5. 本书中的大量实例都经过作者在 JDK 环境和 JBuilder X 中测试通过。

本书是 CCAT 资格认证指定教材，适用于社会各界人士以及在校学生参加“全国信息化计算机应用技术资格认证”考试的需求，尤其适用于高等院校、大中专学校等进行课程置换，作为相关课程的教材，亦可作为计算机职业技能考试及继续教育的培训教材或自学教材。

本书由胡剑锋博士、姚华副教授主编，何胜文参与编写。

由于时间仓促，加之编者水平有限，书中难免有疏漏之处，恳请广大读者批评指正。

编 者  
2006 年 12 月

# 第 1 章 概 述

最近几年，随着全球 Internet 的迅猛发展及万维网（World Wide Web，WWW）的日益普及和快速增长，整个计算机环境正在经历着深刻的变革。1989 年，HTML（Hypertext Markup Language，超文本标记语言）和 WWW 的产生是 Internet 数据描述语言的一次飞跃。Java 语言产生后，由于它独具特点，逐渐成为在 Internet 网络以及操作系统等其他方面的最受欢迎的开发与编程语言。

## 1.1 Java 的诞生与发展规律

美国硅谷有一句行话，每 10~15 年有一次轮回。最近的一次轮回就是从 Java 开始。Java 是一个由 Sun 公司开发而成的新一代编程语言。使用它可在各式各样不同种机器、不同操作平台的网络环境中开发软件。不论使用的是哪一种 WWW 浏览器、哪一种计算机、哪一种操作系统，只要 WWW 浏览器上面注明了“支持 Java”，就可以看到生动的主页。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式，带来了自 PC 机以来的又一次技术革命，为迅速发展信息世界增添了新的活力。

### 1.1.1 Java 的诞生

美国 Sun 公司的 Java 语言开发小组成立于 1991 年，其目的是开拓消费类电子产品市场，例如，交互式电视、烤面包箱等。Sun 内部人员把这个项目称为 Green，那时 World Wide Web（万维网）还在图纸上。该小组的领导者是 James Gosling，是一位非常杰出的程序员。他出生于 1957 年，于 1984 年加盟 Sun 公司，之前在一家 IBM 研究机构工作。在研究开发过程中，Gosling 深刻体会到消费类电子产品和 workstation 产品在开发哲学上的差异：消费类电子产品要求可靠性高、费用低、标准化、使用简单，用户并不关心 CPU 的型号，也不欣赏专用昂贵的 RISC 处理器，他们需要建立在一个标准基础之上，具有一系列可选的方案，从 8086 到 80586 都可以选取。为了使整个系统与平台无关，Gosling 首先从改写 C 编译器着手。但是 Gosling 在改写过程中感到仅 C 是无法满足需要的，于是在 1991 年 6 月份开始准备开发一个新的语言，那么给它起一个什么名字呢？Gosling 回首向窗外望去，看见一棵老橡树，于是取名 Oak（橡树），这就是 Java 语言的前身。后来发现 Oak 已是 Sun 公司另一个语言的注册商标，才改名为 Java，即太平洋上一个岛屿的名字，该岛出产一种味道非常美妙的咖啡。Gosling 在开始写 Java 时，并不局限于扩充语言机制本身，更侧重于语言所运行的软硬件环境。他要建立一个系统，这个系统运行于一个巨大的、分布的、异构的网格环境中，完成各电子设备之间的通信与协同工作。Gosling 在设计中采用了虚拟机码（Virtual Machine Code）方式，即 Java 语言编译后产生的是虚拟机，虚拟机运行在一个解释器上，每一个操作系统均有一个解释器。

这样一来，Java 就成了平台无关语言。这和 Gosling 设计的 SunNeWs 窗口系统有着相同的技术味道。在 NeWs 中用户界面统一用 Postscript 描述，不同的显示器有不同的 Postscript 解释器，这样便保证了用户界面的良好的可移植性。

