



21世纪应用型人才培养规划教材

# Java

## 程序设计与应用开发

於东军 杨静宇 李千目 王国全 编著



清华大学出版社

21 世纪应用型人才培养规划教材

# Java 程序设计与应用开发

於东军 杨静宇 李千目 王国全 编著

清华大学出版社

北 京

## 内 容 简 介

本书作为 Java 程序的入门与应用教材,共分为 3 个部分:第一部分讲解 Java 程序设计的基础知识,包括 Java 基本编程语言、面向对象设计思想、类、对象、接口以及异常处理。第二部分讲解 Java 程序设计的高级知识,包括:GUI 编程、套接口编程、I/O 系统、数据库访问以及多线程编程。第三部分详细分析一个实际项目的开发过程,包括系统分析及功能实现。在项目实例中综合应用第一、二部分的 Java 知识,能够帮助读者进一步巩固与提高。

本书易教易学、学以致用、注重能力,对初学者容易混淆的内容进行了重点提示并配有习题。适合作为普通高等院校应用型本科(含部分专科、高职类)各相关专业的程序设计教材,也适合编程开发人员培训、自学使用。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

Java 程序设计与应用开发/於东军,杨静宇,李千目,王国全编著.—北京:清华大学出版社,2005.3  
(21 世纪应用型人才培养规划教材)  
ISBN 7-302-10565-0

I. J… II. ①於…②杨…③李…④王… III. Java 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 013827 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:章忆文

文稿编辑:刘 颖

封面设计:陈刘源

排版人员:宋 康

印 装 者:北京市清华园胶印厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:20.75 字数:487 千字

版 次:2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号:ISBN 7-302-10565-0/TP·7163

印 数:1~4000

定 价:27.00 元

# 丛书序

21世纪人类已迈入“知识经济”时代，科学技术正发生着深刻的变革，社会对德才兼备高素质应用型人才的需求更加迫切。如何培养出符合时代要求的优秀人才，是全社会尤其是高等院校面临的一项急迫而现实的任务。

为了培养高素质应用型人才，必须建立高水平的教学计划和课程体系。在教育部有关精神的指导下，我们组织全国高校计算机专业的专家教授组成《21世纪应用型人才培养规划教材》编审委员会，全面研讨计算机和信息技术专业的应用型人才培养方案，并结合我国当前的实际情况，编审了这套《21世纪应用型人才培养规划教材》丛书。

## 编写目的

配合教育部提出要有相当部分高校致力于培养应用型人才的要求，以及市场对应用型人才需求量的不断增加，本套丛书以“理论与能力并重，应用与应试兼顾”为原则，注重理论的严谨性、完整性，案例丰富、实用性强。我们努力建设一套全新的、有实用价值的应用型人才培养系列教材，并希望能够通过这套教材的出版和使用，促进应用型人才培养的发展，为我国建立新的人才培养模式作贡献。

## 首推书目

本丛书首批推出如下书目：

- Visual Basic 程序设计与应用开发
- Visual FoxPro 程序设计与应用开发
- Java 程序设计与应用开发
- Visual C++程序设计与应用开发
- Delphi 程序设计与应用开发
- C 语言程序设计与应用开发
- 计算机应用基础(等级考试版)
- 计算机网络技术
- 微机原理与接口技术
- 软件技术基础——数据结构与算法·程序设计·软件工程·数据库

## 丛书特色

➤ 理论严谨，知识完整。本丛书内容详实、系统性强，对基本理论进行了全面、准确的剖析，便于读者形成完备的知识体系。

➤ 入门快速，易教易学。突出“上手快、易教学”之特点，用任务来驱动，以教与学的实际需要取材谋篇。

➤ 学以致用，注重能力。将实际开发经验融入基本理论之中，力求使读者在掌握基本理论的同时，获得实际开发的基本思想方法，并得到一定程度的项目开发实训，以培养学生独立开发较为复杂的系统的能力。

➤ 示例丰富，实用性强。以实际开发案例以及部分考试真题为示例，兼顾应用与应试。

➤ 深入浅出，螺旋上升。内容和示例的安排难点分散、前后连贯，并采用循序渐进、层次清晰、步骤详细的编写风格，便于学生理解和实现。

➤ 提供教案，保障教学。本丛书绝大部分教材提供电子教案，便于老师教学使用，并提供源代码下载，便于学生上机调试。

### ▣ 读者定位

本系列教材主要面向普通高等院校和高等职业技术学院，适合本科和高职高专教学需要；同时也非常适合编程开发人员培训、自学使用。

### ▣ 关于作者

丛书编委特聘请执教多年、且有较高学术造诣和实践经验丰富的名师参与各册之编写。他们长期从事有关的教学和开发研究工作，积累了丰富的经验，对相应课程有较深的体会与独到的见解，本丛书凝聚了他们多年的教学经验和心血。

