

中等职业学校计算机系列规划教材

Java 语言案例教程

陈海宁 索 旺 编著

北京工业大学出版社

内 容 提 要

本书作为中等职业学校相关专业的教材，循序渐进地介绍了 Java 2 的基础编程知识，包括 Java 语言的基本语法，Java 语言的类和对象，代码中的异常捕捉处理，数组和字符串的应用，标准输入输出及目录文件的管理，图形用户界面的事件编程，多媒体、多线程编程以及网络编程等。本书使用通俗而简洁的语言对各个实例进行描述，图文并茂，有利于学生在比较少的课时内认识并掌握 Java 语言编程技术。

本书也适合 Java 语言初学者进行自学，可供对 Java 语言有一定了解的读者作为提高技术水平之用。

图书在版编目 (CIP) 数据

Java 语言案例教程/陈海宁，索旺编著. —北京：北京
工业大学出版社，2007.6
ISBN 978-7-5639-1788-4

I. J… II. ①陈… ②索… III. JAVA 语言—程序设计—
专业学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 069717 号

Java 语言案例教程

陈海宁 索 旺 编著

*

北京工业大学出版社出版发行

邮编：100022 电话：(010) 67392308

各地新华书店经销

徐水宏远印刷厂印刷

*

2007 年 6 月第 1 版 2007 年 6 月第 1 次印刷

787mm×1 092mm 16 开本 14 印张 337 千字

印数：1~4 000 册

ISBN 978-7-5639-1788-4/T · 303

定价：19.50 元

前 言

为了贯彻《中共中央国务院关于深化教育改革推进素质教育的决定》的精神，落实《面向 21 世纪教育振兴行动计划》所提出的职业教育课程改革和教材建设规划，编者本着素质教育的思想，依据教育部 2001 年颁布的《中等职业学校计算机及应用专业教学基本要求》，从社会发展对中、初级计算机人才的实际需求出发，编写了这套中等职业学校计算机专业的教材。

Java 编程语言是由美国 SUN 公司研发的第一种在因特网上使用的、具有“硬件/软件中立性”的编程语言。Java 语言是功能强大且最具吸引力的编程语言，它所具有的纯面向对象、简单高效、与平台无关、安全、完整、支持多线程等特点，成为网络上的世界语言。Java 语言是 20 世纪 90 年代最重要的技术发明。

全书共分 10 章，按照由浅入深的方式进行安排。第 1 章到第 4 章主要介绍 Java 语言的基础知识和面向对象的基本概念，以及如何配置相关的编程环境和必要的工具，从而使读者对 Java 语言编程有一个基本的认识，为后面的 Java 语言开发实例讲解做好铺垫。

从第 5 章到第 10 章，分专题分别讲述了流、多线程、Swing、多媒体、网络和数据库编程。在讲述实例的过程中，重点向读者介绍 Java 语言编程思想。读者可以一边学习实例，一边在本书的指导下培养自己面向对象软件开发的思维。在每章的最后，编者对本章所讲述的重点做了小结。由于 Java 语言的应用十分广泛，考虑到读者的实际需要，因此本书通过这些实用的小例子，讲述了 Java 语言在各个领域的应用，而更重要的是向读者阐述了面向对象的编程思想。

全书内容覆盖 Java 语言的各个要点，知识面广泛，注重条理性，编程步骤清晰、完善，而且易于操作。本书对基础概念的讲解比较全面，在内容安排上由浅入深，既照顾到了初学者，也为欲了解面向对象程序开发基础知识的读者提供了有益的参考。建议读者在学习过程中适当配合上机实践，这样会达到更好的学习效果。

本教材是作者多年从事教学实践工作的心得之作。教材的结构体系为中等职业教育教材改革进行了新的尝试。教材根据中等职业教育的特点，结合了中等职业教育的实际情况，通过 Java 案例引入相应的教学内容，由表到里，由浅入深，