Patrick Naughton 也是 Sun 公司的技术骨干，曾经是 OpenWindows 项目的负责人。当 Naughton 加入该小组后，整个工作进展神速。经过 17 个月的奋战，整个系统胜利完成。它是由一个操作系统、一种语言（Java）、一个用户界面、一个新的硬件平台、三块专用芯片构成的。通常情况下，这样的项目在 Sun 公司要 75 个人干 3 年。项目完成后，在 Sun 公司内部做了一次展示和鉴定，观众的反应是：在各方面都采用了崭新的、非常大胆的技术。许多参观者对 Java 留下了非常深刻的印象，特别得到 Sun 的两位领导人 Scott McNealy 和 Bill Joy 的关注，但 Java 的前途未卜。到了 1994 年，WWW 已如火如荼地发展起来。Gosling 意识到 WWW 需要一个中性的浏览器，它不依赖于任何硬件平台和软件平台，它应是一种实时性较高、可靠安全、有交互功能的浏览器。于是 Gosling 决定用 Java 开发一个新的 Web 浏览器。这项工作由 Naughton 和 Jonathan Payne 负责，到 1994 年秋天，完成了 Web Runner 的开发工作。Web Runner 是 HotJava 的前身，这个原型系统展示了 Java 可能带来的广阔市场前景。Web Runner 改名为 HotJava，并于 1995 年 5 月 23 日发表后在产业界引起了巨大的轰动，Java 的地位也随之而得到肯定。又经过一年的试用和改进，Java 1.0 版终于在 1996 年年初正式发表。

Java 语言（简称 Java）的诞生对 IT 产业带来了一次变革，从某些意义上讲对人们的日常生活也产生了深远的影响。Java 作为一种 C/C++ 语言，但其平台无关性和与互联网发展紧密结合，预计未来必定成为互联网和计算机应用的主流。Java 当之无愧地被纽约时报评为 1995 年的十大科技成果之一（计算机类仅此一项入选），Java 将作为一项重大发明载入科技史册！微软总裁比尔盖茨曾在观察了一段时间后，十分惭愧地说：“Java 是长时间以来最卓越的程序设计语言”，并确定微软整个软件开发的战略从 PC 单机时代向着以网络为中心的计算机时代转移，而购买 Java 则是他的重大战略决策的实施部署。Sun 公司的总裁 Scott McNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。环球信息网 WWW 的创始人 Berners-Lee 说：计算机事业发展的下一个浪潮就是 Java，并且将很快发生。所以，使用 Java 已成大势所趋！Microsoft 和 IBM 两大公司都在 Internet 上销售用 Java 编写的软件，IBM 著名 Java 开发集成环境 Visual Age For Java、网站集成平台 Webspere，Microsoft 的 VJ++ 都是目前主要常见的 Java 开发环境或产品。另外，Apple, HP, IBM, Microsoft, Novell, SGI, SCO, Tandem 等公司均在各自开发的操作系统中加入了 Java 开发运行环境，而负责开发并推广 Java 技术的 Javasoft 公司（这是 Sun 公司下属的一个子公司），通过颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统，这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程，不同的用户也就可以脱离 Web 浏览器来运行 Java 应用程序，这无疑是很受广大用户的欢迎的，也为 Java 语言的应用开拓了极为广阔的前景。每年一度的 JavaOne 大会几乎都会推出让 Java 更受人欢迎的新技术，曾经有人预言：Java 将是网络上的“世界语”，将会被未来所证实。1998 年，由于 Java 在安全、性能等方面到了一个关键阶段，甚至 Java 被列入了 20 世纪末十大必然死亡的技术之一，十大科技成果被预言必将死亡实在是幽了一个默。1998 至今，随着 Java 2 一系列新技术如 Java2D, Java3D, Swing, Java SOUND, EJB, Servlet, JSP, CORBA, XML 等的引入以及 JVM 自身安全策略的完善、

效率提高,新一轮 Java 热潮再次被掀起。目前,国内电子商务、金融、证券、邮电、电信等行业的大部分系统几乎都正在或者准备采用部分 Java 技术来实现,预计 Java 应用将很快被国内广大行业用户所接受,Java 技术水平也将逐步与世界同步。

## 1.1.2 Java 的现状

未来的计算机体系是基于 Web 的分布式计算,Java 无疑是当前满足这一趋势的最佳语言,由于其本身的开放性体系以及简洁的语法,国内 Java 技术应用得以蓬勃发展起来。国内企业级应用中的关键技术和内容主要集中在数据访问、数据格式转换、逻辑控制、数据传输、数据表示等方面,而这些都是 Java 的优势所在,因而在实际应用中,Java/J2EE (Java 2 企业版)俨然已经成为不可替代的企业级应用解决方案。

企业 Java 计算模型由四部分组成:标准平台定义 (Enterprise Java APIs)、工业强度的应用服务器 (Application Servers)、构件架构 (Component Frameworks) 和简化编码工作的开发工具 (Development Tools)。

