

十一五规划计算机主干课程辅导丛书

Java语言 习题与解析

朱福喜 编著

Exercise

&
Analysis



清华大学出版社

► 十一五规划计算机主干课程辅导丛书

Java 语言习题与解析

朱福喜 编著

清华大学出版社

北 京

内 容 简 介

本书是作者在多年讲授 Java 语言与面向对象程序设计的基础上编写的,与作者编写的《Java 语言程序设计》一书配套使用。

全书共 9 章,结合俄罗斯方块这款经典游戏,讲解了 Java 语言基础、类和对象、继承与多态、图形用户界面、流和文件、多线程与异常处理、Applet 程序设计、网络编程等内容。每章先从涉及的主要概念、使用语法、编程技巧和实例分析入手,然后给出与本章内容密切相关的题目,并对题目进行了解答和分析。题型丰富,包括选择题、填空题、简答题和编程题,题目实用有代表性,其中不少题目来自于 Java 认证考试和某些公司的招聘考题。

本书既可以作为大专院校有关专业 Java 语言程序设计课程的教学辅导书,也可以作为该课程的自学指导书,还可以作为计算机水平考试、Java 认证考试、择业考试和考研的参考书。

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。

本书防伪标签采用特殊防伪技术,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 语言习题与解析/朱福喜编著. —北京:清华大学出版社, 2006

(“十一五”规划计算机主干课程辅导丛书)

ISBN 7-302-13858-3

I. J… II. 朱… III. Java 语言—程序设计—高等学校—解题 IV. TP312-44

中国版本图书馆 CIP 数据核字(2006)第 113960 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社总机: 010-62770175

客户服务: 010-62776969

组稿编辑: 夏非彼

文稿编辑: 刘秀青

封面设计: 林陶

版式设计: 科海

印 刷 者: 北京科普瑞印刷有限责任公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 22.75 字数: 554 千字

版 次: 2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

书 号: ISBN 7-302-13858-3/TP·8338

印 数: 1~5000

定 价: 32.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010) 82896445

丛 书 序

“习题与解析”系列计算机主干课程辅导丛书自 1999 年推出以来,一直被许多院校师生采用并受到普遍好评,其中也有很多人给我们反馈了不少中肯的改进建议。同时,随着计算机科学与技术的持续发展和演化,传统的计算机专业教学模式也在不断扩充与革新。在计算机教学和教材改革的不断深化过程中,如何促进学生将理论用于实践以及通过实践加深对理论的理解,如何提高其分析问题与动手解决实际问题的能力,都是我们 21 世纪计算机教育亟待解决的问题。

正是基于这样的需求,我们组织专家和一线教师对 2004 年修订的第 2 版“习题与解析”丛书的使用情况进行了深入调研,并对自身教学经验进行认真总结提炼之后,再次修订并隆重推出了这套“十一五规划计算机主干课程辅导丛书”。

丛书组成与新增内容

本套丛书根据计算机专业普遍采用的课程体系,在原丛书的基础上,针对教师备课、出题和学生考研需要,新增了《C 语言习题与解析》(B 级)、《数据结构习题与解析》(B 级)和《计算机组成原理习题与解析》(B 级)等专项分册,并将原先与之对应的各课程升级后称之为 A 级;其他课程根据各自的版本号依次升级。升级的原则是,依据各门课程的最新教学大纲,对原有图书内容进行全面的修订和扩充,使其更加完备、充实。升级之后的新版丛书几乎囊括了计算机专业的各个科目,且与当今计算机专业课程体系更加吻合。

丛书特色

本丛书除保留原丛书的体例风格外,还强化了如下几个方面的内容:

☒ 题型完整,内容丰富

从培养学生扎实的基本功入手,通过选择题、填空题、简答题、算法设计题和综合设计题等各种题目的分析讲解,巩固学生课堂学习的基本概念,扩展教材中讲授的知识,弥补课堂教学中因学时紧张带来的不足,从而帮助学生掌握学习各课程知识的基本技巧。

