

■ 现代服务业系列实验教材 ■

Java 与面向对象 程序设计实验教程

■ 雷 擎 主编



对外经济贸易大学出版社
University of International Business and Economics Press

现代服务业系列实验教材

Java 与面向对象程序 设计实验教程

雷 擎 主编

对外经济贸易大学出版社
中国·北京

图书在版编目 (CIP) 数据

Java 与面向对象程序设计实验教程 / 雷擎主编. —
北京: 对外经济贸易大学出版社, 2012
现代服务业系列实验教材
ISBN 978-7-5663-0313-4

I. ①J… II. ①雷… III. ①JAVA 语言 - 程序设计 -
教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 074631 号

© 2012 年 对外经济贸易大学出版社出版发行

版权所有 翻印必究

Java 与面向对象程序设计实验教程

雷 擎 主编

责任编辑: 赵 昕

对外经济贸易大学出版社

北京市朝阳区惠新东街 10 号 邮政编码: 100029

邮购电话: 010-64492338 发行部电话: 010-64492342

网址: <http://www.uibep.com> E-mail: uibep@126.com

北京市山华苑印刷有限责任公司印装 新华书店北京发行所发行

成品尺寸: 185mm × 230mm 17.5 印张 347 千字

2012 年 7 月北京第 1 版 2012 年 7 月第 1 次印刷

ISBN 978-7-5663-0313-4

印数: 0 001 - 3 000 册 定价: 32.00 元

本套教材出版受到以下项目资助：

北京市级现代服务业人才培养实验教学示范中心

北京市级商务信息管理系列课程优秀教学团队

国家级和北京市级电子商务特色专业

现代服务业系列实验教材

编委会成员名单

编委会主任：陈进 对外经济贸易大学

编委会副主任：（按姓氏笔画排序）

王学东	华中师范大学
刘 军	北京交通大学
祁 明	华南理工大学
孙宝文	中央财经大学
汤兵勇	东华大学
张 宁	北京大学
宋远方	中国人民大学
李 琪	西安交通大学
杨 鹏	华道数据处理有限公司
张念录	中国国际电子商务中心
陈德人	浙江大学
柴洪峰	中国银联股份有限公司
章 正	上海财经大学

编委会委员：（按姓氏笔画排序）

刘瑞林	对外经济贸易大学
沈 沉	对外经济贸易大学
赵星秋	对外经济贸易大学
黄健青	对外经济贸易大学
曹淑艳	对外经济贸易大学

总 序

现代服务业是依托于信息技术和现代管理理念而发展起来的知识和技术相对密集的服务业，具有应用信息技术和富于创新发展的主要特点。

现代服务业的发达程度是衡量经济、社会现代化水平的重要标志，是全面建设小康社会时期国民经济持续发展的主要增长点。发展现代服务业是实施国民经济可持续发展战略的需要和实现跨越发展的有效途径，也是调整我国经济结构、促进经济社会和人的全面发展、走向知识社会的必要条件。

近年来，我国十分重视现代服务业的发展，国家规划纲要明确指出坚持市场化、产业化、社会化方向，拓宽领域、扩大规模、优化结构、增强功能、规范市场，提高服务业的比重和水平。

现代服务业的快速发展对现人才培养提出了新的要求，需要大量既具有比较扎实的基础理论与知识水平，又具有比较强的动手能力与操作能力，能适应现代化服务业发展需要的素质高、技能强的服务业创新人才。

对外经济贸易大学现代服务业实验教学示范中心是北京市批准的教育教学质量建设项目，目前已经形成了实验教学的完整体系，开设了电子金融、电子商务、网络营销、ERP 与供应链管理、经营管理中的决策方法、网络实用技术与应用、外贸实训等多门实验课程和实验项目；建立了完整的实验教学资料库；并建立了包括基础实验、核心实验和特色实验的实践教学课程体系；构建了实验课程、科研项目与专业实习有机结合的实践方案和管理系统。

现代服务业系列实验教材是对外经济贸易大学在教育部和北京市质量工程建设过程中，经过总结、提炼、完善，形成的一套针对现代服务业人才培养的实验教材。教材主要目的是培养学生综合素质和实践能力，教材的编者都是具有丰富实验教学经验的教师，凝聚了教师们的心血和汗水。本系列教材面向现代服务业的管理和应用人才，以实践能力和技术应用能力为培养目标。