既便于教学，又便于学生自己阅读学习，能很好地调动学生的积极主动性，提高教学效果。教材在内容的安排上层次清晰，结构严谨，便于理解，着重实际应用。

为了方便教师教学，我们免费为使用本套教材的师生提供电子教学参考资料包：

- ◆ PowerPoint 多媒体课件
- ◆ 习题参考答案
- ◆ 教材中的程序源代码
- ◆ 教材中涉及的实例制作的各类素材

有需要的教师可以登录教学支持网站免费下载。在教材使用中有什么意见或建议也可以直接和我们联系，电子邮件地址：scqcwh@163.com。

本书所选实例内容翔实、结构紧凑、条理清晰、覆盖知识点全面。但由于编写时间较为仓促，书中难免会有疏漏和不足之处，恳请广大读者提出宝贵意见。

编者

目 录

第 1 章 Java 概述	1
1.1 Java 语言的发展史	1
1.2 Java 技术的特点	3
1.3 Java 语言与 C、C++ 语言	5
1.4 Java 与 .NET 技术	6
1.5 J2ME	7
1.6 J2EE	8
1.7 网上资源	10
【本章小结】	12
【习题】	12
第 2 章 第一个 Java 程序	14
2.1 JDK 的安装	14
2.1.1 下载 JDK	14
2.1.2 安装 JDK	15
2.1.3 设置 Java 的环境变量	16
2.2 Java 程序开发流程	18
2.2.1 应用程序	18
2.2.2 小程序	21
2.2.3 小程序的生命周期	23
2.3 常用的集成开发环境	24
【本章小结】	26
【习题】	27
第 3 章 Java 语言基础知识	28
3.1 简单数据类型	28
3.1.1 标志符和保留字	28
3.1.2 数据类型	29
3.1.3 简单数据类型	29
3.1.4 简单数据类型中各类型数据间的优先关系和相互转换	31
3.2 运算符和表达式	31
3.2.1 运算符	31
3.2.2 表达式	33
3.3 控制语句	34

3.3.1	分支语句	34
3.3.2	循环语句	35
3.3.3	跳转语句	36
3.4	数组	37
3.4.1	一维数组	37
3.4.2	多维数组	38
3.5	字符串的处理	40
3.5.1	字符串的表示	40
3.5.2	访问字符串	41
3.5.3	修改字符串	42
3.5.4	其他操作	43
【本章小结】	43
【习题】	44
第 4 章	Java 语言中的面向对象特性	46
4.1	面向对象技术基础	46
4.1.1	面向对象的基本概念	46
4.1.2	面向对象的基本特征	47
4.2	Java 语言的面向对象特性	47
4.2.1	类	47
4.2.2	对象	53
4.2.3	面向对象特性	54
4.2.4	抽象类和接口	60
4.2.5	内部类	61
【本章小结】	63
【习题】	63
第 5 章	输入、输出流	66
5.1	File 类	66
5.1.1	构造函数	66
5.1.2	重要的方法	67
5.2	字节输入流	69
5.2.1	InputStream 类	69
5.2.2	FileInputStream 类	70
5.3	字节输出流	71
5.3.1	OutputStream 类	71
5.3.2	FileOutputStream 类	72
5.4	RandomAccessFile 类	73
5.5	字符输入流	76

5.5.1	InputStreamReader 类	76
5.5.2	FileReader 类	77
5.6	字符输出流	78
5.6.1	OutputStreamWriter 类	79
5.6.2	FileWriter 类	79
5.7	处理回压输入流	81
5.7.1	PushbackInputStream 类	81
5.7.2	PushbackReader 类	82
5.8	对象串行化	83
5.8.1	ObjectInputStream 类	84
5.8.2	ObjectOutputStream 类	85
5.8.3	串行化 (Serializable) 接口	86
5.8.4	操作打印机	88
	【本章小结】	89
	【习题】	89
第 6 章	多线程编程	92
6.1	Java 中的线程	92
6.2	Thread 类与 Runnable 接口	93
6.2.1	Thread 类	93
6.2.2	Runnable 接口	95
6.3	多线程程序案例	96
6.3.1	使用 Thread 类创建多线程	96
6.3.2	实现 Runnable 接口创建多线程	97
6.4	线程间的同步	101
6.4.1	同步代码段	103
6.4.2	同步方法	104
6.5	线程间的通信	106
	【本章小结】	110
	【习题】	110
第 7 章	Java Swing 基础	112
7.1	Swing 概述	112
7.2	Swing 组件与 AWT 组件的区别	113
7.3	从一个简单的 Swing 程序开始	113
7.4	重要的类	123
7.4.1	JComponent 类	123
7.4.2	顶层容器	125
7.4.3	布局管理	128