☒ 以典型题目分析带动能力培养

本丛书注重以典型题目的分析为突破口,点拨解题思路,强化各知识点的灵活运用,启发解题灵感。所有例题不仅给出了参考答案,还给出了详细透彻的分析过程,便于读者在解题过程中举一反三,触类旁通,从而提高分析问题和解决问题的能力。

☑ 全面复习，形成知识体系

本丛书以权威教材为依托，对各知识点进行了全面、深入的剖析和提炼，构成了一个完备的知识体系。在各类考试中，一个微小的知识漏洞，就有可能造成无法弥补的损失，因此复习必须全面扎实。本丛书以基本知识点归纳和例题分析为框架，提纲携领地为考生全程领跑。

☑ 把握知识点间的内在联系，拓展创新思维

把握知识点之间的关系，将掌握的知识变“活”。本丛书通过对知识点的分解，找出贯穿于各知识点之间的内在联系，并配上相关的例题，阐明如何利用这些内在联系解决问题，从而做到不仅授人以“鱼”，更注重授人以“渔”。本套丛书由长期坚持在教学第一线的教授和副教授编写，他（她）们结合自己的教学经验和见解，把多年的教学实践成果无私奉献给读者，希望能够提高学生素质，培养学生的综合分析能力。

☑ 扩充三门核心课程的B级版本，提高学习的针对性

“C 语言程序设计”、“数据结构”和“计算机组成原理”是计算机科学及相关专业的考研课程，考生学习难度更高。本丛书的B级以考研要求为依据，进一步细化各章的知识点，将相关知识点归类，并进行相应的例题剖析；特别是针对近两年研究生入学考试的真题和典型难解题，增加了“知识创新”和“创新应用”部分，前者是作者在总结相关知识点的基础上归纳总结的解题新方法，后者是应用这些新方法去解决问题的过程。因此，B级更适合考研者和相关任课教师参考。

读者对象

本套丛书针对本科院校计算机及相关专业的学生作为教学参考书，也适合广大计算机专业本科生作为考研的学习与辅导教材。

如果说科学技术的飞速发展是21世纪的一个重要特征的话，那么，教学改革将是21世纪教育工作不变的主题，也是需要我们不断探索的课题。紧跟教学改革，不断更新教学内容和教材，真正满足新形势下的教学需求，还需要我们不断地努力实践和完善。本套丛书虽然经过细致地编写与审校，以及几个版次的修订，但仍然难免还有疏漏和不足之处，需要我们不断地补充、修订和完善。我们热情欢迎使用本套丛书的教师、学生和读者朋友提出宝贵意见和建议，使之日臻成熟完善。

本丛书作者的电子邮件：licb@public.wh.hb.cn

本丛书出版者的电子邮件：chiefeditor@khp.com.cn

2006年9月

前 言

Java 自 1995 年发布以来,已成为现今 IT 业界的主流编程语言。各高等院校的计算机和非计算机专业都相继开设了 Java 程序设计课程。

Java 是一种经典的面向对象的程序设计语言,可以为面向对象技术提供全面支持。同时,它又是一种完美而实用的网络编程语言,能够满足迅猛发展的 Internet 和 WWW 编程的需要。Java 带来了 IT 业的蓬勃发展和繁荣,也为众多的学子打开了就业的大门。

本书从 Java 的基本概念入手,介绍了面向对象程序设计的主要概念和方法,并通过实例分析,演示 Java 语言的语法功能、程序设计思路及编程技巧,使读者能掌握其原理、方法与考试技巧,同时具备面向对象的设计和项目开发能力,为各种认证考试、择业考试打下基础。书中列举的大量程序实例都是经过精心挑选,既能阐明原理和方法,又能保证有一定的实用性,同时也能确保一定的广度和深度,其中有些题目来自于 Java 认证考试和 IT 公司的招考题。此外,全书以一个经典游戏——俄罗斯方块为例,来分析用到的 Java 语言知识点,以帮助读者提高学习兴趣和项目开发能力。

