

Broadview  
www.broadview.com.cn

全国计算机等级考试教程

(新大纲版)

# Java 语言程序设计 (二级)

**标准、实用、严谨**

《全国计算机等级考试教程》丛书编委会 主编  
李 楹 梁 妍 夏云龙 刘 洋 等编著

解决考什么——紧紧依据考试大纲，精讲考试重点、难点  
解决怎么考——透彻深入解析例题，整体把握考试形式  
立足笔试特点——综合历年经典试题，强化笔试自我训练



超级模拟软件

立足上机操作——附送“超级模拟软件”，真实感受上机环境



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# 全国计算机等级考试教程

## Java语言程序设计（二级）

（新大纲版）

《全国计算机等级考试教程》丛书编委会 主编  
李 楹 梁 妍 夏云龙 刘 洋 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书是根据 2004 年教育部考试中心公布的《全国计算机等级考试大纲二级 (Java 语言程序设计)》编写而成的。本书包括 Java 语言概述, Java 语言基础知识, Java 语言中面向对象的特性, 异常处理与 I/O 流, 编写用户界面程序, Java 的线程和 Java Applet。本书每章都指出了重点和难点, 并配有例题解析和练习题及答案, 书后有模拟试题以供练习。

本书特别适合于作为全国计算机等级考试 Java 语言程序设计 (二级) 的教材和参考书, 也适合于作为相关计算机专业课程的教材或教学辅导书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

## 图书在版编目 (CIP) 数据

Java 语言程序设计. 二级: 新大纲版 / 李楹等编著. —北京: 电子工业出版社, 2005.1  
全国计算机等级考试教程  
ISBN 7-121-00726-6

I. J… II. 李… III. JAVA 语言—程序设计—水平考试—自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 135460 号

责任编辑: 毕 宁

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 16.25 字数: 370 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 8000 册 定价: 29.00 元 (含光盘 1 张)

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

# 本书导读

(本导读用于说明学习顺序, 图中例子仅作为示意用)

## 内容及考核知识点

- ☑ 五种基本数据类型
- ☑ 常数的类型及其表示
- ☑ 变量的定义及初始化
- ☑ 运算符的优先级和结合性
- ☑ 表达式及其求值规则
- ☑ 数据的输入与输出

## 概括学习内容和知识点

## 学习环节

## 精解内容和相关知识

## 关注考什么

### 3.5 表达式

表达式是程序中使用最频繁、最大量出现的计算手段, 程序中要求计算机进行某种计算或运算主要是通过表达式实现的。不同的表达式进行不同的运算, 达到不同的目的。

#### 3.5.1 算术表达式

算术表达式用+, -, \*, /, %, ++, --等运算符连接数值型的运算量来完成数值计算的功能。其中, %只能用于 int 型或 char 型。

##### ☞ 重点

- ☛ 算术运算符中, % 只能对整型运算量施加运算。在计算两个整数的余数时, 余数的符号与被除数相同。例如,  $3\%( -2)=1$ ,  $-3\%2=-1$ ,  $3\%2=1$ ,  $-3\%( -2)=-1$ 。
- ☛ 计算算术表达式的值时, 运算符的优先级和结合性十分重要。例如:  

```
int a=3;
printf("%d\n", ++a++a);
```

的输出为-1, 因为在计算表达式++a++a时, 先进行++a的运算, 然后再进行“-”运算, 而且第一个a被自加后将影响第二个a的值。

##### ☞ 难点

- ☛ 在C语言中, 整数有多种类型。在类型转换时, 同一个数可能会产生不同的值。例如, unsigned a=32768 转换成 int 型时, 将变成-32768; int a=-1 转换成 unsigned 型时将变成 65535。

## 总结



## 考点

- ☛ 基本数据类型及其长度限制。
- ☛ 不同类型常数的书写和识别。
- ☛ 变量的定义和初始化。
- ☛ 运算符的优先级和结合性。
- ☛ 各种表达式的书写和求值。
- ☛ 赋值表达式和赋值语句的区别。
- ☛ 正确进行数据的输入和输出。

## 看例题

### 3.8 练习题

#### 一、选择题

1. 已知字母A的ASCII码为十进制的65, 下面的程序输出是\_\_\_\_\_。

```
main()
{
    char ch1, ch2;
    ch1='A'+5*2;
    ch2='A'+6*2;
    printf("%d,%c\n", ch1, ch2);
}
```

- A) 67,D      B) 6,C      C) 6,D      D) 不确定的值
2. 设 int 型占 2 字节, 则 unsigned int 所能表示的数据范围是\_\_\_\_\_。
- A) 0~65535      B) -32768~32767      C) 1~65536      D) 0~32767