### ▣ 互动交流

本丛书贯穿了清华大学出版社一贯严谨、科学的图书风格，但由于我国计算机应用技术教育正在蓬勃发展，要编写出满足新形势下教学需求的教材，还需要我们不断的努力实践。因此，我们非常欢迎全国更多的高校老师积极加入到《21 世纪应用型人才培养规划教材》编审委员会中来，推荐并参与编写有特色、有创新的应用型教材。同时，我们真诚希望使用本丛书的教师、学生和读者朋友提出宝贵意见或建议，使之更臻成熟。联系信箱：  
Book21Press@126.com。

《21 世纪应用型人才培养规划教材》编审委员会

E-mail: Book21Press@126.com

# 《21 世纪应用型人才培养规划教材》 编审委员会

主 编：吴文虎 许卓群 王 珊 李 波

总 策 划：清华大学出版社第三事业部

执行策划：何光明

编 委：(以姓氏笔画为序)

毛红梅	王景玉	尹 静	史国川	史春联
许 勇	刘家琪	吴小俊	陈亦愿	杨 明
李勇智	李烈敏	杨静宇	罗 坚	周 松
钱 隆	聂承启	戴仕明	檀 明	魏建香

# 前 言

为适应当前普通高等院校注重培养应用型人才的需求，清华大学出版社推出了《21世纪应用型人才培养规划教材》丛书。本书作为其中之一，依据普通高等院校教学大纲以及实际开发经验组织内容，注重理论的严谨性和完整性。案例丰富、实用性强，力求使学生在掌握 Java 编程语言的同时获得程序开发的基本思想并得到一定程度的项目开发实训，以培养学生独立开发较为复杂系统的能力。

本书共分为三个部分：第一部分讲解 Java 程序设计的基础知识，包括 Java 基本编程语言、面向对象设计思想、类、对象、接口以及异常处理。第二部分讲解 Java 程序设计的高级知识，包括：图形界面编程、套接口编程、输入输出系统、数据库访问以及多线程编程。第三部分详细分析一个实际项目的开发过程，包括系统分析及功能实现。在项目实例中综合应用第一、二部分的 Java 知识，能够帮助读者进一步巩固与提高。

我们认为，Java API 文档始终是学习 Java 最权威、最全面的资料。学会使用 Java API 文档对于日后的实际开发工作意义重大。为此，本书一方面讲解了 Java API 文档的基本使用方法，另一方面在课后习题中提供给读者大量练习使用的机会。

特别感谢中兴通讯股份有限公司的资深高级程序员王国全提供了宝贵的修改意见，并编写了第 7、8 章。李千目编写了第 3、4、7、8 章的习题、习题答案及 7.1 小节，并和王国全合作编写了第 3、4 章。其他章节由东军和杨静宇编写。

本书的出版得到了南京理工大学科研发展基金的资助，作者所在的 603 教研室亦为写作提供了宽松的环境。在成书过程中，还得到了何光明老师的大力支持。张代华、候君、吴丽芬、郑宇杰、毛红梅、钱隆、檀明、李文龙、史春联、周松、赵明等同志对本书亦作出了重要贡献，他们在资料收集、文字校对以及程序测试过程中付出了辛勤劳动，在此一并表示感谢。

本书中完整的源代码可以从 <http://www.wenyuan.com.cn> 下载得到。限于作者水平，书中难免存在不当之处，恳请广大读者批评指正。任何批评和建议请发至：[Book21Press@126.com](mailto:Book21Press@126.com)。