### 1. Enterprise Java Beans 和 Enterprise JavaBean APIs

Enterprise JavaBeans (EJB) 使开发者只编写一次组件,然后便可在最适合他们的应用程序和企业需要的服务器环境中使用它们。标准化的 Enterprise JavaBean APIs 使这一切成为可能。正如 Sun 公司在 Enterprise JavaBeans——Java 的服务器构件中所说明的企业的 Java 平台由一套标准的应用程序编程接口 (API) 到一套核心的企业类基础服务 (其中包括生命周期、命名、远程唤醒、消息处理、交易、数据库访问和管理) 组成。这些基础访问经常是使用不同的产品和技术在不同的平台上实现的,所以很难创建可移植的企业类应用程序系统。Java Enterprise APIs 提供了一个无需考虑实现方式,为服务奠定基石的公共接口。

### 2. 应用程序服务器

应用程序服务器为执行由 Enterprise JavaBeans 创建的中间件提供了一个平台。这些服务器必须具有高度的可伸缩性以支持多用户。用户端可安全地且同时访问应用程序。应用程序能够在任何服务器平台上执行。

### 3. 组件架构

组件是可被用来构造其他应用程序系统的应用程序。在企业内部,重要的部件应该能提供安全的数据库访问。组件可被方便地导入开发工具中并用来为快速开发基于 Java 的商业应用程序提供架构。它们被用来设置应用程序并由 Web 服务器或数据库系统执行。这些组件遵从 Enterprise JavaBeans 的规范。

### 4. 开发工具

企业 Java 开发工具为创建 Java 兼容的应用程序、Applet, Servlets 和 JavaBean 组件提供了一个途径。通过将 Java 客户端自动连接到现存的服务器数据、交易和应用程序上,客户可以利用现存的商业应用程序和 Web 进行日常的商业运作。

### 1.1.3 应用范围

作为一门正处于发展高峰期的程序设计语言，Java 深刻地体现了“设计”的精髓，严谨清晰的结构使其具有良好的扩展能力和广阔的发展空间，简洁的语法提供了良好的可维护性，同时，通过降低程序员编码的难度使其能够将更多的精力专注于业务逻辑的设计与系统框架的构建。在可以预见的未来 3~5 年内，Java 将进一步巩固其在服务器端确立的战略优势并同时向嵌入式系统等应用中延伸。

由于 Java 语言具有许多优秀特性，所以其应用前景必然美好，未来发展肯定会与互联网的发展需求绑定。Java 将应用于以下领域：

- 所有面向对象的应用开发。
- 软件工程中的需求分析、系统设计、开发实现和维护。
- 中小型多媒体系统设计与实现。
- 消息传输系统。
- 分布计算交易管理应用（JTS/RMI/CORBA/JDBC 等技术应用）。
- Internet 的系统管理功能模块的设计，包括 Web 页面的动态设计、网站信息提供管理和交互操作设计等。
- Intranet（企业内部网）上完全基于 Java 和 Web 技术的应用开发。
- Web 服务器后端与各类数据库连接管理器（队列、缓冲池）。
- 安全扫描系统（包括网络安全扫描、数据库安全扫描、用户安全扫描等）。
- 网络/应用管理系统。
- 其他应用类型的程序。

### 1.1.4 Java 语言对软件开发技术的影响

曾有人预言：“Java 语言的出现，将会引起一场软件革命”，这是因为 Java 语言能在执行码（二进制码）上兼容，这样以前所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。时至今日，Java 的优势已经不再仅限于跨平台和动态页面显示，更多的优势逐步在支持 CORBA、Web 服务器后端处理以及应用整合等方面的体现。Java 语言对现在以及未来软件工程和软件发展可能产生影响，可以从以下几个方面分析。

#### 1. 软件工程

Java 适用于 Web 的许多特征，比如平滑移植、安全、健壮和可扩展等，为面向对象软件工程设计提供了实现工具和想向空间。遵循面向对象思想，软件生命周期的各个部分都将从面向对象的角度出发实现。比如用户需求分析将被划分成具有不同状态（数据）和行为的模块，不同模块之间的联系通过消息实现等。

#### 2. 软件的开发方法

由于 Java 语言的面向目标的特性，越来越多的人在用 O-O（Object Orient，面向对象）

开发时都在选择使用 Java 技术与方法来开发，这与 Java 的面向对象特性、模块化特性、安全性以及适应 IT 技术发展密不可分。

### 3. Web 技术

Java 提供的 CGI、ASP 替代方法 Servlet/JSP 以及 EJB 都是与互联网发展需求相呼应的，由此可见，Java 在未来必然紧跟互联网发展趋势，采用 Java 解决企业基于 Web 的应用是明智的选择。