### 3.7 例题解析

#### 一、选择题

1. C语言中允许的基本数据类型包括\_\_\_\_\_。 【答案】B

- A) 整型、实型、逻辑型  
B) 整型、实型、字符型  
C) 整型、字符型、逻辑型  
D) 整型、实型、逻辑型、字符型

2. C语言中能用八进制表示的数据类型是\_\_\_\_\_。 【答案】A

- A) 字符型、整型      B) 整型、实型  
C) 字符型、实型、双精度型  
D) 字符型、整型、实型、双精度型

【解析】C语言中允许的基本数据类型有五种, 即字符型 char、整型 int、浮点型 float、双精度型 double 和无值类型 void。而浮点型和双精度型都是实型, 特别是C语言没有逻辑型。

【解析】C语言规定, 只有字符型和整型数据才能用十进制、八进制或十六进制表示, 而浮点型和双精度型数据只能用十进制表示。

## 做练习

## 模拟 笔试

## 掌握考试形式

## 典型题目分析

### 12.1 笔试模拟试卷(一)

#### 12.1.1 全国计算机等级考试二级笔试试卷

基础部分和C语言程序设计  
(考试时间 120 分钟, 满分 100 分)

#### 一、选择题

## 模拟上机

### 全国计算机等级考试 考试登录

准考证号: \_\_\_\_\_  
考生姓名: \_\_\_\_\_  
身份证号: \_\_\_\_\_

运行模拟软件, 感受上机考试

总结历年考题, 模拟笔试内容

## 备考环节

## 丛 书 编 委 会

顾问：刘瑞挺

主编：边莫英

电子工业出版社计算机图书事业部（北京博文视点资讯有限公司）

编委（以姓氏笔画为序）：

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 丁玄功 | 万振凯 | 于 健 | 于晓华 | 马希荣 |
| 王正明 | 王洪权 | 王温君 | 王毓珠 | 王慧芳 |
| 叶 华 | 刘 洋 | 孙华志 | 孙 锋 | 曲建民 |
| 张立新 | 张 炜 | 张海涛 | 李兰友 | 李 军 |
| 李 辉 | 李 楹 | 邵秀丽 | 陈慰国 | 范俊弟 |
| 姜丽芬 | 赵 喆 | 郝嘉林 | 夏云龙 | 高福成 |
| 梁 妍 | 韩其睿 | 鲁声清 | 潘旭华 |     |

# 出版说明

## 关于丛书

《全国计算机等级考试教程》自 2001 年首次推出以来, 历经多次改版和修订, 受到了广大读者的普遍欢迎, 特别是得到了考生们的一致推崇。本次修订是根据 2004 年教育部考试中心颁布的《全国计算机等级考试大纲》编写而成的。

作为出版者, 我们深深感到, 只有真正好的作品才能“一石激起千层浪”。综观现在的等考书籍市场, 鱼龙混杂, 相当一部分的小出版商只顾卖书的眼前利益, 利欲熏心, 毫无社会责任感, 出版了一大批内容质量低下, 错误百出, 印装粗糙, 胡乱定价的等考书籍。购买这样的书籍去准备考试, 结果可想而知, 很多考生深受其害, 以至屡战屡败。

回想我们在本套丛书历次修订出版时定下的原则——“标准、实用、严谨”, 我们为之付出的是无数次的挑灯夜战, 与作者并肩上阵, 仔细推敲, 反复修改, 目标是推出一套切实能为广大考生服务的等考用书。小而言之, 是为考生考试过关; 大而言之, 考生通过优秀的书籍增长了知识, 提高了自身的资质, 进而可以获得更理想的工作与职位, 实现更大的人生价值, 这样的间接影响恐怕是无法度量的。因此, 作为出版者, 我们把对社会价值的理解落实在我们与考生之间坚不可摧的关系上, 因为我们与考生有着同样深刻的共识, 那就是: **知识改变命运**。

当然, 我们仍然保持着奉献的激情, 并坚定着一个始终不渝的追求——精益求精。

为了更好地为考生服务, 经过深入调研, 我们在 2004 年新大纲的基础上推出了新版系列。在这个版本中:

- 因应新大纲的内容, 做了增、删、改等调整;
- 修订了部分内容, 在新大纲的基础上扩充了部分相关知识, 以满足目前相关专业课教学的需要;
- 综合读者反馈和自查, 修改了以前历次版本中发现的错漏之处。

因此, 本套丛书突出强调了以下几个特点:

1. **解决考什么**——紧紧依据考试大纲, 精讲考试重点、难点
  2. **解决怎么考**——透彻深入解析例题, 整体把握考试形式
  3. **立足笔试特点**——综合历年经典试题, 强化笔试自我训练
  4. **立足上机操作**——提供无忧公司出品的“超级模拟软件”, 真实感受上机环境
- 我们真诚希望, 考生凭借本套丛书, 能轻松通过考试, 实现自己的目标。

## 本书的主要内容和读者对象

等级考试二级中的公共基础知识部分放在本套丛书中的《公共基础知识(二级)》一书中集中介绍, 本书中只包含公共基础知识部分, 不包含程序设计部分内容, 其模拟试题包含在其他考试科目模拟试卷中。

