普通高等教育“十二五”应用型本科规划教材

Java语言程序设计

臧文科 编著

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

普通高等教育“十二五”应用型本科规划教材

Java语言程序设计

编　著　臧文科
编委会　许文杰　马　骁　齐　峰

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本教材内容详尽，取舍和安排恰当、循序渐进，讲解通俗易懂，实例丰富，并注重培养解决实际问题的能力。

全书从 Java 语言基本的概念开始讲述 Java 语言，包括 Java 语言的开发环境建立；Java 数据类型、运算符、表达式与流程控制、数组和方法等。用比较易于理解和接受的讲叙方法、恰当的内容安排对 Java 面向对象程序设计的基本概念，如类、对象、接口、继承和多态等进行了深入浅出的讲解。通过大量的编程实例对 Java 的编程应用进行讲解。对 Java 语言的特点，如异常处理、多线程应用等作了详细的讲解。对 Java 的输入输出处理等通过实例进行了深入的说明。

本书最后针对时下流行的 Android 开发，从 Java 角度做了一定讲解，希望能够从中获得对 Andorid 开发的基本认识。

图书在版编目(CIP)数据

Java 语言程序设计/臧文科编著. —西安：西安交通大学出版社，2014.12

ISBN 978-7-5605-6942-0

Ⅰ. ①J… Ⅱ. ①臧… Ⅲ. ①JAVA 语言-程序设计-高等学校-教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 307138 号

书　　名	Java 语言程序设计
编　　著	臧文科
责任编辑	李　佳　曹　昳
出版发行	西安交通大学出版社 (西安市兴庆南路 10 号　邮政编码 710049)
网　　址	http://www.xjtupress.com
电　　话	(029)82668357　82667874(发行中心) (029)82668315　82669096(总编办)
传　　真	(029)82668280
印　　刷	北京京华虎彩印刷有限公司
开　　本	787mm×1092mm　1/16　印张 19.375　字数 473 千字
版次印次	2014 年 12 月第 1 版　2014 年 12 月第 1 次印刷
书　　号	ISBN 978-7-5605-6942-0/TP·650
定　　价	45.00 元

读者购书、书店添货、如发现印装质量问题，请与本社发行中心联系、调换。
订购热线：(029)82665248　(029)82665249
投稿热线：(029)82669097　QQ：8377981
读者信箱：lg_book@163.com

前言

FOREWORD

本书从 Java 语言最基本的入门概念开始讲述 Java 语言，包括 Java 语言的开发环境建立；Java 数据类型、运算符、表达式与流程控制、数组和方法等。用比较易于理解和接受的讲叙方法，适当的内容安排对 Java 面向对象程序设计的基本概念，如类、对象、接口、继承和多态等进行了深入浅出的讲解。通过大量的编程实例对 Java 的编程应用进行讲解，包括：图形绘制和图像显示，图形用户界面中的基本控制组件、容器和布局、常用的对话框和菜单设计的应用、Java Applet 小应用程序、JDBC 数据库编程、JSP 网络编程等进行了讲述。对 Java 语言的特点，如异常处理、多线程应用等作了详细的讲解。对 Java 的输入输出处理等通过实例进行了深入的说明。本书最后针对时下流行的 Android 开发，从 Java 角度做了初步阐述，希望能够从中获得对 Andorid 开发的基本认识。

本书每章都安排了大量有针对性的练习和编程实训题，便于教师教学和检验学生的学习效果。书中内容比较详尽，内容的取舍和安排恰当、循序渐进，讲解通俗易懂，实例丰富，并注重培养解决实际问题的能力。本书可作为高等院校“Java 程序设计”课程的教材和教学参考书，特别适合 Java 语言的初学者使用，也可作为对 Java 编程感兴趣的读者的参考书。

本书编写过程中，得到了毕雪、任丽艳、毕莹、马风芳、孙佃良、郭启攀等同学的大力支持，他们认真仔细地核对了书中的内容，在此表示感谢。由于时间仓促和水平所限，书中难免有纰漏之处，请各位读者批评指正。

编　者

2014 年 10 月

JAVA YUYAN CHENGXU SHEJI

Java 语言程序设计 目录

CONTENTS

第 1 章 Java 引言

1.1 Java 概述 /001
1.2 Java 虚拟机 JVM /007
1.3 JDK /007
1.4 Eclipse /012
1.5 其他工具 /015
1.6 Java 应用程序和小应用程序 /017
1.7 Java 程序结构 /019
1.8 本章小结 /020
1.9 习题 1 /020