### 4. 分布式计算

Java 对 CORBA, RMI (远程方法调用), EJB, XML 和网络处理的支持及应用将使未来企业分布计算向其倾斜。

### 5. 图形处理与多媒体效果

Java2 以及以后版本提供了 Swing, 2D, 3D, Sound 等图形、图像和多媒体支持类库，弥补了早期 Java 版本在此方面的不足。Java 的媒体处理能力使程序开发出的动画效果远比 GUI 技术更加逼真，尤其是利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面的资源。

### 6. 产品开发

目前 Java 产品主要有基于 Web 的实时信息发布系统、Web 的邮件系统、用于企业分布式计算的消息传输系统、基于 Servlet/JSP/EJB 开发的电子商务平台、提供 Servlet/JSP/EJB 开发和运行环境的应用服务器、结合 CORBA/XML 等技术开发的分布计算支撑平台等。另外，用 Java 开发的多媒体软件可以实现集可视化、可听化和可操作化于一体的效果。

## 1.2 Java 语言的特点

Java 语言到底是一种什么样的语言呢？它是一个简单的、面向对象的、网络适用的、解释型的、健壮的、安全的、独立于平台的、可移植的、可扩展的、高性能的、多线程的以及动态的程序设计语言。

### 1. Java 语言的简单性

Java 语言最初是为对家用电器进行集中控制而设计的一种语言，因此它必须简单明了，易于学习。Java 语言通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 语言是在 C 和 C++ 语言的基础上产生的，它的风格十分接近 C++ 语言，但要比 C++ 简单得多。Java 语言略去了 C++ 语言中容易引发程序错误的地方，例如指针和内存管理。在高级编程语言的所有特性中，不是绝对需要的 Java 语言都已经删去了，例如 Java 语言没有算符重载、预处理、多维数组、多重继承等。增加自动垃圾收集功能，用于回收不再使用的内存区域，这不但使程序易于编写，而且大大减少了

由于内存而引发的问题。Java 语言为程序开发者提供了丰富的类库，使程序的编写变得容易、简单。Java 语言的简单性还体现在小型化上。Java 解释器、系统模块和运行模块都比较小，适合在小型机器上运行，也适合从网上下载。

## 2. Java 语言的面向对象特性

Java 语言是一种彻底的纯面向对象的程序设计语言，它具有面向对象的四大特点：封装、继承、多态和动态。Java 语言支持单继承类层次结构。这就是说，每个类一次只能继承一个别的类。Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码复用。

## 3. Java 语言的网络适用性

Java 语言是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址很方便地访问网络资源。Java 语言包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库，它提供一个 Java.net 包，通过它可以完成各种层次上的网络连接。因此，Java 语言编写的应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。Java 的语言的另一个 Socket 类提供的可靠流式网络的连接，使程序设计者可以非常方便地创建分布式的客户机（Client）和服务器（Server）应用程序。

## 4. Java 语言的类、类装载性

Java 语言提供了大量的类，以满足网络化、多线程和面向对象系统的需要。

(1) 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等，可以用它简单地实现 Java 语言程序的运行平台。

(2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

(3) 输入、输出包用统一的“流”模型来实现所有格式的 I/O 操作，包括文件系统、网络、输入/输出等。

(4) 低级网络包用于实现 Socket 编程。

(5) 抽象图形用户接口包实现了不同平台的计算机的图形用户接口部件，包括窗口、选单、滚动条、对话框等，使得 Java 语言可以移植到不同平台的机器。

(6) 网络包支持 Internet 的 TCP/IP 协议，提供了与 Internet 的接口。它支持 URL 连接及 WWW 的即时访问，并且简化了客户机/服务器模型的程序设计。

## 5. Java 语言的健壮性

Java 语言在编译和运行时，要对可能出现的问题进行检查，以防止错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时产生错误。通过集成的面向对象的例外处理机制，在编译时，Java 语言提示出可能出现但未被处理的例外，帮助程序员正确进行选择以防止系统的崩溃。另外，Java 语言在编译时还可以捕获类型声明中的许多常见错误，防止动态运行时不匹配的出现。

## 6. Java 语言的安全特性

用于网络、分布环境下的 Java 语言必须防止病毒的入侵, Java 语言不支持指针, 一切对内存的访问都必须通过对象的实例变量来实现, 这样就防止了程序员使用“特洛伊木马”等欺诈手段访问对象的私有成员, 同时也避免了指针操作中容易产生的错误。

## 7. Java 语言的体系结构中立特性

网络一般由各种类型的计算机构成, Internet 也是这样。为了使 Java 程序在任何地方都能运行, Java 解释器生成了与体系结构无关字节码 (bytecode) 指令, 只要安装了 Java 运行时系统, Java 程序就可以在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示, Java 解释器得到字节码后, 对它进行转换, 使之能够在不同的平台上运行。