作者

2005 年 2 月于南京

# 目 录

第 1 章 快速入门 .....	1	2.3 字符串 .....	34
1.1 Java 简介 .....	1	2.3.1 字符串连接 .....	34
1.1.1 发展简史 .....	1	2.3.2 修改字符串 .....	35
1.1.2 Java 的特点 .....	2	2.3.3 判断字符串是否相等 .....	36
1.2 Java 开发环境的构建 .....	4	2.3.4 使用帮助文档 .....	36
1.2.1 Java 开发包的安装 .....	4	2.4 控制流 .....	39
1.2.2 环境变量设定 .....	6	2.4.1 if 条件语句 .....	39
1.2.3 基本 DOS 命令 .....	7	2.4.2 for 循环语句 .....	40
1.3 Java 应用程序 .....	9	2.4.3 while/do while 循环语句 .....	41
1.3.1 编写源代码 .....	9	2.4.4 switch 分支选择语句 .....	42
1.3.2 编译源代码 .....	10	2.4.5 break、continue .....	44
1.3.3 运行程序 .....	11	2.5 数组 .....	45
1.3.4 常见问题解答 .....	11	2.5.1 一维数组 .....	45
1.3.5 理解例子 .....	12	2.5.2 数组拷贝 .....	48
1.4 Java Applet .....	15	2.5.3 多维数组 .....	48
1.4.1 编写 Applet 源代码 .....	15	习题 .....	50
1.4.2 编写嵌入 Applet 的 HTML 文件 .....	16	第 3 章 类与继承 .....	52
1.4.3 运行 Applet .....	16	3.1 类和对象 .....	52
习题 .....	17	3.1.1 类声明 .....	52
第 2 章 Java 基本语法 .....	19	3.1.2 类成员 .....	53
2.1 变量 .....	19	3.1.3 关键字 this .....	58
2.1.1 基本数据类型 .....	19	3.1.4 方法重载 .....	59
2.1.2 变量 .....	22	3.1.5 类继承 .....	60
2.1.3 变量作用域 .....	23	3.1.6 类的初始化过程 .....	61
2.1.4 变量初始化 .....	25	3.1.7 源文件 .....	62
2.2 操作符 .....	25	3.2 面向对象特性 .....	63
2.2.1 算术操作符 .....	26	3.2.1 封装性 .....	63
2.2.2 关系与条件操作符 .....	28	3.2.2 继承性 .....	65
2.2.3 位操作符 .....	29	3.2.3 多态性 .....	69
2.2.4 其他类型操作符 .....	29	3.3 关键字 static .....	70
2.2.5 数字类型转换 .....	31	3.4 关键字 final .....	72
2.2.6 操作符优先级 .....	32	3.4.1 final 变量 .....	72
2.2.7 表达式、语句和块 .....	33	3.4.2 final 方法 .....	72
		3.4.3 final 类 .....	72



3.5 对象复制 .....	73	<b>第 6 章 GUI 编程</b> .....	108
3.6 内部类 .....	75	6.1 Swing 起步 .....	108
习题 .....	77	6.1.1 Swing 概述 .....	108
<b>第 4 章 接口、抽象类与包</b> .....	80	6.1.2 一个 GUI 实例 .....	109
4.1 接口 .....	80	6.1.3 面板 .....	110
4.1.1 接口定义 .....	80	6.1.4 改变应用程序的观感 .....	113
4.1.2 接口的实现 .....	81	6.2 AWT 事件处理 .....	114
4.1.3 接口作为类型 .....	83	6.2.1 事件处理简介 .....	114
4.1.4 接口不应改变 .....	84	6.2.2 事件处理实例 .....	115
4.2 抽象类 .....	84	6.2.3 使用事件适配器 .....	118
4.2.1 抽象方法 .....	84	6.2.4 AWT 事件继承关系 .....	120
4.2.2 抽象类 .....	85	6.3 布局管理 .....	121
4.2.3 抽象类和接口的比较 .....	86	6.3.1 流式布局 .....	121
4.3 包 .....	88	6.3.2 边框布局 .....	123
4.3.1 包的作用 .....	88	6.3.3 网格布局 .....	124
4.3.2 包命名 .....	89	6.3.4 网格袋布局 .....	125
4.3.3 包的使用 .....	90	6.4 常用 GUI 组件 .....	130
习题 .....	92	6.4.1 标签 .....	130
<b>第 5 章 异常处理</b> .....	94	6.4.2 文本输入类组件 .....	132
5.1 概述 .....	94	6.4.3 选择类组件 .....	139
5.2 异常处理 .....	94	6.4.4 菜单类组件 .....	145
5.2.1 遭遇异常 .....	94	6.4.5 对话框 .....	151
5.2.2 捕获异常 .....	96	6.4.6 表格 .....	158
5.2.3 声明方法抛出异常 .....	99	习题 .....	163
5.3 异常的抛出策略 .....	101	<b>第 7 章 网络编程</b> .....	165
5.3.1 不要声明抛出所有异常 .....	102	7.1 网络编程的基本概念 .....	165
5.3.2 异常声明的数量 .....	103	7.1.1 网络基础 .....	165
5.3.3 保持 throws 子句的 稳定性 .....	103	7.1.2 TCP/IP 协议簇 .....	165
5.3.4 异常抛出策略小结 .....	104	7.1.3 TCP 与 UDP .....	166
5.4 异常的捕获策略 .....	104	7.2 InetAddress 类 .....	168
5.4.1 混淆的异常 .....	105	7.2.1 创建 InetAddress 对象 .....	168
5.4.2 代码重构的困难 .....	105	7.2.2 类 InetAddress 的应用 .....	169
5.4.3 捕获超级异常的合理 情形 .....	106	7.3 TCP 程序设计 .....	171
5.4.4 异常捕获策略小结 .....	107	7.3.1 Java 客户端套接字 .....	171
习题 .....	107	7.3.2 Java 服务器套接字 .....	175
		7.4 UDP 程序设计 .....	179
		7.4.1 DatagramPacket 类 .....	180
		7.4.2 DatagramSocket 类 .....	181