本书包括 Java 语言概述, Java 语言基础知识, Java 语言中面向对象的特性, 异常处理与 I/O 流, 编写用户界面程序, Java 的线程和 Java Applet。本书每章都指出了重点和难点, 并配有例题解析和练习题。附录 A 为练习题答案。

本书适合于作为全国计算机等级考试 Java 语言程序设计(二级)的教材和参考书, 也适合于作为相关计算机专业课程的教材或教学辅导书。

## 关于作者

本丛书编委会由长期在计算机教育和等级考试培训领域一线的专家、教授组成, 他们有丰富的教学经验, 能准确把握考试要点和难点, 了解考生在学习时会遇到的诸多问题, 因此全书内容的安排有非常强的针对性, 读者掌握了全书的内容就一定能通过考试。

本书第 1 章、第 2 章、第 5 章由李楹编写, 第 3 章由梁妍编写, 第 4 章由夏云龙编写, 第 6 章由刘洋编写。此外, 梁景莲等同志也参与了本书的资料收集工作, 马希荣同志对全书进行了审阅, 并提出了宝贵的建议。全书由李楹统稿。

## 反馈

如果在阅读和学习的过程中, 读者发现有叙述不清或有疑问的地方, 可与我们联系。

由于水平有限和时间仓促, 书中难免存在疏漏之处, 欢迎广大读者批评指正。对书中内容的勘误, 读者可上 [www.broadview.com.cn](http://www.broadview.com.cn) 网站进行查阅。同时, 为进一步鼓励读者积极参与对本书的勘误, 我们将对首先发现错误的读者或提供重大建设性意见和建议的读者, 赠送纪念品。

问题及意见反馈请发往:

(100036) 北京万寿路 173 信箱电子工业出版社 计算机图书事业部 收

或通过电子邮件:

[editor@broadview.com.cn](mailto:editor@broadview.com.cn) [jsj@phei.com.cn](mailto:jsj@phei.com.cn)

电子工业出版社计算机图书事业部  
(北京博文视点资讯有限公司)



# 目 录

|                     |    |
|---------------------|----|
| 第1章 Java 语言概述       | 1  |
| 1.1 Java 语言的发展和应用   | 1  |
| 1.1.1 Java 的产生与发展   | 1  |
| 1.1.2 Java 的应用      | 1  |
| 1.2 Java 语言的特点和实现机制 | 2  |
| 1.2.1 Java 语言的特点    | 2  |
| 1.2.2 Java 的实现机制    | 3  |
| 1.3 开发环境设置          | 5  |
| 1.3.1 J2SE 的下载和安装   | 5  |
| 1.3.2 设置 J2SE       | 6  |
| 1.3.3 开发环境          | 7  |
| 1.4 Java 体系结构       | 8  |
| 1.4.1 J2SE 目录结构     | 8  |
| 1.4.2 Java 的 API 结构 | 9  |
| 1.4.3 Java 程序结构     | 9  |
| 1.4.4 Java 程序的编译及运行 | 10 |
| 1.5 例题解析            | 12 |
| 1.6 练习题             | 15 |
| 第2章 Java 语言基础知识     | 17 |
| 2.1 Java 简单数据类型及转换  | 17 |
| 2.1.1 常量和变量         | 17 |
| 2.1.2 基本数据类型        | 19 |
| 2.1.3 类型转换          | 22 |
| 2.2 运算符和表达式运算       | 23 |
| 2.2.1 运算符           | 23 |
| 2.2.2 表达式运算         | 27 |
| 2.3 Java 语言的基本语句    | 28 |
| 2.3.1 表达式语句         | 28 |
| 2.3.2 分支语句          | 29 |
| 2.3.3 循环语句          | 32 |
| 2.3.4 注释语句          | 37 |
| 2.4 Java 程序规范       | 39 |
| 2.5 例题解析            | 39 |
| 2.6 练习题             | 43 |