本书第 1 章主要介绍 Java 开发工具的下载和环境的安装,也对 Java 程序的调试和程序文档的建立做了介绍;第 2 章介绍了 Java 语言的基础知识,包括 Java 的数据类型、变量、表达式和循环控制语句的用法。第 3~4 章介绍了 Java 面向对象程序设计知识,围绕类和接口详细地讨论了面向对象技术的封装、抽象、继承和多态等特征以及它们在面向对象程序设计中的具体应用,通过实例展示 Java 语言和面向对象程序设计的精髓;第 5 章介绍 Java 图形用户界面的设计和编程实现,帮助读者编写出丰富多彩的程序界面,开发具有专业水平的应用程序外观;第 6 章介绍流和文件,阐述 Java 输入输出操作的原理;第 7 章介绍 Java 的多线程编程和异常处理,帮助读者掌握编写多线程程序的异常处理程序的技巧,保障所开发程序的强壮性;第 8 章介绍 Applet 的设计,充分展现 Java 语言在 Web 上的魅力,特别是多媒体方面的表现力;第 9 章介绍网络编程,这一章对原教材上的内容进行了扩展,进一步展示了 Java 的强大网络编程功能和读者在网络编程方面应该掌握的技巧。

本书既可以作为大专院校有关专业 Java 语言程序设计课程的教学辅导书,也可以作为该课程的自学指导书,还可以作为计算机水平考试、Java 认证考试、择业考试和考研的参考书。读者可到网站 www.khp.com.cn 下载俄罗斯方块实例的完整代码,以免去输入代码的劳累。

在本书的编写过程中,朱丽达、白圣猛、唐晓军、邓炜、田明等验证了本书的所有实例,在此向他们表示衷心感谢。同时,本书的编写得到了清华大学出版社夏非彼老师的大力支持,在此谨向她表示衷心感谢。

由于时间和水平所限,难免有错,恳请读者批评指正,使本书得以改进和完善。

E-mail: lxq@khp.com.cn

编 者

2006 年 9 月

目 录

第 1 章 Java 环境配置与简单编程	1
1.1 基本知识点	1
1.1.1 JDK 的下载、安装与环境配置	1
1.1.2 Java 编译器与 Java 虚拟机	3
1.1.3 独立应用程序与 Applet 程序	4
1.1.4 建立文档说明	5
1.2 实例分析	5
1.2.1 从编辑到执行一个应用程序的完整过程	5
1.2.2 简单的键盘输入类	9
1.2.3 简单的 Applet 类的实例	11
1.2.4 测试和代码调试	12
1.2.5 程序文档建立的完整过程	17
1.3 习题与解答	18
1.3.1 选择题	18
1.3.2 填空题	20
1.3.3 简答题	21
1.3.4 编程题	22
第 2 章 Java 语言基础	25
2.1 基本知识点	25
2.1.1 基本数据类型	25
2.1.2 变量的声明和赋值	26
2.1.3 数值类型的运算	27
2.1.4 快捷运算符和类型转换	28
2.1.5 逻辑运算符和比较运算符	29
2.1.6 条件执行语句	29
2.1.7 循环控制语句	31
2.1.8 字符串操作	33
2.1.9 数组	34
2.2 实例分析	36
2.2.1 俄罗斯方块介绍	36
2.2.2 数组作为方法的输入参数举例	40



2.3 习题与解答	41
2.3.1 选择题	41
2.3.2 填空题	48
2.3.3 简答题	56
2.3.4 编程题	57
第3章 面向对象程序设计基础	69
3.1 基本知识点	69
3.1.1 面向对象程序设计的基本概念	69
3.1.2 类与对象	69
3.1.3 创建对象和构造方法	70
3.1.4 finalize 方法与垃圾回收	71
3.1.5 成员方法的定义	72
3.1.6 方法的参数传递	73
3.1.7 变量的作用域	74
3.1.8 成员变量与成员方法的访问	75
3.1.9 包	75
3.1.10 private、public 和 protected 修饰符	76
3.1.11 静态成员与非静态成员	77
3.1.12 重载	78
3.2 实例分析	78
3.2.1 一个 Length 类的完整创建过程	78
3.2.2 静态成员变量与成员变量举例	84
3.2.3 重载成员方法举例	87
3.3 习题与解答	88
3.3.1 选择题	88
3.3.2 填空题	98
3.3.3 简答题	103
3.3.4 编程题	106
第4章 面向对象程序设计	113
4.1 基本知识点	113
4.1.1 继承	113
4.1.2 覆盖	114
4.1.3 抽象	115
4.1.4 Java 的基类 Object	116
4.1.5 接口	117
4.1.6 多态性	118
4.2 实例分析	118