7.4.4 Swing 高级组件简介	131
【本章小结】	137
【习题】	137
第 8 章 Java 与多媒体	139
8.1 图像文件的显示	139
8.2 声音文件的播放	141
8.3 Java 媒体框架	142
【本章小结】	147
【习题】	148
第 9 章 网络编程	149
9.1 网络基础	149
9.2 套接字	151
9.2.1 TCP/IP 客户套接字	151
9.2.2 TCP/IP 服务器套接字	151
9.3 UDP 数据报	157
9.3.1 Java.net.DatagramPacket 类	158
9.3.2 Java.net.DatagramSocket 类	158
9.4 Java.net.URL 类	163
【本章小结】	165
【习题】	166
第 10 章 数据库编程	167
10.1 JDBC 简介	167
10.2 JDBC 的体系结构及主要接口	167
10.2.1 JDBC 驱动程序的类型	168
10.2.2 JDBC 的组成及其主要接口	169
10.3 建立数据库和设置数据源	169
10.3.1 SQL Server 2000 的设置	169
10.3.2 Access 的设置	170
10.4 JDBC 应用程序设计	170
10.4.1 加载 JDBC 驱动程序	171
10.4.2 建立连接对象	173
10.4.3 建立语句对象	174
10.4.4 建立结果集对象	176
10.5 JDBC 编程案例	179
【本章小结】	193
【习题】	193

第 11 章 课程设计	194
11.1 中国象棋	194
11.1.1 设计要求	194
11.1.2 总体设计	194
11.1.3 具体设计	194
11.1.4 提交成果	205
11.2 图书查询系统	206
11.2.1 设计要求	206
11.2.2 总体设计	206
11.2.3 具体设计	206
11.2.4 提交成果	211
【本章小结】	211

第 1 章 Java 概述

【学习目标】

1. 了解 Java 语言的发展历史。
2. 了解 Java 技术的特点和 Java 平台的应用划分。
3. 了解 Java 与 .NET 技术的关系与作用。
4. 学会利用网上资源。

1.1 Java 语言的发展史

1. Java 语言在因特网时代获得巨大成功

要把运行于 Windows 平台上的程序不做任何修改,就直接拿到 UNIX 系统上运行是不行的。因为程序的执行最终必须转换成为计算机硬件所能识别的机器指令来执行,专门为某一种计算机硬件和操作系统编写的程序是不能在另一种计算机硬件上执行的,至少要做代码移植工作。要想让程序能够在不同的计算机系统上运行,就要求程序设计语言能够跨越各种软件和硬件平台,而 Java 语言满足了这一需求。

1995 年, Sun 公司正式向 IT 界推出了 Java 语言, Java 语言具有安全、跨平台、纯面向对象、简单、适用于网络等显著特点。当时以网页为主要形式的因特网正在迅猛发展, Java 语言的出现迅速引起所有程序员和软件公司的极大关注,程序员们纷纷尝试用 Java 语言编写网络应用程序,并利用网络把程序发布到世界各地。包括 IBM、Oracle、微软、Netscape、Apple、SGI 等大公司纷纷与 Sun 公司签订合同,从而获得授权使用 Java 平台技术。微软公司主席比尔·盖茨在经过系统地研究后认为“Java 语言是长时间以来最卓越的程序设计语言”。目前, Java 语言已经成为网络编程的首选语言。

在经历了以大型机为核心的集中计算模式和以个人计算机为主的分散计算模式之后,因特网使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机的类型和操作系统是多样的,例如 Sun 工作站的硬件是 SPARC 体系,软件是 UNIX 中的 Solaris 操作系统,而个人计算机的硬件是 Intel 体系,操作系统是 Windows 或者 Linux,因此以前的编程语言基本上只是适用于一种计算机系统,例如 COBOL、FORTRAN、Prolog、C、C++ 等;网络计算模式的另一个特点是代码可以通过网络在不同的计算机上进行迁移,这样就迫切需要一种跨平台的编程语言,使得用它编写的程序能够在网络中的各种计算机上正常运行, Java 语言就是在这种需求下应运而生的。正是因为 Java 语言符合了因特网时代的发展要求,才使它获得了巨大的成功。Java 语言发展到今天,已经不仅是一种单纯的程序开发语言,而且发展成为了一个以 Java 语言为核心的应用于各种计算平台的一整套解决方案。Java 不仅包括语言、标准、组件、计算模式,还包括设备、网络技术及软件开发思想体系等一系列相关技术。我们统称为 Java 技术。

另外需要说明的是，Sun 公司是 Java 语言的创始者，同时也是 Java 技术发展的原动力和推动者。读者可以访问 Sun 公司的 Java 网站 (<http://java.sun.com>) 来获得最新的 Java 信息。