我们希望现代服务业系列实验教材在人才培养实验教学改革和教学实践过程中起到积极作用。

本套教材在编写的过程中广泛吸纳了众多师生的宝贵意见，同时也得到了对外经济贸易大学出版社的领导和编辑们的大力支持，对他们表示衷心的感谢。

《现代服务业系列实验教材》编委会

2012 年 3 月

前 言

1. 书的编写背景

Java 语言, 1995 年诞生于 Sun 公司。简单、面向对象、自动的内存管理、分布计算、稳定、安全、解释执行、结构中立、平滑移植、多线程及异常处理等多方面的优点, 使得 Java 语言从跨平台和动态页面显示的最初应用, 逐步扩展到的 Corba、Web 服务器后端处理、应用整合和移动服务等 IT 产业的更多应用领域。Java 本身也从一门编程语言发展成为一门技术, 包括软件设计模式、软件开发思想、软件体系架构、软件应用框架和软件支撑平台等。从某些意义上, Java 语言的产生对编程语言、软件工程和软件的发展产生了深远的影响。到目前为止, 国内外大学中与 IT 相关的专业, 大多数都把 Java 语言相关的课程设置为专业基础课或专业必修课。

在笔者多年 Java 语言的授课过程中发现, 在许多教材中, 为了说明一个知识点提供的程序示例代码通常比较长而且复杂, 常常影响到学生理解基本概念、基本思想和语法基础, 而小程序的实践编程可以充分激发他们的兴趣和想象, 帮助学生更加容易地学习 Java 语言。

本书是一本实践性非常强的 Java 语言实验教程, 是笔者对实际教学过程中的教学经验、示例程序以及学生练习实验的总结, 紧扣 Java 语言的基础教学, 注重基本概念和基础知识, 可以说是一个 Java 短小程序的集锦, 非常有针对性。本书可以作为 Java 语言授课教材的配套实验教材来使用, 内容章节的组织与大多数 Java 教材相符; 本书也可以作为实验教材独立使用, 或作为 Java 初学者自学的辅助教材使用。

2. 本书的特色

本书覆盖内容全面, 深入浅出, 实验设计与基本概念和基础知识结合紧密, 实验要求明确, 实验步骤详细清晰, 内容具有很强的实践性和针对性。本书具有以下特色。

(1) 遵循教学的特点和规律。在内容安排上将紧扣 Java 语言教学进行设计, 充分考虑老师的教学需求和学生的学习需要, 每个实验都提供实验目的、课时要求、实验内容和实验要求的详细描述, 为教师授课和学生自学提供参考。

(2) 紧扣 Java 语言的基础教学, 注重基本概念和基础知识。每个教学知识点对应一个实验步骤, 具有一个或几个简单的示例代码, 帮助学生完成实验要求。学生每独立完成一个步骤, 就能够完成一个教学知识点的学习, 初步理解和掌握相关的概念或语法。每一章的实验都由易到难, 循序渐进。

(3) 实验设计时知识点划分明确, 指导细致: 每一章的实验部分包括多个实验, 每个基本的知识点设计一个实验; 每个实验包括多个步骤, 分别针对基本知识点下相关的基本概念、基本语法或基本应用; 每个步骤下包括多个操作指导, 每个操作指导具体提示学生应该掌握的基本概念、基本技能和基本语法。

(4) 详细的代码注释: 在每个示例代码中, 凡是新出现或难的知识点, 均提供与所要掌握的知识对应的注释, 帮助学生理解示例代码。

(5) 实验环境和平台跟踪 Java 技术新发展: 本书所使用的示例代码都基于 JDK 1.6 和 Eclipse 3.7.1 简体中文版环境编写, 全部测试通过。

(6) 实验平台使用开源集成开发工具: 实验中基于的 Eclipse 平台是开源的绿色平台, 操作步骤和代码示例的讲解都结合 Eclipse 开发工具, 学生也不会受到软件版权等限制。