第 2 章 Java 编程基础

2.1 简单数据类型 /022
2.2 运算符和表达式 /028
2.3 控制语句 /032
2.4 数组 /042
2.5 异常处理 /050
2.6 集合类 /053
2.7 本章小结 /055
2.8 习题 2 /055

第 3 章 面向对象编程

3.1 面向对象问题求解的提出 /061
3.2 面向对象的基本特征 /063
3.3 Java 中的类 /065
3.4 类的修饰符 /073
3.5 类成员变量 /076
3.6 类成员方法 /080
3.7 对象的初始化和清除 /084
3.8 习题 3 /086

第 4 章　图形用户界面

4.1　图形界面设计基础 /089
4.2　框架窗口 /092
4.3　其他控件 /095
4.4　布局设计 /098
4.5　面板 /103
4.6　文本框和文本区 /105
4.7　选择框和单选按钮 /111
4.8　列表和组合框 /115
4.9　菜单 /119
4.10　对话框 /123
4.11　滚动条 /125
4.12　鼠标事件 /126
4.13　键盘事件 /133
4.14　习题 4 /134

第 5 章　Web 编程

5.1　JSP 简介 /136
5.2　JSP 开发环境 /137
5.3　JSP 运行机制与基本语法 /143
5.4　JSP 内置对象 /157
5.5　Java Bean /164
5.6　JSP 文件操作 /168
5.7　习题 5 /177

第 6 章　网络编程

6.1　Java 网络编程起步 /181
6.2　URL /188
6.3　Java 与 TCP 网络协议开发 /199
6.4　ServerSocket /209
6.5　基于多线程的通信程序 /219

6.6 数据报 /225
6.7 习题 6 /235

第 7 章 Java 数据库编程

7.1 数据库知识 /237
7.2 结构化查询语句 SQL /237
7.3 SQL Server 数据库 /238
7.4 数据库的连接 /240
7.5 Java 数据库应用案例 /256
7.6 本章小结 /266
7.7 习题 7 /266

第 8 章 移动编程

8.1 Android 平台 /268
8.2 Android 平台搭建 /269
8.3 Android 开发初探 /272
8.4 Android 设计进阶 /280
8.5 Android 高级应用 /297
8.6 本章小结 /301
8.7 习题 8 /301

参考文献

第1章 Java 引言

JAVA YUYAN

1.1 Java 概述

1.1.1 现代编程语言的诞生:C 语言

C 语言的产生震撼了整个计算机界,它从根本上改变了编程的方法和思路。C 语言的产生是人们追求结构化、高效率、高级语言的直接结果,可用它替代汇编语言开发系统程序。当设计一种计算机语言时,经常要从以下几方面进行权衡:

- 易用性与功能性
- 安全性和效率性
- 稳定性和可扩展性

C 语言出现以前,程序员们不得不经常在有优点又有欠缺的语言之间做出选择。例如,尽管公认 FORTRAN 在科学计算应用方面可以编写出相当高效的程序,但它不适合编写系统程序。BASIC 虽然容易学习,但功能不够强大,并且谈不上结构化,这使它应用到大程序的有效性受到怀疑。汇编语言虽能写出高效率的程序,但是学习或有效地使用它却不容易。而且,调试汇编程序也相当困难。

另一个问题是,早期设计的计算机语言(如 BASIC,COBOL,FORTRAN 等)没有考虑结构化设计原则,使用 GOTO 语句作为对程序进行控制的一种主要方法。这样做的结果是,用这些语言编写的程序往往成了"意大利面条式的程序代码",一大堆混乱的跳转语句和条件分支语句使得程序几乎不可能被读懂。Pascal 虽然是结构化语言,但它的设计效率比较低,而且缺少几个必需的特性,因而无法在大的编程范围内使用。

因此,在 C 语言产生以前,没有任何一种语言能完全满足人们的需要,但人们对这样一种语言的需要却是迫切的。在 20 世纪 70 年代初期,计算机革命开始了,对软件的需求量日益增加,使用早期的计算机语言进行软件开发根本无法满足这种需要。学术界付出很多努力,尝试创造一种更好的计算机语言。此时,促使 C 语言诞生的另一个,也许是最重要的因素,即计算机硬件资源的富余为其诞生带来了机遇。计算机不再像以前那样被紧锁在门里,程序员们可以随意使用计算机,可以随意进行自由尝试,因而也就有了可以开发适合自己使用的工具的机会。所以,在 C 语言诞生的前夕,计算机语言向前飞跃的时机已经成熟。