4.2.1 俄罗斯方块游戏中的类的继承关系	118
4.2.2 俄罗斯方块游戏中的 ErsBox 类	119
4.2.3 继承的使用举例	121
4.2.4 覆盖的使用举例	124
4.2.5 抽象类和抽象方法的使用举例	124
4.2.6 用 3 种方式实现累计多种形状的面积	127
4.3 习题与解答	131
4.3.1 选择题	131
4.3.2 填空题	138
4.3.3 简答题	144
4.3.4 编程题	146
第 5 章 用户界面设计	158
5.1 基本知识点	158
5.1.1 抽象窗口工具 (AWT)	158
5.1.2 基本窗口类	158
5.1.3 简单的 GUI 输入输出组件	159
5.1.4 事件响应	160
5.1.5 事件处理	160
5.1.6 布局管理	162
5.1.7 Menu, MenuBar 和 MenuItem 类	163
5.1.8 图形类与图形的绘制	163
5.1.9 为什么要使用 Swing	164
5.2 实例分析	165
5.2.1 主类 ErsBlocksGame	165
5.2.2 控制面板类 ControlPanel	174
5.2.3 游戏显示窗口 GameCanvas	179
5.3 习题与解答	182
5.3.1 选择题	182
5.3.2 填空题	187
5.3.3 简答题	191
5.3.4 编程题	193
第 6 章 流和文件	208
6.1 基本知识点	208
6.1.1 数据流	208
6.1.2 字节级输入输出类	209
6.1.3 字符级输入输出类	210
6.1.4 对象级输入输出类	210

6.1.5	文件操作	211
6.1.6	字符与字节流转换.....	211
6.1.7	系统 I/O 流	212
6.1.8	RandomAccessFile.....	212
6.2	实例分析	213
6.2.1	标准输入输出以及重定向的举例.....	213
6.2.2	读文件的举例.....	215
6.2.3	序列化输入输出的举例.....	218
6.3	习题与解答	224
6.3.1	选择题	224
6.3.2	填空题	228
6.3.3	简答题	231
6.3.4	编程题	233
第 7 章	多线程与异常处理.....	245
7.1	基本知识点	245
7.1.1	多线程	245
7.1.2	异常处理机制.....	248
7.2	实例分析	251
7.2.1	一个用两种方式实现线程的程序.....	251
7.2.2	通过继承 Thread 类实现俄罗斯方块.....	252
7.2.3	通过 runnable 接口实现游戏类.....	256
7.3	习题与解答	258
7.3.1	选择题	258
7.3.2	填空题	265
7.3.3	简答题	272
7.3.4	编程题	276
第 8 章	Applet	290
8.1	基本知识点	290
8.1.1	Applet 的基本概念.....	290
8.1.2	Applet 标记.....	291
8.1.3	利用标记参数向 Applet 传递信息	291
8.1.4	Applet “沙盒”	292
8.1.5	统一资源定位器 URL.....	292
8.1.6	向 Applet 中载入数据.....	293
8.1.7	在 Applet 中读文件的一般规则	293
8.2	实例分析	294
8.2.1	一个既可作 Applet 又可作应用程序的例子	294

8.2.2	Applet 读取文件的一般方法举例	297
8.2.3	Applet 与线程结合读取文件的举例	299
8.3	习题与解答	303
8.3.1	选择题	303
8.3.2	填空题	305
8.3.3	简答题	308
8.3.4	编程题	309
第 9 章	Java 网络编程	325
9.1	基本知识点	325
9.1.1	Java 对网络应用的支持	325
9.1.2	Java 的客户机/服务器应用模式	326
9.1.3	用 Socket 方式实现通信	326
9.1.4	用 Datagram 方式实现通信	327
9.2	实例分析	329
9.2.1	用 Socket 实现客户服务模式的实例	329
9.2.2	用 Datagram 方式实现的客户服务实例	332
9.3	习题与解答	336
9.3.1	选择题	336
9.3.2	填空题	337
9.3.3	简答题	340
9.3.4	编程题	343
参考文献		351