2. Java 语言的产生

Java 语言来自于 Sun 公司的一个叫做 Green 的项目。1991 年，Sun 公司的一个研究小组为了能够在消费电子产品上开发应用程序，开始积极寻找合适的编程语言。消费电子产品种类繁多，包括 PDA、机顶盒、手机等，由于同一类消费电子产品所采用的处理芯片和操作系统也不相同，也就存在着跨平台的问题。当时，最流行的编程语言是 C 和 C++，Sun 公司的研究人员开始试用 C++ 来开发消费电子产品的应用程序，但是结果表明，对于消费电子产品而言，C++ 语言过于复杂和庞大，并不适用，安全性也不令人满意。于是，Bill Joy 领导的研究小组就着手设计和开发出一种语言，称之为 Oak。该语言沿袭了许多 C 语言的语法，并提高了安全性，而且是纯面向对象的语言，但是 Oak 语言在商业上并未获得成功。到了 1995 年，由于因特网在世界上蓬勃发展，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因特网的需要，于是改进了该语言的设计，以达到如下几个目标：

- 创建一种纯面向对象的程序设计语言，而不是面向过程的语言；
- 提供一个解释执行的程序运行环境，使程序代码独立于平台；
- 吸收 C 和 C++ 的优点，使程序员容易过渡；
- 去掉 C 和 C++ 中影响程序鲁棒性的部分，使程序更安全，例如指针、内存申请和释放；
- 实现多线程功能，使得程序能够同时执行多个任务；
- 提供动态下载程序代码的机制；
- 提供代码校验机制以保证安全性。

最终，Sun 公司给该语言取名为 Java 语言，从而造就了一种成功的编程语言。

3. Java 平台 —— 是不断扩展的计算平台

Java 不仅是编程语言，还是一个开发平台，Java 技术给程序员提供了许多工具：编译器、解释器、文档生成器和文件打包工具等。同时 Java 还是一个程序发布平台，有两种“发布环境”，首先是 Java 运行时环境 (Java runtime environment, JRE) 包含了完整的类文件包，其次是许多浏览器都提供了 Java 解释器和运行时环境。目前 Sun 公司把 Java 平台划分成 J2EE、J2SE、J2ME 三个平台，针对不同的市场目标和设备进行定位。J2EE 是 Java 企业版 (Java 2 Enterprise Edition)，主要目的是为企业提供一个应用服务器的运行和开发平台。J2EE 本身是一个开放的标准，任何软件厂商都可以推出自己的符合 J2EE 标准的产品，使用户可以有多种选择。IBM、Oracle、BEA、HP 等 29 家公司已经推出了自己的产品，其中尤以 BEA 公司的 weblogic 产品和 IBM 公司的 websphere 最为著名。J2EE 将逐步发展成为可以与微软的 .NET 战略相对抗的网络计算平台。J2SE 是 Java 标准版 (Java 2 Standard Edition)，主要目的是为台式机和 workstation 提供一个开发和运行的平台。我们在学习 Java 的过程中，主要是采用 J2SE 来进行开发。J2ME (Java 2 Micro Edition)，主要是面向消费电子产品，为消费电子产品提供一个 Java 的运行平台，使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行。

1.2 Java 技术的特点

1. Java 虚拟机

Java 虚拟机并非一台真正的计算机，而是软件模拟的计算机，可以在任何处理器上（无论是在计算机中还是在其他电子设备中）安全并且兼容地执行保存在.class 文件中的字节码（Bytecode）。Java 虚拟机的“机器码”保存在.class 文件中，有时也可以称之为字节码文件。Java 程序的跨平台主要指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上运行，Java 虚拟机中的 Java 解释器负责将字节码文件解释成为特定的机器码来运行。Java 源程序需要通过编译器编译才能成为.class 文件（字节码文件或类文件）。

但是，针对不同的软硬件平台需要做专门的 Java 虚拟机，既要考虑硬件：处理器的型号；也要考虑软件：操作系统的种类。目前在 SPARC 结构、X86 结构、MIPS 和 PPC 等嵌入式处理芯片上，在 UNIX、Linux、Windows 和部分实时操作系统上都有专门的 Java 虚拟机。