许多人认为 C 语言的产生标志着现代计算机语言时代的开始。它成功地综合处理了长期困扰早期语言的矛盾属性。C 语言是功能强大、高效的结构化语言,简单易学,而且它还包括一个无形的方面:它是程序员自己的语言。在 C 语言出现以前,计算机语言要么被作为学术实验而设计,要么由官僚委员会设计。而 C 语言不同。它的设计、实现、开发由真正的

从事编程工作的程序员来完成，反映了现实编程工作的方法。它的特性经由实际运用该语言的人们不断去提炼、测试、思考、再思考，使得 C 语言成为程序员们喜欢使用的语言。确实，C 语言迅速吸引了许多狂热的追随者，也受到许多程序员的青睐。简言之，C 语言是由程序员设计并由他们使用的一种语言。

1.1.2 对 C++的需要

在 20 世纪 70 年代末和 80 年代初，C 成为了主流的计算机编程语言，至今仍被广泛使用。既然 C 是一种成功且有用的语言，为什么还需要新的计算机语言？答案是复杂性(complexity)。程序越来越复杂这一事实贯穿编程语言的历史，C++正是适应了这一需求而产生的。

自从计算机发明以来，编程方法经历了戏剧性的变化。例如，当计算机刚发明出来时，编程是通过面板触发器用人工打孔的办法输入二进制机器指令来实现的。对于只有几百行的程序，这种办法是可行的。随着程序不断增大，人们发明了汇编语言，它通过使用符号来代替机器指令，这样程序员就能处理更大、更复杂的程序。随着程序的进一步增大，高级语言产生了，它给程序员提供了更多的工具来处理复杂性问题。

第一个被广泛使用的高级语言是 FORTRAN。尽管 FORTRAN 最初给人留下了深刻的印象，但它无法开发出条理清楚易于理解的程序。20 世纪 60 年代提出了结构化编程方法。这种结构化的编程思想被像 C 这样的语言所应用，第一次使程序员可以相对轻松地编写适度复杂的程序。然而，当一个工程项目达到一定规模后，即使使用结构化编程方法，编程人员也无法对它的复杂性进行有效管理。20 世纪 80 年代初期，许多工程项目的复杂性都超过了结构化方法的极限。为解决这个问题，面向对象编程(object-oriented programming，OOP)新方法诞生了。这里给出一个简短的定义：面向对象的编程是通过使用继承性、封装性和多态性来帮助组织复杂程序的编程方法。

总之，尽管 C 是世界上伟大的编程语言之一，但它处理复杂性的能力有限。一旦一个程序的代码超过 25000 行，就很难从总体上把握它的复杂性了。C++突破了这个限制，帮助程序员理解并且管理更大的程序。

1979 年，Bjarne Stroustrup 在新泽西州的 Murray Hill 实验室工作时发明了 C++。Stroustrup 最初把这种新语言称为“带类的 C”。1983 年，改名为 C++。C++通过增加面向对象的特性扩充了 C。因为 C++产生在 C 的基础之上，因此它包括了 C 所有的特征、属性和优点。这是 C++作为语言成功的一个关键原因。C++的发明不是企图创造一种全新的编程语言，而是对一个已经高度成功的语言做出改进。C++在 1997 年 11 月被标准化，目前的标准是 ANSI/ISO。

1.1.3 Java 的出现

在 20 世纪 80 年代末和 90 年代初，使用面向对象编程的 C++语言占主导地位。有一段时间程序员似乎都认为已经找到了一种完美的语言。因为 C++既有面向对象的特征，又有 C 语言高效和格式上的优点，因此它是一种可以被广泛应用的编程语言。然而，就像过去一样，推动计算机语言进化的力量正在不断酝酿。在随后的几年里，万维网(WWW)和 Internet 的发展终于促成编程的另一场革命。

Java 的最初推动力并不是因特网，而是源于对独立于平台(也就是体系结构中立)语言

的需要，这种语言可创建能够嵌入微波炉、遥控器等各种家用电器设备的软件。用作控制器的 CPU 芯片是多种多样的，但 C 和 C++以及其他绝大多数语言的缺点是只能对特定目标进行编译，而创建编译器是一项既耗资巨大又耗时较长的工作。因此需要一种简单且经济的解决方案。为了找到这样一种方案，Gosling 和其他人开始一起致力于开发一种可移植、跨平台的语言，该语言能够生成运行于不同环境、不同 CPU 芯片上的代码。他们的努力最终促成了 Java 的诞生。

在 Java 的一些细节被设计出来的同时，第二个并且也是最重要的因素出现了，该因素将对 Java 的未来起着至关重要的作用，这第二个因素就是万维网(WWW)。如果万维网的成型和 Java 的实现不是同时发生的话，那么 Java 可能保持它有用、但默默无闻的用于电子消费品编程语言的状态。随着万维网的出现，Java 被推到计算机语言设计的最前沿，因为万维网也需要可移植的程序。