## 8. Java 语言的可移植特性

与平台无关的特性使 Java 程序可以方便地移植到网络上的不同机器。同时, Java 类库中也实现了与不同平台的接口, 使这些类库可以移植。另外, Java 编译器是由 Java 语言实现的, Java 运行时系统由标准 C 语言实现, 这使得 Java 系统也具有可移植性。

## 9. Java 语言的解释特性

Java 语言是解释执行的。程序运行时, Java 解释器直接对字节码进行解释执行。字节码本身携带了许多编译信息, 使得连接过程更加简单。

## 10. Java 语言的高性能

Java 语言的解释器和其他解释执行的语言如 BASIC 不同, Java 语言字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。

## 11. Java 语言的多线程

多线程机制使应用程序能够并行执行, 并且同步机制保证了对共享的数据的正确操作。通过使用多线程, 程序设计者可以分别用不同的线程完成特定的行为, 而不需要采用全局的事件循环机制, 这样就很容易地实现网络上的实时交互行为。

## 12. Java 语言的动态特性

Java 语言的设计使它适合于一个不断发展的环境, 在类库中可以自由地加入新的方法和事例变量而不会影响用户程序的执行。另外, Java 语言通过接口来支持多重继承, 使之比严格的类继承具有更灵活的方式和扩展性。

# 1.3 与 C 和 C++ 语言的异同

对于变量声明、参数传递、操作符、控制语句等方面, Java 语言使用了和 C 及 C++ 语言

相同的风格，使得熟悉 C 及 C++ 语言的程序员很方便地进行编程。同时，Java 语言为了实现其简单性、分布性、安全性等特性，抛弃了 C 和 C++ 语言中许多不合理的内容。下面介绍 Java 语言和 C 及 C++ 语言的区别。

### 1. 全局变量

在 Java 语言程序中，不能在所有类之外定义全局变量，只能通过在一个类中定义公用的、静态的变量来实现一个全局变量。例如：`class GlobalVar {public static global_var;}` 在类 `GlobalVar` 中定义变量 `global_var` 为 `public static`（公用、静态），使得其他类可以访问和修改该变量。Java 语言对全局变量进行了更好的封装；而在 C 及 C++ 语言中，依赖于不加封装的全局变量常常造成系统的崩溃。

### 2. goto 语句

Java 语言不支持 C 及 C++ 语言中的 `goto` 语句，而是通过例外处理语句 `try, catch, finally` 等代替 C++ 语言中用 `goto` 语句处理遇到错误时跳转的情况，使程序更可读且更结构化。

### 3. 指针

Java 语言不支持 C 及 C++ 语言中的 `goto` 语句，因为由指针所进行的内存地址操作会造成不可预知的错误，同时通过指针对某个内存地址进行显式类型的转换后，可以访问一个 C++ 语言中的私有成员，从而破坏安全性，造成系统崩溃。Java 语言对指针进行完全的控制，程序员不能直接进行任何指针操作，例如，把整数转化为指针或者通过指针释放某一内存地址等。同时，数组作为类在 Java 语言中实现，很好地解决了数组访问越界这一 C 及 C++ 语言中不做检查的错误。

### 4. 内存管理

在 C 语言中，程序员通过库函数 `malloc()` 和 `free()` 来分配和释放内存，C++ 语言中通过运算符 `new` 和 `delete` 来分配和释放内存；再次释放已释放的内存块或未被分配的内存块，会造成系统的崩溃；同样，忘记释放不再使用的内存块也会逐渐耗尽系统资源。而在 Java 语言中，所有的数据结构都是对象，通过运算符 `new` 为它们分配内存堆。通过 `new` 得到对象的处理权，而实际分配给对象的内存可能随程序的运行而改变，Java 运行系统对此自动地进行管理并且进行垃圾（无用内存）收集，有效地防止了由于程序员的误操作而导致的错误，并且更好地利用了系统资源。

### 5. 数据类型的支持

在 C 及 C++ 语言中，对于不同的平台，编译器对于简单的数据类型如 `int` 与 `float` 等分配不同长度的字节数，但在 Java 语言中，对于这些数据类型总是分配固定长度的位数，这就保证了 Java 语言的平台无关性。