说明：字节码文件是与平台无关的二进制码，执行时由解释器解释成为本地机器码，解释一句，执行一句。

2. 内存自动回收机制

在程序的执行过程中，部分内存在使用过后没有释放，而处于废弃状态，如果不及时进行无用内存的回收，就会造成内存泄露，进而导致系统崩溃。在 C++ 语言中，无用内存是由程序员人工进行回收的，程序员需要在编写程序的时候把不再使用的对象内存释放掉；但是这种人为的管理内存释放的方法却往往由于程序员的疏忽而致使内存无法回收，同时也增加了程序员的工作量。而在 Java 运行环境中，始终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，避免了内存泄露，也大大减轻了程序员的工作量。

3. 代码安全性检查机制

安全和方便总难两全其美。Java 语言的出现使得客户端计算机可以方便地从网络上下载 Java 程序到本机上运行，但是如何保证该 Java 程序不携带病毒或者不包含恶意攻击的代码呢？Java 语言以代码安全性检查机制来保证执行的安全性，虽然有时候少数程序员会抱怨说 applet 连文件系统也不能访问，不是很方便，但是正是实行各种安全措施才确保了 Java 语言的安全。

字节码的执行需要经过三个步骤，首先，由类装载器（class loader）负责把类文件加载到 Java 虚拟机中，在此过程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，例如 applet 程序中访问本机文件系统的操作；如果字节码校验器检验通过，则由 Java 解释器把该类文件解释成为机器码来执行。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以外的内存，如果是 applet 程序，还不允许访问客户端计算机的文件系统。

4. Java 语言的特点

(1) 简单、纯面向对象

Java 语言的简单首先体现在精简的系统上，Java 语言力图用最小的系统实现足够多的功

能；对硬件的要求不高，在小型的计算机上便可以良好地运行。和所有新一代的程序设计语言一样，Java 语言也采用了面向对象技术并实行得更加彻底，所有的 Java 程序和 applet 程序均是对象。其中，封装性实现了模块化和信息隐藏，继承性实现了代码的复用，用户可以建立自己的类库。而且 Java 语言采用的是相对简单的面向对象技术，去掉了运算符重载、多继承的复杂概念，而采用了单一继承、类强制转换、多线程、引用（非指针）等方式。无用内存自动回收机制也使得程序员不必费心管理内存，使程序设计更加简单，同时大大减少了出错的可能。Java 语言采用了 C 语言中的大部分语法，熟悉 C 语言的程序员会发现 Java 语言在语法上与 C 语言极其相似。

（2）强壮并且安全

Java 语言在编译及运行程序时，都要进行严格的检查。作为一种强制类型语言，Java 语言在编译和连接时都进行大量的类型检查，防止不匹配问题的发生。如果引用一个非法类型、或执行一个非法类型操作，Java 语言将在解释时指出该错误。在 Java 程序中不能采用地址计算的方法通过指针访问内存单元，大大减少了错误发生的可能性；而且 Java 语言的数组并非用指针实现，这样就可以在检查中避免数组越界的发生。无用内存自动回收机制也增加了 Java 语言的鲁棒性。

作为网络语言，Java 语言必须提供足够的安全保障，并且要防止病毒的侵袭。Java 语言在运行应用程序时，严格检查其访问数据的权限，比如不允许网络上的应用程序修改本地的数据。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，一旦字节代码被核实，便由 Java 解释器来执行，该解释器通过阻止对内存的直接访问来进一步提高 Java 的安全性。同时 Java 极高的鲁棒性也增强了 Java 语言的安全性。

（3）可移植性

Java 语言采用了多种机制来保证可移植性：首先，Java 语言从本质上是属于解释型语言，这意味着任何一台机器只要装备了 Java 解释器，即可以运行 Java 程序；其次，Java 语言的数据类型在任何机器上都是一致的，不支持特定硬件环境的数据类型。Java 语言的可移植性使得应用软件能够在任意平台上运行，这大大加快了软件产品的开发过程，增强了软件产品的可复用性。同时，也给软件产业带来一种崭新的开发模式。

（4）高性能

虽然 Java 语言是解释执行的，但它仍然具有非常高的性能，在一些特定的 CPU 上，Java 字节码可以快速的转换成为机器码进行执行。而且 Java 字节码格式的设计就是针对机器码的转换，实际转换时相当简便，自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和“即时编译”（just in time）技术的出现使得 Java 的执行速度有了更大的提高。