1993 年，Java 设计小组的成员发现，他们在编制嵌入式控制器代码时经常遇到的可移植性问题在编制因特网代码的过程中也出现了。事实上，开始被设计为解决小范围问题的 Java 语言同样可以被用在大范围的因特网上。这个认识使他们将 Java 的重心由电子消费品转移到 Internet 编程。因此，中立体系结构编程语言的需要是促使 Java 诞生的源动力，而 Internet 却最终导致了 Java 的成功。

Java 的大部分特性是从 C 和 C++中继承的。Java 设计人员之所以故意这么做，主要是因为他们觉得，在新语言中使用熟悉的 C 语法及模仿 C++面向对象的特性，将使他们的语言对经验丰富的 C/C++程序员有更大的吸引力。除了表面类似外，其他一些促使 C 和 C++成功的因素也帮了 Java 的忙。首先，Java 的设计、测试、精炼由真正从事编程工作的人员完成，它根植于设计它的人员的需要和经验，因而也是一个程序员自己的语言。其次，Java 是紧密结合的且逻辑上是协调一致的。最后，除了那些 Internet 环境强加的约束以外，Java 给了编程人员完全的控制权。如果你程序编得好，你编写的程序就能反映出这一点。相反，如果你的编程手法拙劣，也能在你的程序中反映出来。换一种说法，Java 并不是训练新手的语言，而是供专业编程人员使用的语言。

由于 Java 和 C++之间的相似性，容易使人将 Java 简单地想象为“C++的版本”。但其实这是一种误解。Java 在实践和理论上都与 C++有重要的不同点。尽管 Java 受到 C++的影响，但它并不是 C++的增强版。例如，Java 与 C++既不向上兼容，也不向下兼容。当然，Java 与 C++的相似之处也是很多的，如果你是一个 C++程序员，你会感觉到对 Java 非常熟悉。另外一点是：Java 并不是用来取代 C++的，设计 Java 是为了解决某些特定的问题，而设计 C++是为了解决另外一类完全不同的问题。两者将长时间共存。

1.1.4 Java 语言特点

Java 语言是适用于分布式计算环境的面向对象编程语言，它虽类似 C 和 C++，但比 C++简单，忽略了许多为提高计算效率而使初学者较难掌握的程序语言特性。Internet 得到 Java 语言的支持，可以实现真正的交互，人们使用浏览器能“漫游”丰富多彩的 Internet 世界。

Java 语言主要有以下特点：

(1)强类型

Java 语言是一种强类型语言，强类型能约束程序员必须遵守更多的编程规定，也能让编

译器检测出程序中尽可能多的错误。

(2)编译和解释

Java 语言是一种高级编程语言,用 Java 语言编写的源程序在计算机上运行需经过编译和解释执行两个严格区分的阶段。Java 语言的编译程序先将 Java 源程序翻译成与机器无关的字节码(byte code),不是通常的编译程序将源程序翻译成计算机的机器代码。运行时,Java 的运行系统和链接需要执行的类,作必要的优化后,解释执行字节码程序。

(3)自动无用内存回收功能

Java 语言具有自动无用内存回收功能,程序可以按需使用内存,但不需要对无用内存显式地撤销分配。系统有一个垃圾收集器(garbage collector),自动收集程序不再使用的内存。这样,能避免显式的撤销分配所引起的安全问题。Java 语言不再含有任何不安全的语言成分。例如,没有指针,数组元素都要检查下标是否越界。

(4)面向对象

面向对象是程序员编写大型程序、有效控制程序复杂性的重要手段。Java 语言在面向对象方面,比 C++更"纯",它的所有数据类型,包括布尔类型、整形、字符型等,都有相应的类,程序可完全基于对象编写。

面向对象语言主要有封装性、继承性和多态性三个特点。封装就是将实现细节隐藏起来,只给出如何使用的信息。数据及数据上的操作用类封装,对象是类的实例,外界使用对象中的数据及可用的操作受到一定的限制。继承体现众多的一种层次对象的特性,下一层的类可从上一层的类继承定义,从上一层类派生的类的对象能继承上一层对象的特性,同时可以改变和扩充一些特性,以适应其自身的特点。多态性的意义主要体现在逻辑上相同的不同层次上的操作,使用相同的操作名,根据具体对象,能自动选择对应的操作。Java 语言很实用地实现了这三种特性。