(7) 知识点的讲解简明扼要: 每一章在第一部分知识要点中, 只提供关键知识点的简要描述, 方便学生实践学习时的容易查找、参考。

(8) 实验设计灵活: 每个实验都是独立的示例代码, 相互之间没有过多的关联, 教师在指导中可以选择部分实验, 也可以改变介绍的顺序, 以适应学生和课程目标。

3. 本书主要内容

第 1 章“Java 概述”和第 2 章“Java 语言基础”着重指导学生进行 Java SDK 的安装, 使用 JDK 提供的命令进行最基本的 Java 应用程序创建、编译和运行, 配制环境变量, 进行 Eclipse 的安装和使用等基本技能实践; 着重指导学生练习使用 Java 语言的基础语法, 进行基本数据类型的使用和类型的转换, 以及注释的使用和生成 JavaDoc 等实践。这两章共 10 个实验, 需编写 20 多个短小程序。

第 3 章“类和对象”及第 4 章“继承与多态”两章是本书的重点, 也是篇幅最长的两章。这两章的目的是使学生理解面向对象程序设计的概念, 理解继承和抽象的概念, 理解现实问题与 Java 类的描述相互之间的关系, 掌握使用类描述事物属性和功能的方法, 掌握 Java 类的定义、初始化、访问控制语法, 掌握对象的创建和使用语法, 掌握子类的定义方法, 理解子类对父类成员的继承、覆盖和隐藏, 掌握子类构造方法的定义语法和父类构造方法的调用方法, 理解继承中构造方法链等知识。这两章共 19 个实验, 近 60 个细分知识点, 近 100 段代码, 帮助学生掌握 Java 语言最基础思想和概念。

第 5~8 章分别着重指导学生进行了 Java 异常机制处理、数组与集合的使用、多线程的使用和输入输出流的使用。这四章共 17 个实验, 近 40 个细分知识点, 需编写 40 多个短小程序。

第 9 章“用户图形界面”的目标主要是通过本章的实验, 使学生了解 JavaGUI 的技术, 理解 Applet、布局管理、事务处理的概念, 掌握利用 AWT 工具包和 Swing 工具实

现 Java 图形界面的语法, 掌握实现 Applet、布局管理、事务处理的语法。本章共提供了 6 个实验, 通过这些实验的练习, 学生能够掌握定义 Applet 和 Application 类型的 GUI 图形界面的基本方法和语法, 理解 Applet 的生命周期, 理解布局管理的概念, 掌握设置 BorderLayout、FlowerLayout 和 GridLayout 布局管理的语法, 理解事件和事件处理机制, 掌握实现单个事件处理、多个事件处理的方法, 掌握实用事件处理的三种方法: 实现监听器接口、定义内部类和定义匿名内部类, 掌握 Swing 的顶级容器、中间容器和组件的使用方法和语法。

第 10 章“网络应用”的实验, 主要针对套接字通信和 JDBC 的使用。本章共提供了 3 个实验, 指导学生学习使用套接字实现简单通信的服务器和客户端, 掌握配置 Eclipse 数据库访问环境, 掌握访问数据库的基本方法。

4. 本书源代码

本书所有源代码都作为保留版权的免费软件提供, 保留版权的主要目的是保证源代码得到正确的引用, 并防止在未经许可的情况下, 在印刷材料中发布代码。通常, 只要源代码获得了正确的引用, 则在大多数媒体中使用本书的示例都没有什么问题。

读者可在自己的开发项目中使用代码, 并可在课堂上引用(包括学习材料), 但要确定版权声明在每个源文件中得到了保留。

5. 致谢

本书是笔者多年 Java 授课过程中理论与实践的积累和体会, 这里对在本书写作过程中给予支持的单位、领导、专家、朋友们和家人致以真挚的感谢!