|                      |    |
|----------------------|----|
| 第3章 Java语言中面向对象的特性   | 46 |
| 3.1 面向对象编程基础         | 46 |
| 3.2 类                | 47 |
| 3.2.1 声明类            | 47 |
| 3.2.2 定义成员变量         | 48 |
| 3.2.3 定义方法           | 48 |
| 3.2.4 访问说明符          | 50 |
| 3.3 对象               | 51 |
| 3.3.1 声明对象           | 51 |
| 3.3.2 访问对象           | 52 |
| 3.3.3 初始化对象          | 54 |
| 3.3.4 撤销对象           | 57 |
| 3.4 方法               | 57 |
| 3.4.1 递归调用方法         | 57 |
| 3.4.2 参数传递           | 58 |
| 3.4.3 重载方法           | 60 |
| 3.4.4 main()方法       | 60 |
| 3.5 面向对象高级特性         | 62 |
| 3.5.1 类继承            | 62 |
| 3.5.2 实例成员和类成员       | 65 |
| 3.5.3 final 修饰符      | 67 |
| 3.5.4 abstract 修饰符   | 68 |
| 3.6 接口与包             | 70 |
| 3.6.1 接口             | 70 |
| 3.6.2 包              | 71 |
| 3.7 数组               | 72 |
| 3.7.1 定义数组           | 72 |
| 3.7.2 初始化数组          | 72 |
| 3.7.3 访问数组元素         | 73 |
| 3.7.4 二维数组           | 73 |
| 3.8 字符串处理            | 74 |
| 3.8.1 String 类       | 74 |
| 3.8.2 StringBuffer 类 | 78 |
| 3.9 Java 类库的常用类和接口   | 79 |
| 3.9.1 java.lang      | 79 |
| 3.9.2 java.util      | 80 |
| 3.9.3 java.io        | 80 |
| 3.9.4 java.net       | 81 |
| 3.9.5 java.applet    | 81 |

|            |                     |            |
|------------|---------------------|------------|
| 3.10       | 例题解析                | 82         |
| 3.11       | 练习题                 | 88         |
| <b>第4章</b> | <b>异常处理与 I/O 流</b>  | <b>94</b>  |
| 4.1        | 异常 (Exception)      | 94         |
| 4.1.1      | 什么是异常               | 94         |
| 4.1.2      | 异常与错误               | 94         |
| 4.1.3      | 处理异常                | 94         |
| 4.2        | 输入/输出 (I/O)         | 106        |
| 4.2.1      | 什么是 I/O 流           | 106        |
| 4.2.2      | 字节流和字符流             | 106        |
| 4.2.3      | 控制台 I/O             | 108        |
| 4.2.4      | 文件                  | 111        |
| 4.2.5      | 文件 I/O              | 115        |
| 4.3        | 序列化 (Serialization) | 119        |
| 4.3.1      | 什么是序列化              | 119        |
| 4.3.2      | 接口                  | 119        |
| 4.3.3      | 类                   | 120        |
| 4.3.4      | 示例                  | 122        |
| 4.4        | 类集 (Collection)     | 124        |
| 4.4.1      | 什么是类集               | 124        |
| 4.4.2      | 接口                  | 124        |
| 4.4.3      | 类                   | 127        |
| 4.4.4      | 迭代函数                | 133        |
| 4.5        | 例题解析                | 136        |
| 4.6        | 练习题                 | 138        |
| <b>第5章</b> | <b>编写用户界面程序</b>     | <b>140</b> |
| 5.1        | 概述                  | 140        |
| 5.2        | 用 AWT 编写图形用户界面      | 140        |
| 5.2.1      | java.awt 包          | 140        |
| 5.2.2      | 常用容器                | 141        |
| 5.2.3      | 常用组件                | 147        |
| 5.2.4      | 布局管理器               | 160        |
| 5.2.5      | 颜色与字体               | 168        |
| 5.3        | AWT 事件处理模型          | 169        |
| 5.3.1      | 事件类                 | 171        |
| 5.3.2      | 事件监听器               | 173        |
| 5.3.3      | 事件适配器               | 174        |
| 5.3.4      | 使用内部类和匿名类           | 176        |

|                                    |            |
|------------------------------------|------------|
| 5.4 用 Swing 编写图形用户界面               | 177        |
| 5.4.1 Swing 概述                     | 177        |
| 5.4.2 Swing 组件和容器                  | 180        |
| 5.4.3 布局管理器                        | 187        |
| 5.4.4 Swing 的事件处理机制                | 187        |
| 5.4.5 Swing 程序简介                   | 187        |
| 5.5 例题解析                           | 190        |
| 5.6 练习题                            | 193        |
| <b>第 6 章 Java 的线程和 Java Applet</b> | <b>195</b> |
| 6.1 线程的概念和作用                       | 195        |
| 6.1.1 线程的概念                        | 195        |
| 6.1.2 线程的作用                        | 200        |
| 6.2 线程的共享、同步与互斥                    | 202        |
| 6.2.1 线程的共享、同步与互斥                  | 202        |
| 6.2.2 多线程的同步                       | 204        |
| 6.2.3 死锁及其防治                       | 207        |
| 6.3 小应用程序概念                        | 208        |
| 6.4 安全机制                           | 209        |
| 6.4.1 Applet 的安全限制                 | 209        |
| 6.4.2 Java 中的沙箱模型                  | 209        |
| 6.5 Applet 执行过程                    | 210        |
| 6.6 Applet 的图形绘制                   | 212        |
| 6.6.1 paint(), repaint()和 update() | 212        |
| 6.6.2 方法的交互                        | 213        |
| 6.7 Applet 的窗口                     | 213        |
| 6.8 Applet 的工作环境                   | 215        |
| 6.9 Java Application 和 Applet      | 215        |
| 6.10 例题解析                          | 216        |
| 6.11 练习题                           | 224        |
| 全国计算机等级考试二级笔试试卷一                   | 226        |
| 全国计算机等级考试二级笔试试卷二                   | 233        |
| 附录 A 练习题答案                         | 240        |