(5)与平台无关

与平台无关是对程序可移植性最直接最有效的支持。Java 语言的设计者在设计时重点考虑了 Java 程序的可移植性,采用多种机制来保证可移植性,其中最主要的是定义了一种虚拟机(virtual machine),以及虚拟机使用的 Java 字节码。在任何平台上,Java 源程序被 Java 编译器编译成虚拟机能够识别的字节码。这样,只要有 Java 虚拟机的平台,就能解释执行 Java 字节码程序,从而实现 Java 与平台无关。另外,Java 语言还采用基于国际标准的数据类型,在任何平台上,同上种数据类型是一致的。例如,用 int 标识 32 位二进制位(bit)整型数据,那么无论在哪一台计算机上,Java 的 int 数据都是 32 位整数。相反,C 语言会随着硬软件平台的改变,用 int 标识的整数位数也可能不全相同。

Java 语言提高可移植性的代价是降低程序的执行效率。出于 Java 语言也是一种解释执行的语言,Java 程序的执行速度与 C 程序的执行速度有较大的差别。不过,为了尽量弥补执行效率低的缺陷,Java 的字节码在设计上非常接近现代计算机的机器码,这有助于提高解释执行的速度。

(6)安全性

Java 是在网络环境中使用的编程语言,必须考虑安全性问题,主要有以下两个方面:

设计的安全防范:Java 语言没有指针,避免程序因为指针使用不当,访问不应该访问的

内存空间；提供数组元素上标检测机制，禁止程序越界访问内存；提供内存自动回收机制，避免程序遗漏或重复释放内存。运行安全检查：为了防止字节码程序可能被非法改动，解释执行前，先对字节码程序作检查，防止网络“黑客”对字节码程序已作了恶意改动，达到破坏系统的目的。最后，浏览器限制下载的小应用程序不允许访问本地文件，避免小应用程序破坏本地文件。

(7)分布式计算

Java 语言支持客户机/服务器计算模式。Java 程序能利用 URL 对象，能访问网络上的对象，如同访问本地的文件一样，实现数据分布。另外，Java 的客户机/服务器模式也可以把计算从服务器分散到客户机端，实现操作分布。

(8)多线程

线程是比进程更小的一种可并发执行的单位，每个进程都有自己独立的内存空间和其他资源，当进程切换时需要进行数据和资源的保护与恢复。若干协同工作的线程可以共享内存空间和资源，线程切换不需要数据的保护与恢复。

Java 的运行环境采用多线程实现，可以利用系统的空闲时间执行诸如内存回收等操作；Java 语言提供语言级多线程支持，用 Java 语言能直接编写多线程程序。

1.1.5 对 Internet 的重要性

Internet 使 Java 成为网上最流行的编程语言，同时 Java 对 Internet 的影响也意义深远。在网络中，有两大类对象在服务器和个人计算机之间传输：被动的信息和动态的、主动的程序。例如，当你阅读电子邮件时，你在看被动的数据。甚至当你下载一个程序时，该程序的代码也是被动的数据，直到你执行它为止。但是，可以传输到个人计算机的另一类对象却是：动态的、可自运行的程序，虽然这类程序是客户机上的活动代理，但却是由服务器来初始化的。例如，被服务器用来正确地显示服务器传送数据的程序。

Java 可用来生成两类程序：应用程序(Application)和小应用程序(Java applet)。应用程序是可以在你的计算机的操作系统中运行的程序，从这一方面来说，用 Java 编制的应用程序多多少少与使用 C 或 C++编制的应用程序有些类似。在创建应用程序时，Java 与其他计算机语言没有大的区别。而 Java 的重要性就在于它具有编制小应用程序的功能。小应用程序是可以在 Internet 中传输并在兼容 Java 的 Web 浏览器中运行的应用程序。小应用程序实际上就是小型的 Java 程序，能像图像文件、声音文件和视频片段那样通过网络动态下载，它与其他文件的重要差别是，小应用程序是一个智能的程序，能对用户的输入作出反应，并且能动态变化，而不是一遍又一遍地播放同一动画或声音。

如果 Java 不能解决两个关于小应用程序的最棘手的问题：安全性和可移植性，那么小应用程序就不会如此令人激动。让我们先说明这两个术语对 Internet 的意义。

1. 安全性

每次当你下载一个“正常”的程序时，你都要冒着被病毒感染的危险。在 Java 出现以前，大多数用户并不经常下载可执行的程序文件；即使下载了程序，在运行它们以前也都要进行病毒检查。尽管如此，大多数用户还是担心他们的系统可能被病毒感染。除了病毒，另一种恶意的程序也必须警惕。这种恶意的程序可通过搜索你计算机本地文件系统的内容来收集你的私人信息，例如信用卡号码、银行账户结算和口令。Java 在网络应用程序和你的计