### 6. 类型转换

在 C 及 C++ 语言中，可以通过指针进行任意的类型转换，常常带来不安全性，而在 Java

语言中，运行时系统对对象的处理要进行类型相容性检查，以防止不安全的转换。

### 7. 头文件

在 C 及 C++ 语言中，用头文件来声明类的原型以及全局变量、库函数等，在大的系统中维护这些头文件是很困难的。Java 语言不支持头文件，类成员的类型和访问权限都封装在一个类中，运行时系统对访问进行控制，防止对私有成员的操作。同时，Java 语言中用 `import` 语句来与其他类进行通信，以便使用它们的方法。

### 8. 结构体和联合体

在 C 及 C++ 语言中，结构体和联合体中所有成员均为公有，这就带来了安全性问题。Java 语言中不包含结构体和联合体，所有的内容都封装在类中。

### 9. 宏定义

在 C 及 C++ 语言中，用宏定义来实现的代码给程序的可读性带来了困难。Java 语言不支持宏，它通过关键字 `final` 来声明一个常量，以实现宏定义中广泛使用的常量定义。

## 1.4 Java 平台

在设计上，Java 是以软件为运作范畴，并架设在各种硬件环境上的一种纯软件平台，它与一般和硬件有关的各式操作系统（如 Windows, Linux 等）不同。一般所谓的操作平台是指撰写程序时依赖的 API (Application Program Interface)，通常都预先定义在计算机中的操作系统上。因此在 Java 之前，程序设计师就只能按所在环境的 API 编写单一操作系统的程序。若要改写其他操作系统版本的程序，就很困难，或者根本不可能。但 Java 平台虽然不算是一种操作系统，却提供了甚至比现今各种操作系统更深、更广的 API。Java 平台包括 Java 虚拟机 (JVM) 以及 Java API，如图 1-1 所示。

目前 Java 平台有以下 3 种版本，以适应各种需求：

(1) Java 2 Standard Edition J2SE，是 Java 的标准版。

(2) Java 2 Enterprise Edition J2EE，是 Java 的企业版。

(3) Java 2 Micro Edition J2ME，是仅支持 Java 核心类的子集，对不同的移动信息设备进行了抽象。它利用 `Configure` 的概念，定义了两种不同的规范：CLDC 和 CDC。CLDC 针对运算功能有限、电力供应有限的移动设备，如手机、PDA；CDC 针对运算能力较强、电力供应相对较充足的嵌入装置，如电冰箱、机顶盒和空调。

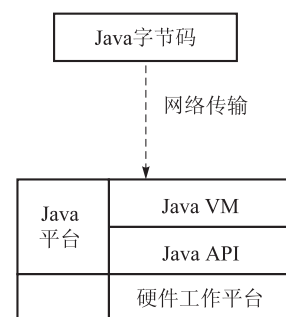


图 1-1 Java 平台

## 1.5 Java 程序种类

Java 是一种网络程序语言，它能写出包含前端（Applet）、本地（Application）及后端（Servlet）的程序，虽然这三者的结构不同，但基本的语法都一样，所以能彼此沟通，构造出一所谓的平台。现在又加上了 JSP（Java Server Pages）作为服务器端的加强，使得开发网络程序更快速，让 Java 平台更加完整了。

### 1.5.1 Applet

Applet（小程序）是一种可嵌入于 Web 文件中的一种小型程序，其程序因网络传输速度关系都很短小，但在应用上往往比 Application 更有用，一般都是通过浏览器（比如 Internet Explorer）来观看 Applet。图 1-2 显示的是一个三维动画 Applet 范例。



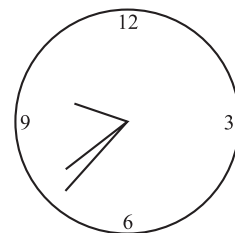
图 1-2 Applet 范例

### 1.5.2 Application

Application（应用程序）是在命令行环境下执行的程序。程序中由 main()方法作为程序起点，执行顺序由程序本身来控制，而 Applet 却是由浏览器来掌管的。选择 Applet 还是 Application 来开发，要视所欲表达的内容与传输机制而定。Application 程序可以独立进行，不必使用支持 Java 的浏览器或专门的执行工具。另外，Application 程序与 Applet 相比，限制更少。例如，Application 可以访问本地的文件系统，而 Applet 程序却受到安全机制的限制，无法实现本地访问。Application 程序还可以加载任何本地程序，包括动态链接库等，Applet 程序却无法实现。

### 1.5.3 Servlet

Servlet 是用 Java 代码编写的服务器端应用程序，用于处理客户机和服务器之间的消息传递。Servlet 可以通过动态构造一个发回客户机的响应来响应客户机请求，可在不同平台和不同的 Web 应用服务器间移植，适合于实现复杂的商业应用逻辑，例如数据查询、聊天室或日历等。图 1-3 显示的是一个由 Servlet 产生的计时时钟。