（5）解释执行、多线程并且是动态的

Java 语言的动态特性是其面向对象设计的延伸。Java 程序的基本组成单元为类，而 Java 的类又是运行过程动态装载的，这使得 Java 可以在分布环境中动态的维护应用程序及其支持类库之间的一致性，而不像 C++ 那样，每次当其所支持类库升级之后，相应的应用程序都必须重新编译。

1.3 Java 语言与 C、C++语言

为了使 C、C++程序员快速地了解 Java 语言，现做以下 Java 与 C、C++语言的比较。

1. 全局变量

Java 程序不能定义程序的全局变量，而类中的公共、静态变量就相当于这个类的全局变量。这样就使全局变量封装在类中，保证了安全性，而在 C、C++语言中，不加封装的全局变量往往会由于使用不当而造成系统的崩溃。

2. 条件转移指令

C、C++语言中用 `goto` 语句实现无条件跳转，而 Java 语言没有 `goto` 语言，通过例外处理语句 `try`、`catch`、`finally` 来取代它，提高了程序的可读性，也增强了程序的鲁棒性。

3. 指针

指针是 C、C++语言中最灵活，但也是最容易出错的数据类型。用指针进行内存操作往往造成不可预知的错误，而且，通过指针对内存地址进行显示类型转换后，可以访问类的私有成员，破坏了程序的安全性。在 Java 语言中，程序员不能进行任何指针操作，同时 Java 语言中的数组是通过类来实现的，很好地解决了数组越界这一 C、C++语言中不做检查的弱点。

4. 内存管理

在 C 语言中，程序员使用库函数 `malloc()`和 `free()`来分配和释放内存，C++语言中则是运算符 `new` 和 `delete`。再次释放已经释放的内存块或者释放未被分配的内存块，会造成系统的崩溃，而忘记释放不再使用的内存块也会逐渐耗尽系统资源。在 Java 语言中，所有的数据结构都是对象，通过运算符 `new` 分配内存并得到对象的使用权。无用内存回收机制保证了系统资源的完整，避免了由于内存管理不完善而导致的系统崩溃。

5. 数据类型的一致性

在 C、C++语言中，不同的硬件平台，编译器对简单的数据类型，如 `int`、`float` 等分配不同的字节数。例如：`int` 数据类型在 IBM PC 上为 16 位，在 VAX-11 上就为 32 位，从而导致了代码数据的不可移植。在 Java 语言中，对数据类型的字节数分配总是固定的，而不管是在何种的计算机平台上。因此就保证了 Java 语言数据的平台无关性和可移植性。

6. 类型转换

在 C、C++语言中，可以通过指针进行任意的类型转换，不安全因素大大增加。而在 Java 语言中系统在对对象的处理过程中，要进行严格的相容性检查，防止不安全的转换。

7. 头文件

在 C、C++语言中，使用头文件声明类的原型和全局变量及库函数等，在大的系统中，维护这些头文件是非常困难的。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，防止非法的访问。同时，Java 语言中用 `import` 语句与其他类进行通信，以便访问其他类的对象。

8. 结构和联合

在 C、C++语言中用结构和联合来表示一定的数据结构，但是由于其成员均为公有的，

因此存在安全性问题。Java 语言不支持结构和联合，通过类把数据结构及对该数据的操作都封装在类里面。

9. 预处理

C、C++语言有宏定义，用宏定义实现的代码往往影响程序的可读性，而 Java 语言不支持宏定义。

为易于实现跨平台性，Java 语言被设计成为解释执行，字节码本身包含了许多编译时生成的信息，这使连接过程更加简单。而多线程使应用程序可以同时进行不同的操作，处理不同的事件。在多线程机制中，不同的线程处理不同的任务，互不干涉，不会由于某一任务处于等待状态而影响了其他任务的执行，这样就可以容易地实现网络上的实时交互操作。Java 语言在执行过程中，可以动态地加载各种类库，这一特点使之非常适合于网络运行，同时也有利于软件的开发，因为即使是更新类库也不必重新编译使用这一类库的应用程序。

1.4 Java 与 .NET 技术

2000 年 6 月，微软公司发布了 .NET 战略，该技术是与 Sun 公司的 Java One 技术相对应的。抛开商业竞争的因素，从技术角度来讲，二者是相似的，它们都是因特网现状下的基于网页的分布式计算方案。.NET 是微软公司的 XML Web 服务 (Web Service) 平台，XML Web 服务是为应用程序提供数据和服务的逻辑单元。XML Web 服务结合了组件技术和 Web 技术的优秀方面，它是 .NET 编程模型的基石。