算机之间提供了一道防火墙(firewall),消除了用户的这些顾虑。

当使用一个兼容 Java 的 Web 浏览器时,你可以安全地下载 Java 小应用程序,不必担心病毒的感染或恶意的企图。Java 实现这种保护功能的方式是,将 Java 程序限制在 Java 运行环境中,不允许它访问计算机的其他部分,后面将介绍这个过程是如何实现的。下载小应用程序并能确保它对客户机的安全性不会造成危害是 Java 的一个最重要的方面。

2. 可移植性

许多类型的计算机和操作系统都连接到 Internet 上。要使连接到 Internet 上的各种各样的平台都能动态下载同一个程序,就需要有能够生成可移植性执行代码的方法。很快将会看到,有助于保证安全性的机制同样也有助于建立可移植性。实际上,Java 对这两个问题的解决方案是完美的也是高效的。

1.1.6 Java 的魔力:字节码

Java 解决上述两个问题——安全性和可移植性的关键在于 Java 编译器的输出并不是可执行的代码,而是字节码(byte code)。字节码是一套设计用来在 Java 运行时系统下执行的高度优化的指令集,该 Java 运行时系统称为 Java 虚拟机(Java Virtual Machine,JVM)。在其标准形式下,JVM 就是一个字节码解释器。事实上,出于对性能的考虑,许多现代语言都被设计为编译型,而不是解释型。然而,正是通过 JVM 运行 Java 程序才有助于解决在 Internet 上下载程序的主要问题。这就是 Java 输出字节码的原因。

将一个 Java 程序翻译成字节码,有助于它更容易地在一个大范围的环境下运行程序。原因非常直接:只要在各种平台上都实现 Java 虚拟机就可以了。在一个给定的系统中,只要系统运行包存在,任何 Java 程序就可以在该系统上运行。记住:尽管不同平台的 Java 虚拟机的细节有所不同,但它们都解释同样的 Java 字节码。如果一个 Java 程序被编译为本机代码,那么对于连接到 Internet 上的每一种 CPU 类型,都要有该程序的对应版本。这当然不是一个可行的解决方案。因此,对字节码进行解释是编写真正可移植性程序的最容易的方法。

对 Java 程序进行解释也有助于它的安全性。因为每个 Java 程序的运行都在 Java 虚拟机的控制之下,Java 虚拟机可以包含这个程序并且能阻止它在系统之外产生副作用。正如你将看到的,Java 语言特有的某些限制增强了它的安全性。

被解释的程序的运行速度通常确实会比同一个程序被编译为可执行代码的运行速度慢一些。但是对 Java 来说,这两者之间的差别不太大。使用字节码能够使 Java 运行时系统的程序执行速度比你想象的快得多。

尽管 Java 被设计为解释执行的程序,但是在技术上 Java 并不妨碍动态将字节码编译为本机代码。SUN 公司在 Java 发行版中提供了一个字节码编译器——JIT(Just In Time,即时)。JIT 是 Java 虚拟机的一部分,它根据需要、一部分一部分地将字节码实时编译为可执行代码。它不能将整个 Java 程序一次性全部编译为可执行的代码,因为 Java 要执行各种检查,而这些检查只有在运行时才执行。记住这一点是很重要的,因为 JIT 只编译它运行时需要的代码。尽管如此,这种即时编译执行的方法仍然使性能得到较大提高。即使对字节码进行动态编译后,Java 程序的可移植性和安全性仍能得到保证,因为运行时系统(该系统执行编译)仍然能够控制 Java 程序的运行环境。不管 Java 程序被按照传统方式解释为字节码,还是被动态编译为可执行代码,其功能是相同的。

1.2 Java 虚拟机 JVM

Java 虚拟机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它电子设备中)安全并且兼容的执行保存在. class 文件中的字节码。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 程序的跨平台主要是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(Java 命令)负责将字节码文件解释成为特定的机器码进行运行。

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，避免了内存的泄露，也减轻了程序员的工作量。

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(. class 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(byte code 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 程序中写本机文件系统的操作；如果字节码校验器检验通过，由 Java 解释器负责把该类文件解释成为机器码进行执行。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。

1.3 JDK

1.3.1 认识 JDK

JDK(Java Development Kit，Java 开发包，Java 开发工具)是 Sun Microsystems 针对 Java 开发的产品，是一个写 Java 的 applet 和应用程序的程序开发环境。它由一个处于操作系统层之上的运行环境还有开发者编译，调试和运行用 Java 语言写的 applet 和应用程序所需的工具组成。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。