星期一四月 05 21:38:37 2004

图 1-3 由 Servlet 产生的计时时钟

### 1.5.4 JSP

JSP 是 Java Server Pages 的缩写，是 Sun 公司于 1999 年推出的一种动态网页技术标准。

JSP 是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术，利用这一技术可以建立安全、跨平台的先进动态网站，这项技术还在不断地更新和优化。

## 1.6 Java 应用程序开发

在看过了多姿多彩的 Java 世界后，是不是忍不住要亲手调一杯美味的 Java 咖啡来尝尝？要调出一杯香浓可口的 Java 咖啡，对一个初学者，不是三下两下就能办到的。因为编写 Java 程序和编写一般在 PC 窗口环境下程序不一样，因为 Java 是一种网络程序语言，而不仅是 PC 上的一种语言。要学好 Java，首先要了解 Java 程序的运作原理及开发流程，其次要了解面向对象概念，接着熟悉 Java 基本程序结构，最后学习使用开发工具。

Java 语言设计的理念就是以整个 Internet 为运作平台，以程序代码能在各种操作系统及各种机器上运行为目标。为此发展出 Java 字节码 (Byte Code)、Java 虚拟机 (Java Virtual Machine, JVM) 及 Java API 等达到此目的。Java 字节码 (.class 文件) 是一种具有可移植性的程序代码，与一般程序通过编译器编译成的机器码不同，它不是真正令 CPU 运行的程序代码，故又称为伪码。Java 字节码是由 Java 源文件通过 Java 编译器编译而成，如图 1-4 所示。

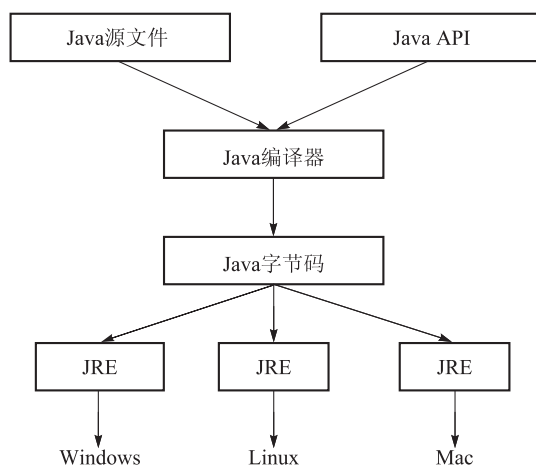


图 1-4 Java 的编译

在客户端收到网络所传输过来的字节码后，便可通过一种与各平台有关的运行阶段环境 (Java Runtime Environment, JRE) 中的 JVM 以及 JRE 所提供的运行时所需类库，转换成本地码 (Native Code, CPU 机器指令)，如此便可达到“一次编写，到处运行”的目的。JRE 所包含的软件如图 1-5 所示。

JVM 包括类加载器、字节确认器以及 JIT 实时编译器。类加载器用来取得从网络获取的或存于本地机器上的类文件字节码。字节确认器接着确认这些类文件是否有正确的格式，以确定在运行时不会有破坏内存的行为。而 JIT 编译器可将字节码转换成本地机器码，使原本是直译式的 VM 能提高到编译式的运行效果。Java HotSpot Runtime 就是实时编译技术的一大杰作，如图 1-5 所示。

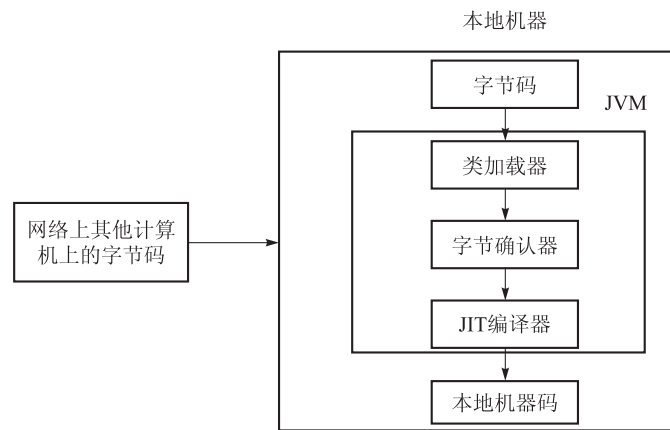


图 1-5 JVM

### 1.6.1 JDK 环境

JDK 可以免费从 Internet 上获得，很多网站都提供 JDK 的下载，如：<http://java.sun.com/products/jdk/> 站点。Sun 公司提供了多种操作系统平台的 JDK，这里以最常见的 Windows 操作系统为例，说明 JDK 的安装和设置。