第 1 章 Java 环境配置与简单编程

本章学习要点

- ☒ JDK 的下载、安装与环境配置
- ☒ 了解 Java 虚拟机的特点以及 Java 语言的特点
- ☒ 掌握编译和执行 Java 程序的基本步骤
- ☒ 理解 Java 程序的入口并能够通过命令行传递参数
- ☒ 会编写通过键盘输入数据的简单 Java 程序
- ☒ 会编写简单的 Applet 程序
- ☒ 会使用浏览器和 appletviewer 运行 Applet 程序
- ☒ 会调试简单的 Java 程序
- ☒ 会书写必要的注释并建立程序文档

1.1 基本知识点

1.1.1 JDK 的下载、安装与环境配置

JDK 是 Java 开发工具。JDK 工具包包含了编译和运行 Java 程序所需的 Java 编译器和 Java 虚拟机, 以及大量的文档和对开发 Java 程序非常有用的类库。在 Sun 公司的网站可以找到 JDK 的最新版本, 能自由下载。当前 JDK 的最高版本是 JDK 1.5, 它的下载网址是:

<http://java.sun.com/j2se/1.5.0/download.jsp>

因为有可能随着时间的变化, 下载 JDK 的网址有所变化, 但我们只要在如图 1.1 所示的 Sun 公司的网站中找到 Downloads 的页面, 就可以找到下载 JDK 的具体位置。

这里我们要下载支持 Windows XP/2000/NT 操作系统的版本, 下载的文件名类似于 jdk-1_5_0-windows-i586.exe。当然, 也可能由于版本的差异, 文件名略有不同。

对于在 Windows 环境下 JDK 的安装, 直接双击所下载的 jdk-1_5_0-windows-i586.exe 文件的图标, 即可运行该文件。因为它是一个自解压的 EXE 文件, 解压后, 自动进入安装过程, 此时可以按照提示逐步完成安装。安装时可以自选一个比较简单的目录, 例如, C:\JDK15, 避免拼写复杂的目录名给操作带来的很多不便。

如果在低版本的 Windows 操作系统下, 可直接编辑自动批处理文件 Autoexec.bat 文件, 并在该文件中添加如下设置语句:



```
set JAVA_HOME=C:\jdk15
set classpath = .; %JAVA_HOME%\LIB\dt.jar; %JAVA_HOME%\LIB\TOOLS.jar;
path=%path%; %JAVA_HOME%\bin; %JAVA_HOME%\jre\bin
```

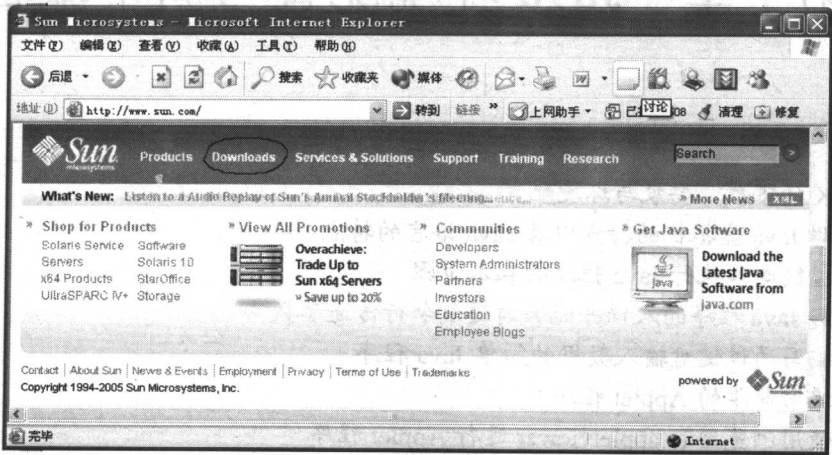


图 1.1 Sun 的网站及下载区

在 Windows 2000/XP 下，环境变量的配置可以在“我的电脑”图标上单击右键，选择“属性”选项，打开“系统属性”对话框，如图 1.2 所示，然后选中“高级”选项卡，再单击“环境变量”按钮，打开“环境变量”对话框，如图 1.3 所示。