JDK 是一切 Java 应用程序的基础，所有的 Java 应用程序是构建在这个之上的。它是一组 API，也可以说是一些 Java Class。JDK 是许多 Java 专家最初使用的开发环境，尽管许多编程人员已经使用第三方的开发工具，但 JDK 仍被当作 Java 开发的重要工具。

JDK 中还包括完整的 JRE(Java Runtime Environment，Java 运行环境)，也被称为 private runtime。包括了用于产品环境的各种库类，以及给开发员使用的补充库。JDK 中还包括各种例子程序，用以展示 Java API 中的各部分。

从初学者角度来看，采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的关系，有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java 版本的升级而升级。但它的缺点也是非常明显的就是从事大规模企业级 Java 应用开发非常困难，不能进行复杂的 Java 软件开发，也不利于团体协同开发。

JDK 一般有三种版本：

(1)SE(J2SE)，standard edition，标准版，是我们通常用的一个版本。

(2)EE(J2EE)，Enterprise edition，企业版，使用这种 JDK 开发 J2EE 应用程序。

(3)ME(J2ME)，micro edtion，主要用于移动设备、嵌入式设备上的 Java 应用程序。

作为 JDK 实用程序，工具库中有七种主要程序。

(1)Javac:Java 编译器，将 Java 源代码转换成字节码。

(2)Java:Java 解释器，直接从类文件执行 Java 应用程序字节代码。

(3)appletviewer:小程序浏览器，一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。

(4)Javadoc:根据 Java 源码及说明语句生成 HTML 文档。

(5)Jdb:Java 调试器，可以逐行执行程序，设置断点和检查变量。

(6)Javah:产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件。

(7)Javap:Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。

1.3.2 下载 JDK

接下来我们来学习如何搭建 Java 程序开发环境。这里用到的 Java 程序开发环境由Java开发工具包 JavaSE Development Kit 7.0 及 Java 程序设计集成环境 Eclipse 组成。

打开浏览器，在地址栏内输入 JDK 下载网址：

http://www.oracle.com/technetwork/Java/Javase/downloads/index.html

下载最新版本的 JDK，如图 1-1 所示。在下载页面中，选择合适的版本，点击其链接，将其下载到本地。可以在百度里输入“JDK”获得完整下载地址。

Java SE Development Kit 7u17

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	106.65 MB	jdk-7u17-linux-i586.rpm
Linux x86	92.97 MB	jdk-7u17-linux-i586.tar.gz
Linux x64	104.78 MB	jdk-7u17-linux-x64.rpm
Linux x64	91.71 MB	jdk-7u17-linux-x64.tar.gz
Mac OS X x64	143.78 MB	jdk-7u17-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.39 MB	jdk-7u17-solaris-i586.tar.Z
Solaris x86	91.67 MB	jdk-7u17-solaris-i586.tar.gz
Solaris SPARC (SVR4 package)	135.92 MB	jdk-7u17-solaris-sparc.tar.Z
Solaris SPARC	95.32 MB	jdk-7u17-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	jdk-7u17-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.59 MB	jdk-7u17-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	22.61 MB	jdk-7u17-solaris-x64.tar.Z
Solaris x64	15.02 MB	jdk-7u17-solaris-x64.tar.gz
Windows x86	88.75 MB	jdk-7u17-windows-i586.exe
Windows x64	90.42 MB	jdk-7u17-windows-x64.exe

图 1-1 JDK 的下载

这里作为开发人员，我们选择 JDK 而不是 JRE(JDK 里包含 JRE)，因此用鼠标点击 JDK 下面的 DOWNLOAD 按钮，进入新的网页。要下载相应版本必须接受相应的许可协议，缺省情况下是不接受相应的许可协议。必须先点击左边的单选按钮“Accept License Agreement”，表示接受相应的许可协议，才能下载。

这里有不同平台的版本可供下载，对于 Windows 平台，有 32 位和 64 位两种，根据自己电脑的 Windows 平台的版本进行相应选择，如果用的是 64 位 Windows 7 系统，应选择下载

jdk-7u17-windows-x64. exe，如果电脑的 Windows 系统是 32 位，则应选择下载 jdk-7u17-windows-i586. exe。

1.3.3 安装 JDK

如果下载的是 exe 可执行程序，则这是一个安装程序，只需要运行该程序进行安装就行了，当然安装过程中可以指定安装的路径。如果下载的不是可执行程序，而是一个压缩文件，一般是 zip 文件，例如早期的 JDK 版本。如果是这种情况，只需要把压缩文件解压到自己所希望的 Java 安装目录下，然后设置环境变量就行了。Windows 平台上的解压缩工具很多，如 WinRAR、7-Zip 等，根据自己的喜好程度自己选择就是了。