Java 编程人员一般都是先从 Sun 公司提供的开发包 JDK (Java Developers Kits) 开始学习的，Java 不仅提供了一个丰富的语言和运行环境，而且还提供了一个免费的 Java 开发工具集 JDK。编程人员和最终用户可以利用这个工具来开发 Java 程序或调用 Java 内容。

JDK 主要包括以下工具：

- **Javac:** Java 语言编译器，输出结果为 Java 字节码。
- **Java:** Java 字节码解释器。
- **Javap:** Java 字节码分解程序，本程序返回 Java 程序的成员变量及方法等信息。
- **AppletViewer:** 小应用程序 Applet 浏览工具，用于测试并运行 Applet。
- **Jdb:** Java 调试工具。
- **Javadoc:** Java 代码注释自动生成工具。

Java 开发环境还包括 Java 类库（包括 I/O 类库、用户界面、网络类库等）和 HotJava Web 浏览器。其中，HotJava 浏览器提供了在 Web 环境下运行 Java 代码的一个运行系统，而且还为 Web 开发人员提供了一个 Java 开发框架。Java 解释器是面向 Java 程序的一个独立运行系统，它可以一种稳定、高性能的方式运行各种平台的 Java 字节码。Java 编译器则用于生成这些字节码。

### 1.6.2 Java 程序的编译

Java 程序的编译程序是 javac.exe。javac 命令将 Java 程序编译成字节码，然后可用 Java 解释器来解释执行这些字节码。Java 程序源码必须存放在后缀为 .java 的文件里。对于 Java

程序里的每一个类，javac 都将生成与该类名称相同但后缀名为.class 的文件。编译器把.class 文件放在.java 文件的同一个目录里，除非用了-d 选项。当引用到某个自己定义的类时，必须指明它们的存放目录，这就需要利用环境变量参数 CLASSPATH。环境变量 CLASSPATH 由一些被分号分割的路径名组成。如果传递给 javac 编译器的源文件里引用到的类定义在本文件和传递的其他文件中找不到，则编译器会按 CLASSPATH 定义的路径来搜索。例如“CLASSPATH=.; d:\jbuilderX\classes”，编译器先搜索当前目录，如果没搜索到，则继续搜索 d:\jbuilderX\classes 目录。注意，系统总是将系统类的目录默认加在 CLASSPATH 后面，除非使用-classpath 选项来编译。在命令行界面下敲入 javac，将出现如图 1-6 所示的界面。

```

C:\jdk1.3.0_01\bin>javac
用法: javac <选项> <源文件>
可能的选项包括:
-g                生成所有调试信息
-g:none          生成无调试信息
-g:{lines,vars,source} 生成只有部分调试信息
-O              优化:可能妨碍调试或者增大类文件
-nowarn         生成无警告
-verbose        输出关于编译器正在做的信息
-deprecation    输出使用了不鼓励使用的API的源程序位置
-classpath <路径> 指定用户类文件的位置
-sourcepath <路径> 指定输入源文件的位置
-bootclasspath <路径> 覆盖自举类文件的位置
-extdirs <目录<多个>> 覆盖安装的扩展类的位置
-d <目录>       指定输出类文件的位置
-encoding <编码> 指定源文件中所用的字符集编码
-target <版本>  生成指定虚拟机版本的类文件

```

图 1-6 javac 的使用方法

具体各项内容解释如下：

- -classpath: 定义 javac 搜索类的路径。它将覆盖默认的 CLASSPATH 环境变量的设置。
- -d: 指明类层次的根目录，格式如下: javac -d my\_dir MyProgram。Javac 把 MyProgram 编译的.class 文件存放在 my\_dir 目录里。
- -g: 带调试信息编译，调试信息包括行号与使用 Java 调试工具使用到的局部变量信息。
- -nowarn: 关闭警告信息，编译器将不显示任何警告信息。
- -O: 优化编译。
- -verbose: 显示被编译的源文件名和被加载的类名。

### 1.6.3 Java 程序的执行

java.exe 是 Java 语言解释器，解释 Java 字节码，在命令行中敲入 java，将出现如图 1-7 所示的界面。

其中，class 参数是要执行的类名称。注意，在类名称后的任意参数都将传递给要执行类的 main 函数。Java 执行完 main 函数后退出，除非 main 函数创建了一个或多个线程。如果 main 函数创建了其他线程，则 Java 总是等到最后一个线程结束后才退出。