.NET 的核心概念就是“把软件当做服务”，也就是把软件应用产品与商业、内容、信息服务合并成一种事务，使之成为可以在网络上订阅使用的服务形式。人们设计、构造、实施、运作、集成和使用软件的方式都可以通过网络完成，因此也就要按照使用这些服务的不同方式支付相应的费用。.NET 模式希望把计算模式从单机、客户机/服务器和 Web 网站的方式彻底地转向分布式计算。在 .NET 策略中有几个重要的概念：

- XML 技术。XML 被称为网络计算的世界语，是一种代替 HTML 的可扩展标记语言。XML 采用文本标记的形式定义各种可交换数据结构，并且可以利用标准的网络协议进行传输。正因为这些特性，XML 实际上代表了平台中性和进行网络计算的趋势。
- 网页服务。网页服务是 .NET 的核心概念，它是基于网络的分布式应用程序的基本构造模块，而这些程序是以平台、对象模板和多语言方式构建的。网页服务建立在像 SOAP 和 XML 之类的开放的因特网标准之上，并且由此形成了可编程网络理念的基础。
- 通用语言运行时 (common language runtime, CLR)。它可以调用并运行任何编程语言所写的代码。CLR 具有交叉语言集成、自述组件、简单配制、版本化以及集成安全服务等特点。这一点十分类似于 Java 虚拟机的概念。
- C# 语言。它是一种新型的编程语言，为 .NET 策略而设计。它与 Java 语言比较类似。

从一个 Java 程序员的眼中看 .NET 策略，可以发现它继承了许多 Java 技术的思想精髓，并且用微软的商业模式成功地加以包装。.NET 的最初想法是希望进行接近操作系统平台的定制开发，当然，这是指使用 Windows 系统 (目前是 Windows XP、Windows ME 和 Windows 2000)。Visual Basic 和 C# 是 .NET 平台上最重要的开发语言，并且它们不能在其他平台上运

作。微软声称有许多开发商在开发与 .NET 的 CLR 相合作的语言，但直到今天，我们看到 CLR 还只是一个 Windows 版的技术。这就说明存在一个重要的互用性问题，因为每种编程语言（根据定义来划分）都有其各自特定的数据类型和数据结构。

图 1-1 简要的对 .NET 与 J2EE 技术进行了对比，从中不难发现 J2EE 与 .NET 既有许多相似之处，又存在不少区别。微软作为最成功的 PC 软件制造商，它推出的 .NET 策略必然以它自己的商业模式加以推广，而作为优秀的跨平台分布式计算技术的 Java 语言，其发展前景更加广阔，因此如何做到二者的互用性是一个非常重要的问题。

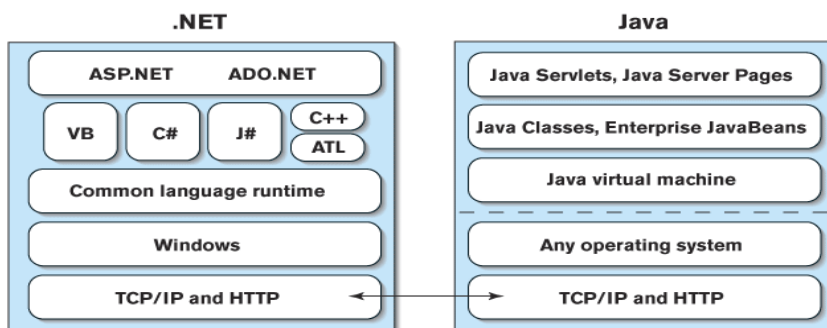


图 1-1 .NET 与 Java 的比较

1.5 J2ME

手机作为现代最重要的通信手段，是最重要的嵌入式电子产品之一。目前，Java 在手机设备上的应用方兴未艾，Java 正在成为手机应用软件开发的主流技术。

针对嵌入式设备及消费类电器的 J2ME 在推出之后，首先做出反应的就是全球各大手机制造商，他们把 J2ME 引进到手机上，使我们平常通信用手机更加易用，更加个性化。J2ME 一经推出，摩托罗拉公司作为 J2ME 的主要支持者，第一个在自己生产的手机上移植了 KVM（千字节虚拟机），又领导全球手机厂家制定了 MIDP（动信息设备框架）规范。目前 KVM 和 MIDP 的手机软件开发体系已经成为 Java 手机程序开发的事实标准。