单击“系统变量”选项组的“新建”按钮，然后在“变量名”文本框中输入“JAVA_HOME”，在“变量值”文本框中输入：c:\jdk15，然后单击“确定”按钮；再次单击“新建”按钮，在“变量名”文本框中输入“classpath”，在“变量值”文本框中输入“.; %JAVA_HOME%\LIB\dt.jar; %JAVA_HOME%\LIB\TOOLS.jar”，单击“确定”按钮；接着修改“系统变量”选项组的 path 变量，将其值加上“%JAVA_HOME%\bin; %JAVA_HOME%\jre\bin”。最后单击“环境变量”对话框的“确定”按钮。

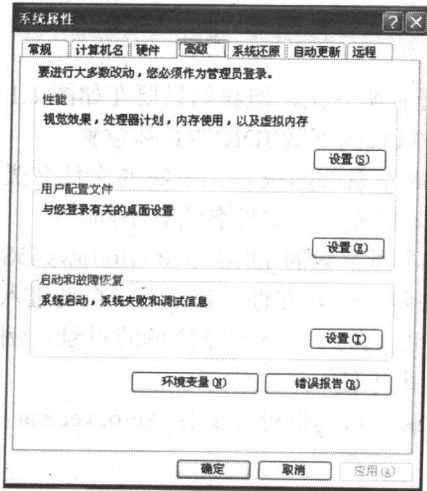


图 1.2 “系统属性”对话框

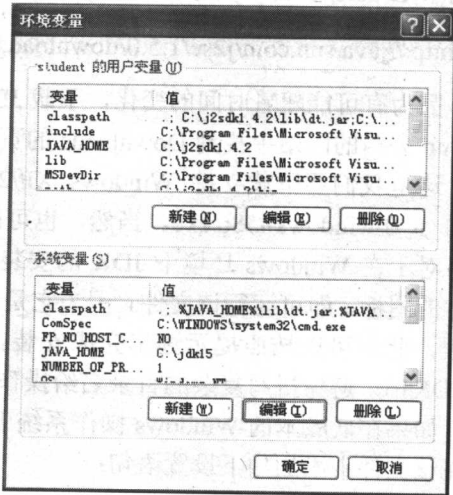


图 1.3 “环境变量”对话框

配置完成后，需重新启动计算机，环境变量方能生效。

1.1.2 Java 编译器与 Java 虚拟机

Java 编译器

一个 Java 源代码文件必须经过转换才能被计算机识别和执行，该转换过程被称为编译，完成这个转换工作的就是 Java 编译器。在 JDK (Java Developer's Kit) 中，编译工具是 `javac`。Java 源程序编译后得到的文件被称为类文件，类文件名与源代码文件相同，后缀为“.class”。

在 Windows 和 Unix 操作平台下，直接在命令行方式下输入“`javac FileName.java`”，就可以生成类文件。

Java 虚拟机 (JVM)

Java 虚拟机 (Java Virtual Machine) 是一个想像中的机器，用来运行 Java Applet 和 Java 应用程序的引擎，是在实际的计算机上通过软件模拟来实现。Java 虚拟机有自己想像中的硬件，如处理器、堆栈、寄存器等，还具有相应的指令系统。它具有平台无关性，这是因为它能理解与平台相关的文件格式，并能在不同的平台上运行由编译器产生的类文件，在一个平台上生成的类文件就能（借助 Java 虚拟机）在不同的平台上运行。

字节码是 Java 虚拟机所支持的指令，它是与机器无关的代码，由 Java 编译器生成，Java 解释器执行。每个字节码完成一种基本运算，例如，把一个数加到寄存器，子程序转移等。Java 虚拟机支持大约 248 个字节码。Java 指令集相当于 Java 语言的汇编语言。