# 第 1 章 Java 语言概述

## 内容及考核知识点

- ☒ Java 语言的发展与应用
- ☒ Java 语言的特点与实现机制
- ☒ 开发环境设置
- ☒ Java 体系结构

## 1.1 Java 语言的发展和应用

### 1.1.1 Java 的产生与发展

1991 年, Sun MicroSystem 公司的 Jame Gosling、Bill Joe 等人为在电视、烤箱等家用消费类电子产品上进行交互式操作而着手设计和开发出一种语言, 称之为 Oak (橡树)。该语言采用了许多 C 语言的语法, 提高了安全性, 并且是面向对象的语言, 但是 Oak 语言在商业上并未获得成功。

1994 年下半年, Internet 迅猛发展, Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点非常符合互联网的需要, 于是改进了该语言的设计, 并命名该语言为 Java (印尼一个盛产咖啡的小岛) 语言。1995 年 Sun 公司正式推出 Java, 从而造就了一代成功的编程语言。

Java 应用于 WWW 后, 因其在 WWW 上支持各种动态和交互效果而迅速闻名, 此后, 各种浏览器竞相支持 Java, 大量用 Java 编写的软件产品纷纷涌现。1995 年美国的著名杂志 PC Magazine 评选 Java 语言为十大优秀科技产品。微软公司总裁比尔·盖茨先生在经过研究后认为“Java 语言是长时间以来最卓越的程序设计语言”。当时有人预言: Java 将是网络上的“世界语”, 今后所有用其他语言编写的软件统统都要用 Java 语言来改写。总之, “Java 语言的出现是计算机界的一件大事”。

目前, Java 语言已经成为最流行的网络编程语言, Java 这杯咖啡已经香飘世界。

### 1.1.2 Java 的应用

Java 程序有可以单独运行的 Java 应用程序, 也有应用于网页的 Java 小应用程序(Applet)。

Java 实际是一种常规用途的程序设计语言, 可解决任何类型的问题。在这点上, Java 的优势不仅反映在它的可移植性上, 也反映在编程本身, 即 Java 提供的许多特性使我们能在较短的时间里创建出更健壮的程序。但其最明显的问题是执行速度较慢 (Java 对此一直在进行多方面的调整)。此外, Java 本身也存在一些限制, 使得它不十分适合解决某些特殊的编程问题。

Java Applet 是动态、安全、跨平台的网络应用程序, 它可以嵌入网页, 并在浏览器中运行。网络用户访问网页时, 其中的 Applet 也随之从网络上进行传输, 然后在支持 Java 的浏览

器中运行。它给网页增添动态效果和交互操作。Java 的这个特性使它可以最大限度地利用网络。

J2EE (Java 2 Enterprise Edition) 是 Sun 对 Java 改进后的一个企业应用程序的开发平台, 包括了对 EJB、Servlet、Java Server Page、JNDI、XML 等的支持。在这个平台上可以开发瘦客户端的多层体系结构的企业应用程序。其中, Servlet 是用 Java 编写的 Server 端程序, 它与协议和平台无关。Servlet 运行于 Java enabled Web Server 中。Java Servlet 可以动态地扩展 Server 的能力, 并采用“请求—响应”模式提供 Web 服务。JSP 是 Java Server Page 的缩写, 是 Sun 公司出品的 Web 开发语言, 它类似于 Microsoft 公司的 ASP, 但由于它的跨平台性, 愈来愈受到广泛的应用。Servlet 与 JSP 之间的交互为开发 Web 服务提供了优秀的解决方案。

随着嵌入式技术的普及推广, Sun 公司对 Java 改进后发布了 J2ME (Java 2 Micro Edition), 它定位于嵌入式系统的应用, 专门针对对内存具有苛刻要求的嵌入式系统而设计。它是 Java API 的一个子集, 包含了能够在大多数消费者的设备中运行的 Java 虚拟机、专用的 API 函数库和用于开发和设置的工具。J2ME 在无线通信领域得到了充分的重视和广泛的应用。

可见, Java 语言有着广泛的应用前景。

## 1.2 Java 语言的特点和实现机制

### 1.2.1 Java 语言的特点

Java 编程语言是一种高级语言, 它具有简单、面向对象、分布式、解释型、鲁棒性、安全、结构中立、可移植、高效率、多线程、动态等特点, 下面对其中一些特点作以下解释:

#### 1. Java 是简单的

从某种意义上说, Java 语言是 C 及 C++ 语言的一个变种, 它删除了 C++ 中容易引发程序错误的地方, 如指针、运算符重载; 用接口 (interface) 的简单语言概念替代了 C++ 的多重继承; 通过自动垃圾收集功能大大简化了程序设计者的内存管理工作, 使程序设计变得简单。与 C++ 相比, Java 具有更强的功能和更少的副作用, 因此有人称它为 “C++—”, C++ 程序员可以很快掌握 Java 编程技术。

#### 2. 面向对象

Java 语言面向对象的特性非常彻底, 它同所有面向对象编程语言一样都支持三个概念: 封装性、多态性和继承性。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及动态的接口模型。所有 Java 应用程序和 Applet (Java 小应用程序) 都是对象, 对象中封装了变量和方法, 实现了模块化和信息隐藏; 类是一类对象的原型, 它通过继承机制使子类可以使用父类的方法, 从而实现代码的重用, 用户可以通过这些机制, 建立自己的类库。

#### 3. 与平台无关, 可以移植

与其他编程语言不同, Java 编译器编译生成的是与平台无关的字节码文件格式, 这使得 Java 程序能在网络的任何地方运行。不同种类的计算机, 只要在其处理器和操作系统上有 Java 运行时环境, 字节码文件就可以在该计算机上运行。因此, 使用 Java, 开发者只需编写应用程序的一个版本就能在所有的平台上运行。与平台无关性使得 Java 系统具有很强的可移植性。

#### 4. 解释型语言

为易于实现跨平台性，Java 设计成为解释执行而不是编译执行的运行环境。解释执行在执行过程中，根据所在的不同的硬件平台把字节码解释为目标机的机器语言，实现跨平台运行。

#### 5. 鲁棒性

Java 在编译和运行程序时都要对可能出现的问题进行检查，以防止错误的产生。Java 编译器可以查出许多其他语言运行时才能发现的错误。Java 不支持指针操作，大大减少了错误发生的可能性。Java 具有异常处理的功能，当程序异常时，它能捕获并响应意外情况，以保证程序能稳妥地结束，计算机系统不会崩溃。

#### 6. Java 是安全的

作为网络程序设计语言，Java 执行多层安全机制，以提供足够的安全保障。下载到用户计算机中的字节代码在其被执行前要经过一个核实工具，以确保不含病毒。字节代码被核实后才由 Java 解释器来执行，解释器阻止程序对内存的直接访问和对本地数据的修改。Java 拥有多个阶层的互锁（interlocking）保护措施，能有效防止病毒的入侵和破坏行为的发生。

#### 7. Java 支持多线程

多线程是指程序可以同时执行几个任务，不同的线程处理不同的任务，互不干涉。Java 支持多线程机制，使应用程序可以同时进行不同的操作，处理不同的事件，这样可以容易地实现网络上的实时交互操作。它还提供了易于使用的同步机制，使程序设计简便。

#### 8. 动态、可扩展

Java 允许在类库中加入方法和变量且不影响该类用户的使用。Java 通过接口支持多重继承，比严格的类继承具有更灵活的方式和更大的扩展性。Java 在执行过程中，可以动态地加载各种类库，这使得即使更新类库也不必重新编译使用这一类库的应用程序。Java 的可扩展性使它能够适应不断发展的环境。

### 1.2.2 Java 的实现机制

#### 1. Java 虚拟机

平台是指程序运行的硬件或软件环境。Java 平台是运行于操作系统和硬件之上的纯软件平台。

Java 平台由两部分组成：

- Java 虚拟机（JVM——Java Virtual Machine）
- Java 应用编程接口（Java API——Java Application Programming Interface）

Java 虚拟机是 Java 平台的基础，可以移植到各种基于硬件的平台上。Java 应用编程接口是软件组件的集合，它们提供了很多有用的功能，如图形用户接口（GUI）。Java 应用编程接口被分组为相关组件的库。图 1-1 描述了一个运行在 Java 平台上的 Java 程序，如应用程序

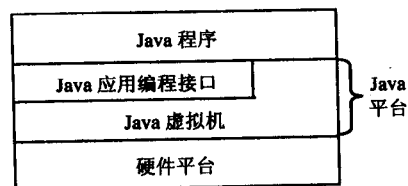


图 1-1 Java 平台

(application) 或小应用程序 (applet)。如图 1-1 所示, Java 应用编程接口和 Java 虚拟机把 Java 程序从硬件依赖中分离出来。

Java 的主要实现机制依靠 Java 虚拟机 (JVM), 它是一种抽象机器, 具有一套虚机器指令, 并有自己的栈、寄存器组等。Java 虚拟机通常由软件实现, 所以可以认为 Java 虚拟机是软件模拟的计算机, 它附着在具体操作系统之上, 是 Java 平台无关性的基础。

编制 Java 程序时, 编程人员先编写好 Java 源文件 (.java 文件), 然后通过 Java 编译器将 Java 源程序编译为一种中间代码——字节码。它是 Java 虚拟机可以识别的“机器码”, 称之为字节码文件 (.class 文件)。在 Java 虚拟机上, 有一个 Java 解释器用来解释 Java 编译器编译后的字节码文件。