回顾移动电话的发展历史，不难发现移动电话上的应用软件的发展经历了三个阶段。传统的移动电话通常只有通话和短消息功能，只能提供基本的语音服务。随后移动电话又增加了一些简单的附加应用，如电话簿和电话铃声编辑功能等。而现在随着 WAP 技术的发展，移动电话增加了访问因特网的功能，使用户可以直接在手机上以无线方式浏览网页。然而，随着无线因特网新应用的出现，新的问题也随之而来。

首先，它面临的是开发瓶颈的问题。通常，手机类嵌入式系统普遍使用 C 语言和专用的实时操作系统，开发速度慢，也没有动态加载应用程序的能力。在移动电话上开发应用程序变得越来越困难，一方面单纯依靠手机厂商自身的软件开发能力难以满足市场的需求，而另一方面，广大的软件开发商却又无法参与进来，来开发适用于移动电话的应用程序。这无疑极大地制约了新应用的推广与普及。

其次，移动电话访问因特网只能通过 WAP 方式，而 WAP 采用 Browser/Server 方式访问因特网功能有限。现在的 WAP 解决方案要求手机通过 WAP 网关才能访问因特网，而且只能

访问 WML 而不是主流的 HTML，也不能显示复杂格式的图形。此外，因为现有的 WAP 解决方案不够智能，而且不能访问本地存储区，如果进行在线交易会增加服务器负荷，反应速度慢，使无线因特网应用受到了很大的限制。

Java 的应用则使上述问题迎刃而解。因为 Java 语言是跨平台运行的，这一特性使第三方软件开发商可以很容易地介入进来开发应用程序，也可以很方便地将应用程序安装移植到移动电话上，开发周期也大大缩短，而且还能支持应用程序的动态下载和升级。Java 还提供了 HTTP 协议，使移动电话能以客户机/服务器方式直接访问因特网的全部信息，不同的客户机访问不同的文件，此外还能访问本地存储区，提供最高效率的在线交易。

基于 KVM 的 Java 除了能够更好地增强完善移动电话上已有的应用外，还进一步增加了字典、图书、游戏、遥控家电和定时提醒等新的应用，并能访问电子邮件、即时消息、股票和电子地图等信息。

最近 Java 技术团体的成员开始大力推动 Java 在小型设备上的应用。他们已经瞄准了那些专门为手机开发应用的独立软件开发商。近日，Java 组织就发布了“Java 规范要求 JSR185”，这是一个专门为手机应用开发商设计的融合了目前主要的 Java API 的体系结构，包括 CLDC、CLDC 1.0、CDC、MIDP、MIDP 2.0、无线信息传送 API 以及无线媒体传送 API 等。用于测试产品兼容性的测试包也即将推出。

目前已经有多家手机制造商表示支持 JSR185 规范，包括摩托罗拉、诺基亚、索尼爱立信以及日本的 NTT DoCoMo 等。

1.6 J2EE

J2EE 是一种全新概念的模型，与传统的互联网应用程序模型相比有着不可比拟的优势。当今许多企业都需要扩展他们的业务范围，降低自身经营成本，缩短他们和客户之间的响应时间，为此这就需要一种简捷、快速的服务。典型地说，提供这些服务的应用软件必须同企业信息系统（EIS）相结合，并向更为广阔的用户提供新的服务。这些服务要具备以下特点：

- 高可用性。来满足现在的全球商业环境。
- 安全性。保护用户的隐私和企业数据的安全。
- 可依赖性和可扩展性。保证商业交易的正确和迅捷。

通常这些服务是由分布的应用程序组成的，包括前端客户端和后端数据源以及它们之间的一层或几层，这些中间层提供了把商业功能和数据与 EIS 相结合的功能。这些中间层把客户端从复杂的商业逻辑中分离出来，利用成熟的因特网技术使用户在管理上所花费的时间最小。

J2EE 降低了开发这种中间层服务的成本和复杂程度，因而使得服务可以被快速地展开，并能够更轻松地面面对竞争中的压力。J2EE 通过定义一种标准的结构来实现它的优势，具体内容如下：

- J2EE Application Programming Model。一种用于开发多层次，瘦型客户用户程序的标准设计模型。
- J2EE Platform。一个标准的平台，用来整合 J2EE 的应用程序，指定一系列的接口和方法。
- J2EE Compatibility Test Suite。一套兼容测试组件，用来检测产品是否同 J2EE 平台兼容。