7.5 URL 程序设计 .....	186	习题 .....	240
7.5.1 URL 类 .....	186	<b>第 10 章 多线程</b> .....	242
7.5.2 URLConnetction 类 .....	189	10.1 什么是线程 .....	242
习题 .....	191	10.2 一个简单的例子 .....	242
<b>第 8 章 Java I/O 系统</b> .....	193	10.3 定制线程类 .....	244
8.1 文件类 .....	193	10.3.1 继承线程类 .....	244
8.1.1 文件类 File .....	193	10.3.2 实现 Runnable 接口 .....	244
8.1.2 文件过滤 .....	197	10.4 线程的生命周期 .....	245
8.2 Java I/O 结构 .....	198	10.4.1 创建线程 .....	246
8.2.1 字节流 .....	198	10.4.2 启动线程 .....	246
8.2.2 字符流 .....	200	10.4.3 线程运行 .....	246
8.3 使用流 .....	201	10.4.4 线程阻塞 .....	246
8.3.1 标准流 .....	201	10.4.5 终止线程 .....	247
8.3.2 文件流 .....	203	10.5 线程中断 .....	248
8.3.3 过滤器流 .....	206	10.6 线程的优先级 .....	249
8.3.4 随机访问文件 .....	210	10.6.1 利己线程 .....	250
8.3.5 读取器和写出器 .....	211	10.6.2 分时方式 .....	250
8.4 对象序列化 .....	213	10.7 线程同步 .....	251
习题 .....	215	10.7.1 对象锁 .....	253
<b>第 9 章 访问数据库</b> .....	216	10.7.2 可重入锁 .....	254
9.1 JDBC 简介 .....	216	10.7.3 notifyAll 和 wait 方法 .....	254
9.2 建立数据库连接 .....	217	习题 .....	256
9.2.1 使用 JDBC/ODBC 桥接 驱动程序 .....	218	<b>第 11 章 项目实践</b> .....	259
9.2.2 使用 JDBC 驱动程序 .....	219	11.1 系统简介 .....	259
9.2.3 使用配置文件 .....	221	11.2 功能设计 .....	259
9.3 执行 SQL 语句 .....	222	11.2.1 计量模拟程序 .....	259
9.3.1 executeUpdate .....	222	11.2.2 结算收费系统 .....	260
9.3.2 executeQuery .....	222	11.3 数据库设计 .....	261
9.3.3 executeBatch .....	223	11.4 代码实现 .....	263
9.4 使用 Prepared Statement .....	225	11.4.1 计量模拟 .....	263
9.5 事务处理 .....	226	11.4.2 快闪屏 .....	267
9.6 编写数据库工具类 .....	228	11.4.3 系统设置对话框 .....	269
9.7 一个实例 .....	232	11.4.4 基础资料管理 .....	272
9.7.1 数据库 .....	232	11.4.5 出账、审核及收费 .....	282
9.7.2 布局及功能简介 .....	232	11.4.6 运行系统 .....	283
9.7.3 源代码 .....	233	习题 .....	284

---

<b>附录 A 参数传递</b> .....	285	A.3.2 参数为对象数组 .....	289
A.1 传值还是传引用.....	285	<b>附录 B 各章习题参考答案</b> .....	292
A.2 基本数据类型的参数传递.....	285	<b>参考文献</b> .....	293
A.3 对象数据类型的参数传递.....	286		
A.3.1 参数为单个对象 .....	286		