只要配备了 Java 虚拟机, 任何处理器 (包括计算机和其他电子设备) 都可以运行字节码文件, 而且不管该字节码是在何种平台上生成的, 如图 1-2 所示。但是, Java 虚拟机的建立需要针对不同的软硬件平台做专门的实现, 既要考虑处理器的型号, 也要考虑操作系统的种类。

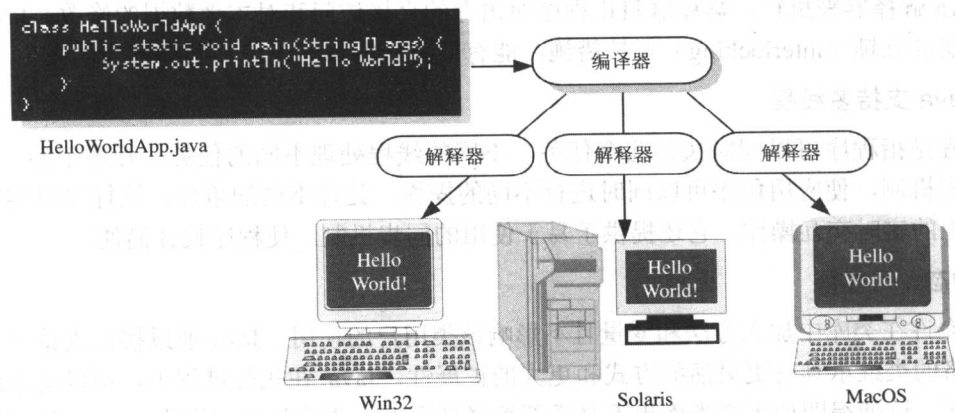


图 1-2 Java 程序与平台无关

2. 垃圾回收机制

Java 程序运行时需要创建类的对象, 这种对象存储在堆内存中, 在执行过程中, 部分对象使用过后就处于废弃状态, 如果不及时进行无用内存的回收, 就会导致内存泄漏。为充分利用资源, Java 运行环境中存在着一个系统级的线程, 专门跟踪内存的使用情况, 定期检测出不再使用的内存, 并进行自动回收, 减轻了程序员的工作量。

3. 代码安全性检测机制

Java 在多个层次上进行安全检测, 编译时, 有语法检查, 执行时, 字节码文件在具体执行时需要经过三个步骤: 首先由类装载器 (class loader) 负责把字节码文件加载到 Java 虚拟机中, 在此过程需要检验该类文件是否符合类文件规范; 其次字节码校验器 (bytecode verifier) 检查该类文件的代码中是否存在着某些非法操作; 如果字节码校验器检验通过, 则由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 程序的编译执行过程如图 1-3 所示。Java 虚拟机采用的是“沙箱”运行模式, 即把 Java 程序的代码和数据都限制在一定内存空间里执行, 不允许程序访问该内存空间外的内存, 如果是 applet 程序, 还不允许访问客户端机器的文件系统。



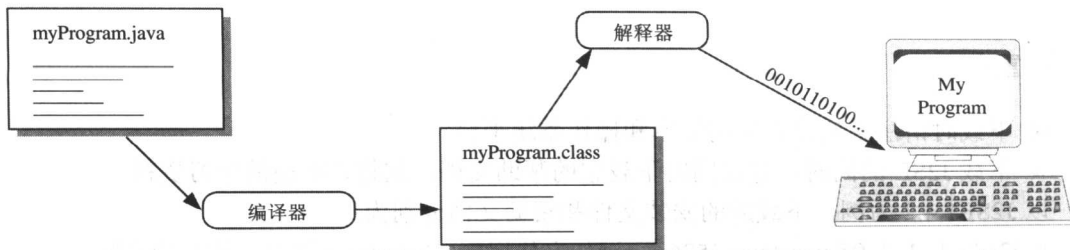


图 1-3 Java 程序的编译执行过程

## 1.3 开发环境设置

Java 不仅仅是一种编程语言，也是一种应用和开发环境，它提供了一整套工具和类库。同时，Java 还是一个程序发布平台，具有两种主要的“发布环境”：一种是 Java 运行时环境（JRE——Java Runtime Environment），它包含了完整的类文件包；另一种是浏览器，许多主要的浏览器都提供了 Java 解释器和运行时环境。

随着 Java 技术的不断发展，Sun 公司根据市场进一步将 Java 细分为：针对企业网应用的 J2EE（Java 2 Enterprise Edition）、针对普通 PC 应用的 J2SE（Java 2 Standard Edition）和针对嵌入式设备及消费类电器的 J2ME（Java 2 Micro Edition）三个版本，如图 1-4 所示。

- Java 2 Platform, Standard Edition (J2SE)
- Java 2 Platform, Enterprise Edition (J2EE)
- Java 2 Platform, Micro Edition (J2ME)