以 exe 文件为例，双击我们刚刚下载的可执行文件，进行 JDK 的安装。随后弹出 JDK 安装向导，它将协助你一步步完成整个 JDK 的安装过程，如图 1-2。在安装过程中，须要注意如图 1-3 所示界面，正确选择 JDK 安装路径及要安装的内容。

图 1-2 JDK 的安装

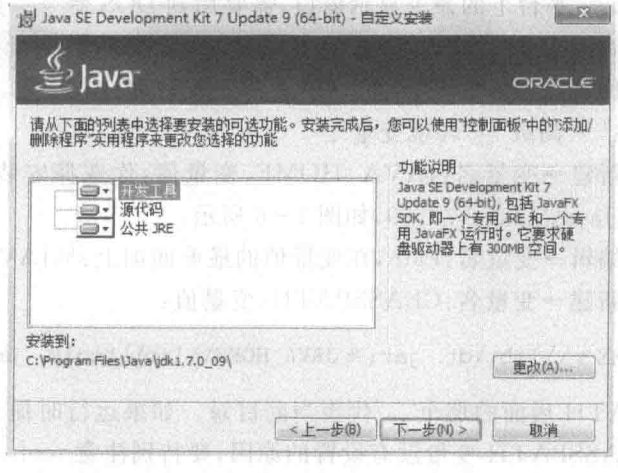

图 1-3 JDK 安装路径选择

安装系统提示 JDK 安装成功后，点击"完成"按钮退出 JDK 安装界面，此时 JRE 的安装界面会自动弹出，同样须要注意其安装路径，最好和刚刚安装的 JDK 放在同一目录下。当安装向导再次提示安装完成后，整个 JDK 的安装才算真正完成。

1.3.4 设置环境变量

安装 JDK 后运行控制台程序 cmd. exe，如图 1-4 所示。

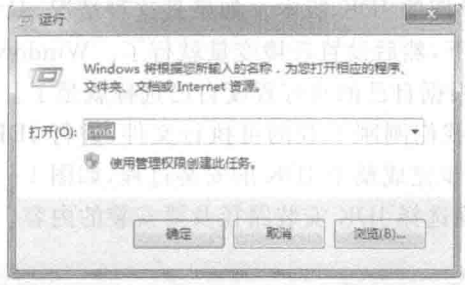

图 1-4 运行控制台程序 cmd. exe

鼠标点击"确定"按钮后 cmd. exe 程序会被启动，输入 Java - version 后按回车键，检查 Java 的环境变量是否配置成功，如果成功，会显示出相应的 Java 版本，笔者电脑上 Java 的版本为 1.7.0_05，如图 1-5 所示：

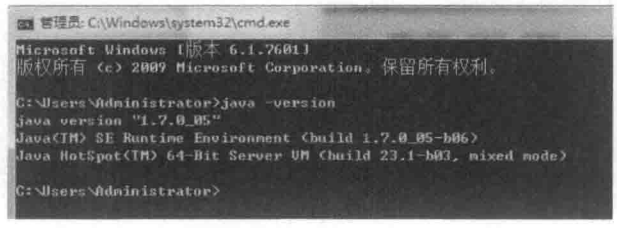

图 1-5 Java 环境变量配置成功

此窗口为 Windows 平台下的命令显示窗口，需要用到 DOS 命令。如果不熟悉，建议同学们学习一下 DOS 命令。如果输入 Java 后不成功，通常是环境变量设置不正常。现在 Windows 平台的 JDK 都是做好的安装包，一般正常安装后都没有问题。回到桌面，鼠标右击"我的电脑"→"属性"→"高级"→"环境变量"。

(1)系统变量→新建→变量名：JAVA_HOME，变量值：你选择安装 Java 的目录，例如 C:\\ProgramFiles\\Java\\jdk1.7.0_09，如图 1-6 所示。

(2)系统变量→编辑→变量名：Path，在变量值的最前面加上：%JAVA_HOME%\\bin;

(3)系统变量→新建→变量名：CLASSPATH，变量值：

```
%JAVA_HOME%\\lib\\dt. jar;%JAVA_HOME\\lib\\tools. jar;
```

说明：CLASSPATH 后面的那个". "代表当前目录。如果运行时提示"找不到主类或无法加载"，可能是 CLASSPATH 变量没有设置的原因，要特别注意 tools. jar 后边是分号。

到此为止，Windows 平台的 JDK 就安装好，可以进行简单的 Java 程序开发了。