# 第 1 章 快速入门

本章介绍 Java 的发展简史及特点，引导读者构建 Java 基本开发环境，学会编写、编译以及运行简单的 Java 应用程序和 Java Applet，旨在让未曾接触过 Java 编程语言的读者快速入门。

## 1.1 Java 简介

Java 由于其与生俱来的诸多优点，目前已经在各行各业得到了广泛应用。到处都在讨论 Java，但是 Java 究竟是什么呢？概括说来，和一般编程语言不同之处在于：Java 不仅是一种面向对象的高级编程语言，它还是一个平台(Platform)；应用 Java 更易于开发出高效、安全、稳定以及跨平台的应用程序。目前 Java 还处于快速发展阶段，新的特性和应用仍在不断涌现。本小节对 Java 的发展历史以及特点进行简要介绍。

### 1.1.1 发展简史

随着 Java 技术的飞速发展，越来越多的人加入到学习 Java 编程语言的热潮中。虽然现在 Java 无处不在，但是其发展历史并不是很长。20 世纪 90 年代初，Sun 公司为了适应消费类家电项目的需求而设计了一种小型的计算机语言，要求占用内存小、适应多种处理芯片。为此，设计人员设计出了一种面向对象的“可移植”的语言。在执行前，生成一个“中间代码”，在任何一种机器上只要安装了特定的解释器，就可以运行这个“中间代码”。这样的“中间代码”非常小，解释器也不大，这就是 Java 的雏形。遗憾的是，当时这门语言并未能被 Sun 公司和消费类家电公司所接受。Sun 公司一度也曾探求 Java 在其他方面的应用，却均以失败告终。

1995 年，随着 Internet 网络的兴起，人们迫切需要一个好的浏览器以方便阅读网上的 Web 页面，Sun 公司为此使用 Java 语言及时开发出了一个浏览器——HotJava，并获得了极大的成功。HotJava 中首次引入了传程序的思想，即 HotJava，可以将服务器上的程序(applet，称为小应用程序)下载到浏览器中执行，这就为原本只能静态显示的页面增添了活力。HotJava 在 1995 年 SunWorld 大会上的出现就引发了世界范围内的 Java 热，至今未衰。

早期版本 Java 1.0 其实并不适合应用程序的开发，它甚至不支持打印功能。直到 1998 年 Java 1.2 版本的出现，Java 才真正意义上成为现代开发工具中的利器。

Java 现在的体系结构已经变得相当庞大，从大的框架上可以分为三大块：J2SE(Java 2 Platform Standard Edition)、J2EE(Java 2 Platform Enterprise Edition) 和 J2ME(Java 2 Platform Micro Edition)。J2EE 主要用于开发服务端应用程序；J2ME 则应用于嵌入式设备，如移动电话；J2SE 是标准的开发工具包，其中包含了基本 Java 的核心应用编程接口(API，Application Programming Interface)。实际上 J2EE 和 J2ME 中都可能使用到 J2SE 中的

API。本书将重点放在 J2SE 上，主要帮助读者掌握必要的 Java 基础知识以及一定的项目开发经验。此外要说明的是，本书主要关注如何使用 Java 开发应用程序，小应用程序 (Applet) 的内容未被更多涉及。

就 J2SE 来讲，也已经提供了大量的 API 供开发者直接使用，其中的类和接口大约有 3000 多个，提供的方法和变量更是数以万计。全部由本书来讲解，显然是不合适的。为此，本书中将重点讲解那些基础的且必需掌握的部分，未曾涉及到的部分希望读者在日后的实际开发过程中，通过查阅 Java API 文档逐步掌握。

## 1.1.2 Java 的特点

学过 C/C++ 语言的都知道，编写完 C/C++ 源代码后需要编译成机器码才能在机器上运行，因此 C/C++ 是编译型的语言。还有一些语言完全是解释型的 (如 Basic)，不需要任何编译工作，是边解释边执行的。与这些语言不同的是，Java 既是编译型的又是解释型的。我们在编写完 Java 源代码后，首先需要将其编译为一种中间状态的字节码 (bytecode)，然后再由 Java 虚拟机 (Java VM) 上的解释器来解释执行。实际上我们可以将字节码看作是虚拟机的机器码。这样，任何一个系统，只要上面具备了 Java 虚拟机，那么它总是可以运行编译好的字节码。正是这种将源代码编译到中间字节码的机制，使得 Java 能够实现“一次编程，到处运行”的目标。

Java 的特点可以概括为：简单、面向对象、分布式、解释型语言、健壮性、安全性、结构中立、可移植性、高性能、多线程和动态。

这里我们选择性地讲几个初学者感兴趣的特点。更多具体内容可以参考 Sun 公司关于 Java 的白皮书 (<http://java.sun.com/docs/white/langenv/>)。