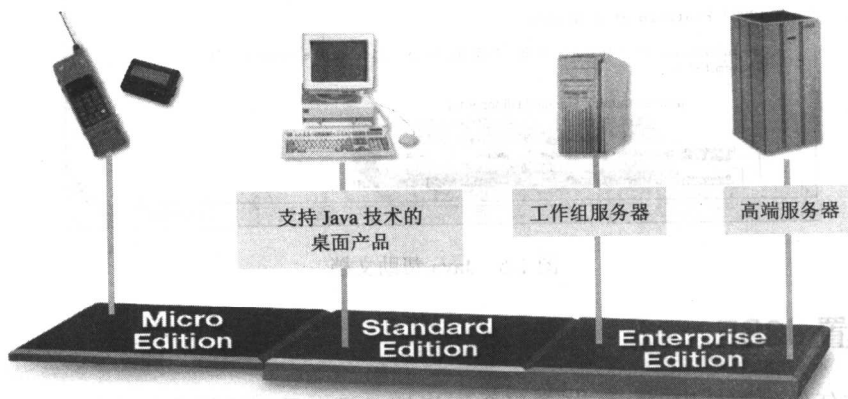


图 1-4 J2EE、J2SE、J2ME

我们在学习 Java 的过程中，主要是采用 J2SE 来进行开发。

### 1.3.1 J2SE 的下载和安装

只要一台普通的 PC 机，安装好一种操作系统，再安装好 JDK，就可以开发 Java 程序了。J2SE 可以在很多光盘上找到，也可以从 <http://java.sun.com/j2se> 网站直接下载。

◀ 下载时请注意选择正确的版本和操作系统平台

◀ 下载 J2SE 的同时，建议同时下载它的帮助文档，这将是有益的学习资料  
以 J2SE1.4.2 为例，下载后的安装文件和帮助文档分别为：

- j2sdk-1\_4\_2\_05-windows-i586-p.exe（安装程序，Windows 平台，简体中文版）
- j2sdk-1\_4\_2-doc.zip（文档）

依版本和平台不同，文件名会有所差异。

安装时可以指定安装目录，该版本的默认安装路径为 `c:\j2sdk1.4.2_05`，我们主要使用的是其 `bin` 子目录下的两个可执行文件：`javac.exe` 和 `java.exe`，前者用于编译 Java 程序，后者用于解释执行 Java 程序。

若安装 Java 帮助文档，可将压缩文件释放到安装目录下，解压为 `docs` 子目录，它以 HTML 文件形式组织，打开 `index.html` 即可查看，如图 1-5 所示。

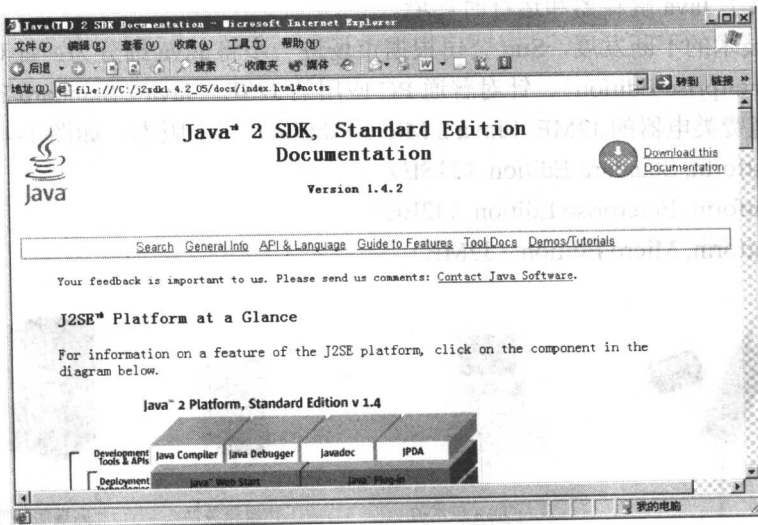


图 1-5 Java 帮助文档

### 1.3.2 设置 J2SE

为了能在任何位置使用 J2SE 的编译和运行等工具，需要设置执行路径和类路径（仍以 J2SE1.4.2 为例）：

在 Windows 95/98/Me 中，可在 Windows 的自动批处理文件 `autoexec.bat` 中添加  
`SET PATH= c:\j2sdk1.4.2_05\bin;%paht%`（注意：“=”两边没有空格）

`SET CLASSPATH= .;c:\j2sdk1.4.2_05\lib;`

然后重新启动计算机。

在 Windows NT 中，选择“控制面板/系统/环境”，单击 `path` 选项，在其下的窗口最前端插入“`c:\j2sdk1.4.2_05;`”（注意：别忘了分号！），然后单击“设置”按钮。

Windows 2000/XP 中，选择“控制面板/系统/高级/环境变量”，如图 1-6 所示。在系统变