如果我们把计算机硬件到用户应用程序之间划分为若干层次，那么 Java 虚拟机所处的层次如图 1.4 所示。

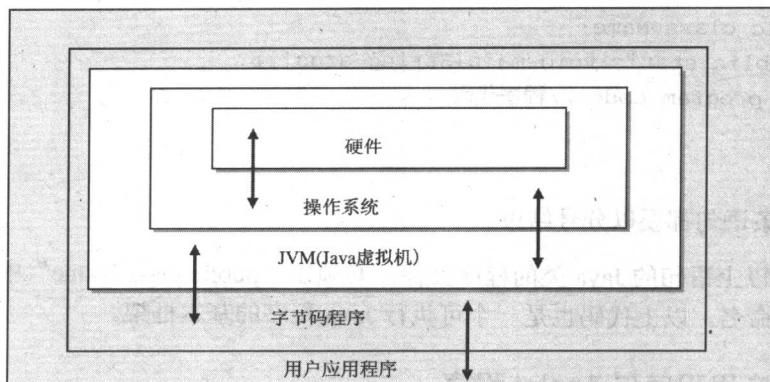


图 1.4 Java 虚拟机在一个完整的 Java 应用中所处的层次

从图 1.4 可以看出，Java 虚拟机是操作系统和 Java 程序之间的一个中间层。我们常说一个 Java 程序可以不用修改就可以在不同的系统上运行，而事实上，Java 程序只在一个系统上运行，那就是 Java 虚拟机。Java 虚拟机可以在不同的操作系统上运行。显然，在不同操作系统上的 JVM 大不相同，但它们为 Java 程序员提供了统一的调用操作系统的接口界面。

由于在一个 Java 程序中可以集成多线程、图形用户界面、多媒体与网络操作等功能，Java 虚拟机就必须对每个细节在操作系统和类文件之间进行协调。从理论上讲，对于任何



操作系统，Java 虚拟机都会产生相同的结果。图 1.5 描述了 Java 程序与编译和运行它的环境的相互关系。

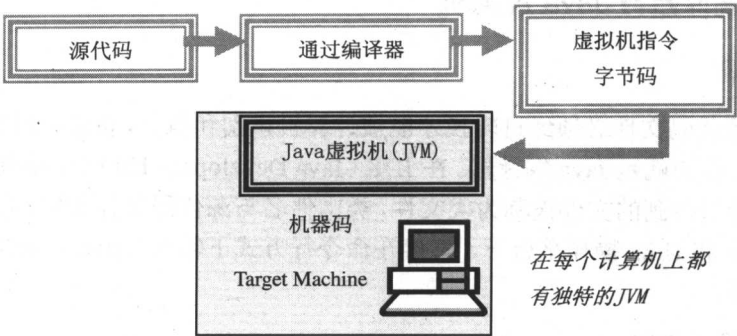


图 1.5 Java 虚拟机在一个完整的 Java 应用所处的层次

另外，当调用 Java 虚拟机运行一个指定的类文件时，它首先会有一个确认过程来确保系统的安全性，然后再执行类文件的指令。

Java 程序设计基本规则

Java 的源代码可以用任何一种文本编辑器编辑，只要符合 Java 语言的语法要求就行。下面是编写 Java 程序的一些最基本规则。

- Java 程序区分大小写。如 Java 和 java 是两个不同的字符串。
- 花括号用来将一组语句聚在一起。
- 一个可执行程序必须至少包含如下的语句：

```
public class Name
{ public static void main(String args[])
{ program code //程序代码
}
}
```

- 每一条语句都要以分号结束。

一个包含以上语句的 Java 类的程序文件，必须以“public class Name”中的“Name”加上“.java”命名。以上代码也是一个可执行 Java 程序的基本框架。

1.1.3 独立应用程序与 Applet 程序

Java 程序分为两种类型：独立应用程序(stand-alone application)和小应用程序(Applet)。独立应用程序编译后由虚拟机直接解释执行，Java JDK 就是源代码编写、编译、执行的集成环境；对于 Applet，需用 Java 编译器先将它编译成 .class 的代码，然后再嵌入到某个 HTML 文件中，由某个浏览器解释执行。Applet 的嵌入可以增加 WWW 页面的交互性和动态性。

Applet 程序必须扩展 Applet 类，而扩展 Applet 类的程序没有构造方法，也不需要 main()