### (1) 简单

对于有过 C++ 编程经验的读者来说，学习 Java 语言不会存在太多困难。因为 Java 语法要比 C++ 的语法简单得多。Java 剔除了 C++ 中那些复杂而且不常使用的语法特性，例如操作符重载、多重继承等等。熟悉 C++ 的读者在学习 Java 时，很多时候需要做的就是设法“忘记” C++ 中的一些语法规则。

但是对于没有编程经验的读者来说，就会发现 Java 还是具有相当难度的。一方面 Java 有一些奇怪的语法 (如匿名内部类)，另一方面需要自己编写大量的代码 (习惯“拖”控件编程的就不适应了)。这也是为什么本书后面建议初学者使用文本编辑工具自己手工编写代码的原因。通过这种方式，读者可以尽快熟悉 Java 语法并强化记忆。

### (2) 面向对象

面向对象是现代编程语言的重要特性之一。历史的经验已经表明，面向对象技术极大地提高了人们的软件开发能力。现在很难想像还使用纯粹的面向过程的语言去开发大型、复杂的项目。Java 语言是一种纯粹的面向对象的语言，在面向对象一些问题的处理上要优于 C++ (如多重继承)。习惯于传统面向过程的读者在刚理解面向对象的概念时，会存在一定的困难。但是考虑到面向对象的优越性，在这方面花点精力是值得的。

### (3) 健壮性

每个程序设计人员都希望自己编写的程序更加可靠。在编写程序时，考虑周到一点固然能降低出错的可能性，但是如果语言本身就能提供一系列的机制防患于未然，必能进一

步提高程序的健壮性。Java 语言的设计目标之一，就是帮助程序员编写出高可靠性的程序，为此，Java 语言提供了很多技术用以提高程序的可靠性，如数组越界检查、运行时类型检查、取消指针操作以及无用单元自动回收等等。

学过 C/C++ 的都知道，指针具有强大的功能和灵活性。也正是这个原因，指针成为程序不稳定的最大隐患之一。由于 C/C++ 存在指针运算(如 `p++`、`p` 为指针)，指针可以在内存中自由移动，因而可能修改其他程序所使用的内存或是系统内存中的内容，从而造成系统的崩溃。而有些语言，例如 Basic，没有显式的指针，这又会使得在实现某些功能时变得困难(如实现自己定制的数据结构)。Java 语言同时考虑到这两方面的问题，提出了一个很好的解决方法：可以有指针，但是取消了指针的运算。这样上面的两个问题便迎刃而解了。

✓Java 中并不是没有指针了，Java 中的引用其实就是指针，只是取消了指针运算。取消指针运算防止了内存泄漏的可能性。

#### (4) 可移植性

Java 作为一种高级编程语言，最让人津津乐道的优点就是所谓的跨平台特性了，也就是在不同的操作系统上源代码不做修改就能得到相同的运行结果。

先来看一下为什么会存在可移植性的问题。我们都知道，同一数据类型在不同平台上大小是不一样的。例如有的平台上整型(int)用 16 位二进制数来表示，而有的平台上却用 32 位二进制数来表示。这样一个在用 32 位二进制数来表示整数的平台上开发的程序拿到用 16 位二进制数表示整数的平台上去运行，很可能就会产生溢出问题。Java 中使用固定大小的数据类型解决了这个问题，例如整型始终用 32 位来表示，和底层平台的转换由 Java 虚拟机来完成。

可移植性问题不仅仅存在于数据类型大小不一致，其他方面如图形界面的显示、多线程等都存在可移植性问题。确实，Java 在可移植性上已经取得了极大的成功，但还不够完善。然而，这并不能掩盖 Java 在这方面所做的贡献以及其他诸多的优点。

#### (5) 多线程

具有多线程处理能力可以使得应用程序能够具有更好的交互性、实时性。Java 在多线程处理方面性能超群，同时也非常简单。有过 C++ 多线程编程经验的读者，一定会对 Java 的多线程编程之简单感到惊叹。Java 中多线程的实现是由操作系统或是线程库来完成的，编程人员在不同平台上使用多线程的 Java 代码是完全相同的。正是由于 Java 中使用多线程的简单性，使得它成为服务器端应用程序开发的利器。

从另一个角度来讲，Java 不仅仅是一种编程语言，它还是一个平台(Platform)。所谓平台是指应用程序运行的软硬件环境，通常是操作系统和硬件的总称。例如 Windows 2000、Linux、Solaris 等等都是不同的平台。Java 平台是运行在这些平台之上的纯粹的软平台，由两大部分组成：Java 虚拟机和 Java 应用程序接口(Java API)。图 1.1.1 显示了 Java 平台，从图中我们可以看出，Java 应用程序是和底层的操作系统(基于硬件的平台)相隔离的，它们之间是通过 Java 平台来进行通信的。

在这里，我们要给大家提一下本地代码(native code)，所谓本地代码是指已经编译好的、可以在特定硬件平台上直接运行的代码，如已经编译好的 C 代码。因此，运行 Java 程序要比本地代码慢一点。然而在现代的网络时代，程序运行速度的瓶颈主要存在于网络

的速度；另一方面，已经出现了很多技术(JIT 等)用以提高 Java 程序的运行速度，所以在编写大部分应用程序时这一点是不用担心的。

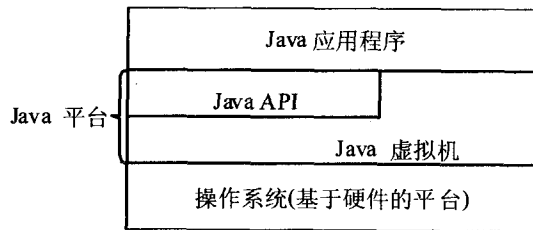


图 1.1.1 Java 平台

## 1.2 Java 开发环境的构建

Java 开发环境的基本要求非常低，只需一个 Java 开发包，再加上一个纯文本编辑器即可。为了提高开发效率，可以使用功能强大的文本编辑工具，例如 TextPad、UltraEdit 等等。对于熟练的开发人员，为了进一步提高开发效率，还可以使用具有可视化功能的 Java 专用开发工具，例如 JBuilder。但是作者不建议初学者使用这类高级专用开发工具，而是建议使用文本编辑工具。这样有助于初学者加强对必要的关键字、常用系统类以及 Java 语法的记忆与理解。

UltraEdit 是一款非常优秀的文本编辑工具，并且能够识别很多编程语言(例如 Java、C 以及 C++等等)的关键字，不同的关键字还可以以不同的颜色醒目地标识出来，非常方便编程人员使用。本文所有的代码均是使用 UltraEdit 编写完成的。使用 UltraEdit 并不困难，读者在获得 UltraEdit 的安装包并正确安装后，即可直接使用。

### 1.2.1 Java 开发包的安装

Sun 公司免费提供了 Java 开发工具包(JDK, Java Development Kit)。该工具包包含了编译、运行以及调试程序所需要的工具，此外还提供了大量的基础类库。基础类库是应用开发中的砖瓦，开发人员灵活地使用就可以建造出各种各样的建筑物——应用软件。

Sun 公司为不同的操作系统(如 Windows、Unix/Linux、Mac OS)提供了相应的 Java 开发包安装程序。本书中使用 Windows 操作系统环境下的 Java 开发包。读者可以登录 Sun 公司的网站(<http://www.sun.com>)获取免费的 Java 开发包安装程序。本书中所给的例子程序均在版本为 1.4.2.04 的 JDK 下运行通过，所使用的操作系统为 Windows 2000 中文专业版。