方法。一个 Applet 程序一般包含 4 个重要方法：init()、start()、stop()和 destroy()。Applet 的执行由 init()方法控制，Applet 程序装载后，init()方法只执行一次；而每次重新访问 Applet 时，则要调 start()方法；每次 Applet 所在的 Web 页面被其他页面所取代时，则要调 stop()方法；destroy()方法完成撤销清理工作并准备下载。

1.1.4 建立文档说明

Java 允许我们给源代码添加文档说明，JDK 开发包中的 javadoc 工具可以对其进行编辑和排版。该工具可以按如下规则在程序的源代码中加入一些指定的标签式注释，并将其处理、排版、索引、链接成 HTML 文档：

- 文档说明开始于符号 “/**”，结束于符号 “*/”
- 文档说明可以包括 HTML 格式指示（除了标题标签）
- 文档说明加在它们所指的类名、变量或方法之前
- 文档说明可以包括特殊的说明行：

@author Name	用于说明代码的作者
@version Number/Date	用于指示版本号和代码完成日期
@see class#references	用于链接其他的文档说明

- 文档说明可以包括一些方法的说明：

@return rType	用于说明方法的返回类型
@exception exType	用于说明方法所抛出的意外

按照软件工程的一般方法，一个正确的程序或类应该包括文档说明。在 Java 中最少应该做到如下几点：

- 使用 javadoc 生成的文档说明给出整个类的作用的简明描述
- 通过 @author 标签包括作者名
- 通过 @version 标签包括版本号和代码日期
- 使用 javadoc 给类中的每个方法生成简明描述，指出方法的所有输入参数和返回类型；若方法在其输入参数上有特殊要求，也要给予说明。

1.2 实例分析

1.2.1 从编辑到执行一个应用程序的完整过程

源代码文件

一个 Java 源代码文件是一个包含了 Java 程序代码的文本文件，其中的代码必须符合 Java 的编程规则。Java 源代码文件的命名格式应该是类名加一个后缀 “.java”，英文名称的



习惯做法是将名称的第一个字母大写。

下面的例子是实现一个简单的屏幕输出：“这是我的第一个程序”，它包含如下语句：

```
public class Test
{
    public static void main(String args[])
    {
        System.out.println("这是我的第一个程序");
    }
}
```

这个类的名称是 `Test`，所以必须保存为 `Test.java`。

虽然我们可以把几条语句写在一行上，但这并不值得提倡。一个好的 Java 程序应该做到使每一条语句的从属关系都很清楚，一般遵循如下惯例：

- 一段 Java 程序以一个花括号开始，并以一个花括号结束。
- 同一段 Java 程序中每行代码都是左对齐的。
- 程序中的代码段成锯齿形缩进格式排列。

编译一个 Java 程序

下面我们用 `javac` 命令将 `Test.java` 的代码编译成类文件。在 Windows 操作系统中，进入 DOS 命令方式，直接输入如下命令：

```
javac Test.java
```

Java 编译器首先检查源代码文件的语法是否正确，如果没有错误，它将产生名为 `Test.class` 的类文件。如果编译器发现了错误，它就列出相应的错误信息，包括出现错误的位置（第几行）和简短的错误说明。

Java 编译器发现错误后，此时要明确如下几个概念：

- 如果编译器指出一个错误，那么在该位置或之前必然存在产生错误的代码。
- 编译器指出的错误数，是产生错误的代码的最小数目。
- 当修正一个错误后，新的错误可能又会显现出来。

我们就可以根据错误信息对源代码文件进行相应的修改，然后再重新编译，直到确认这几个错误完全被改正过来。如果在编译器指定的位置不能发现错误，那么就在这个位置之前的执行代码中寻找。

运行一个 Java 类文件

JDK 提供了一个“`java`”执行文件来运行 Java 编译后的类文件。调用时，类文件作为 `java` 命令的参数。一个 Java 程序的运行过程为：

- 在某台计算机中安装 JDK 后，其 `bin` 目录下的 `java.exe` 即为虚拟机（JVM）
- Java 虚拟机（JVM）对类文件进行定位，然后读入
- Java 虚拟机（JVM）对类文件的安全违例进行检查
- Java 虚拟机（JVM）将执行（或解释）类文件的指令

在 Windows 的 DOS 界面下，只需输入命令：