在得到 Java 开发包后首先需要进行安装。双击 Java 开发包安装程序，出现安装界面，如图 1.2.1 所示。

随后，安装程序会弹出如图 1.2.2 所示的许可证协议阅读画面，选中接受条款后，单击【下一步】按钮继续安装。

安装类型选择：用户可以以两种安装类型进行开发包的安装，典型和自定义方式。当选择【典型】方式安装时，安装程序将开发包内的所有内容安装到系统默认的安装目录

下；如果想选择性地安装开发包的内容或是更改开发包的安装路径，则选择【自定义】方式安装，然后单击【下一步】按钮。这里以自定义方式为例，如图 1.2.3 所示。

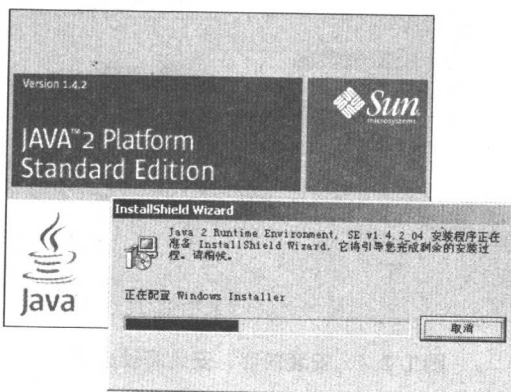


图 1.2.1 安装启动界面

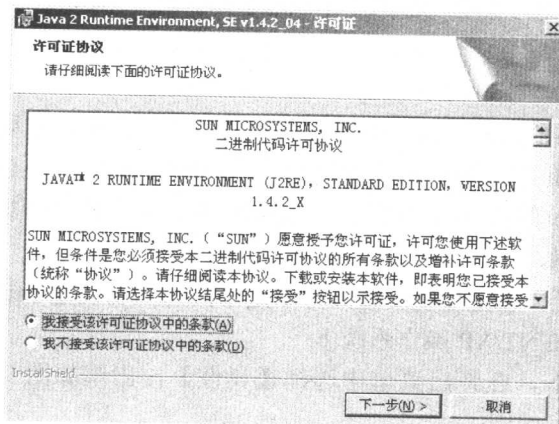


图 1.2.2 许可证协议阅读画面

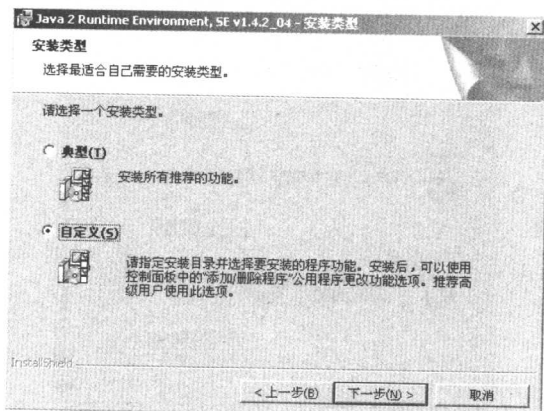


图 1.2.3 安装类型选择

安装内容、安装路径选择：用户可以选择安装开发包的部分或是全部内容，此外还可以选择安装的路径，如图 1.2.4 所示。



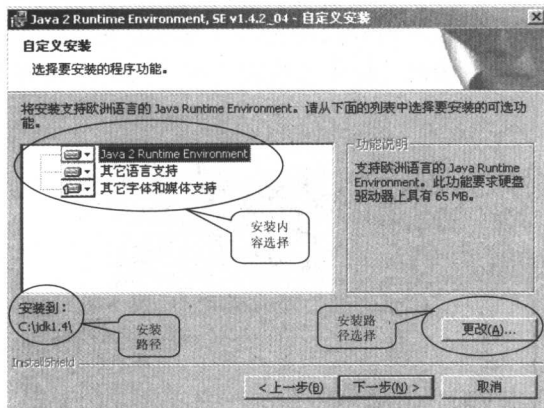


图 1.2.4 安装内容、安装路径选择

在选定了安装内容和安装路径后，单击【下一步】按钮进入正式安装。其他版本的开发包安装过程中除了显示画面略有不同，其他方面类似。

## 1.2.2 环境变量设定

设定环境变量的目的是为了能够正常使用所安装的开发包。主要包括两个环境变量：Path 和 Classpath。Path 称之为路径环境变量，用来指定 Java 开发包中的一些可执行程序(如 java.exe、javac.exe 等)所在的位置；Classpath 称之为类路径环境变量，其意义在后面的第 4 章中我们会提到。

不同的操作系统上，设定环境变量的方法是不同的。以设定 Path 为例：

### (1) Windows 2000/NT/XP 操作系统下

右击【我的电脑】，在弹出菜单中选择【属性】，出现如图 1.2.5 所示的【系统特性】对话框。

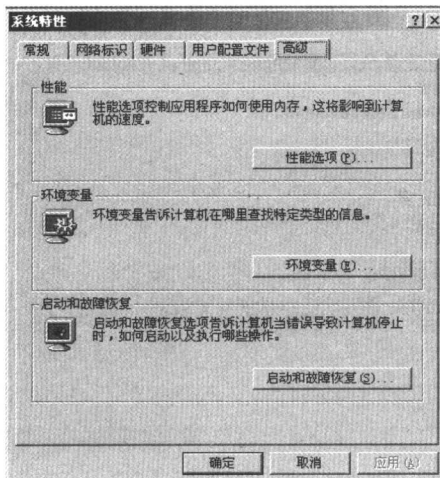


图 1.2.5 【系统特性】对话框

选择【高级】属性页，在该属性页面中有一个【环境变量】按钮，单击该按钮进入