感谢对外经济贸易大学信息学院对本书的编写出版的支持; 感谢刘瑞林老师和实验中心积极关注和支持本书的编写工作; 感谢选修本课程的同学提供给我的建议和学习体会, 对本书的教学设计和教学安排提供了参考。

感谢家人和朋友们; 感谢伊凡先生提供的丰富专业资料, 和他的交流与讨论使得本书的编写受益匪浅。同时对本书内容提供帮助的吴进宝、于晏浩、张苑、王溢、何琪乐表示感谢。

特别感谢对外经济贸易大学出版社的各级领导和本书的策划编辑李晨光等, 由于他们的充分信任、支持和辛勤工作才使得本书能很快与读者见面。

编 者
2012 年 4 月

目 录

第 1 章 Java 概述	1
1.1 知识要点	1
1.2 实验	6
1.3 小结	20
第 2 章 Java 语言基础	21
2.1 知识要点	21
2.2 实验	26
2.3 小结	51
第 3 章 类和对象	53
3.1 知识要点	53
3.2 实验	57
3.3 小结	101
第 4 章 继承与多态	103
4.1 知识要点	103
4.2 实验	104
4.3 小结	145
第 5 章 异常处理	147
5.1 知识要点	147
5.2 实验	150
5.3 小结	164
第 6 章 数组与集合	165
6.1 知识要点	165

2 << Java 与面向对象程序设计实验教程

6.2 实验	167
6.3 小结	181
第 7 章 线程	183
7.1 知识要点	183
7.2 实验	184
7.3 小结	198
第 8 章 输入输出流	199
8.1 知识要点	199
8.2 实验	202
8.3 小结	213
第 9 章 用户图形界面	215
9.1 知识要点	215
9.2 实验	217
9.3 小结	247
第 10 章 网络应用	249
10.1 知识要点	249
10.2 实验	251
10.3 小结	263
参考文献	264

第 1 章

Java 概述

通过本章的实验，理解面向对象的基本概念，掌握 Java 程序的基本结构以及基础编程、编译和运行过程，学会 Java 开发环境的配置。

1.1 知 识 要 点

1.1.1 建立 Java 的开发环境

JDK 是 Java 开发工具包（Java Development Kit）的缩写。JDK 是 Java 最基本的工具和开发环境。Java 技术中的 J2EE、J2SE、J2ME 是 Java API 的三个不同版本，所使用的语言是相同的，只是捆绑的库不同。它们的核心都是 JDK。JDK 是整个 Java 的核心，包括了 Java 运行环境（Java Runtime Environment, JRE）、Java 编译器、大量的 Java 工具以及 Java 基础 API。

JRE（Java Runtime Environment, Java 运行环境），也就是 Java 平台。所有的 Java 程序都要在 JRE 下才能运行，JVM（Java Virtual Machine, Java 虚拟机）是 JRE 的一部分。JDK 的工具也是 Java 程序，也需要 JRE 才能运行。如果要使用 Java 技术开发应用程序，首先需要安装 JDK。为了保持 JDK 的独立性和完整性，在 JDK 的安装过程中，JRE 也是安装的一部分。所以，在 JDK 的安装目录下有一个名为 jre 的目录，用于存放 JRE 文件。

JDK 包含的常用基本工具包括：

➤ **javac**

Java 源程序编译器，将 Java 源代码转换成字节码。

➤ java

Java 解释器，直接从字节码文件，又称为类文件，执行 Java 应用程序的字节代码。

➤ appletviewer

Java Applet 浏览器。appletviewer 命令可以脱离万维网浏览器环境，直接运行 Applet。

➤ jar

Java 应用程序打包工具，可将多个类文件合并为单个 JAR 归档文件。

➤ javadoc

Java API 文档生成器用于从 Java 源程序代码注释中提取文档，生成 API 文档 HTML 页。

➤ jdb

Java 调试器 (debugger)，可以逐行执行程序，设置断点和检查变量。

目前 SUN 公司发布的 JDK 版本最高为 JDK 7，下载的官方网站地址为：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

JDK 有 Linux、Solaris 和 Windows 等操作系统的版本。安装完 Java Development Kit (JDK) 软件后，默认的安装路径为：C:\ProgramFiles\Java\JDK1.7.0。

安装完 JDK 后，在行命令窗口下就可以使用 JDK 中的 javac 和 java 命令了。如果运行命令时，出现错误的命令或找不到某个文件的提示，需要设置环境变量 PATH 和 CLASSPATH。PATH 变量指示 Java 应用程序查找 JDK 命令的路径，假设 JDK 安装在 Windows 操作系统的默认目录下，PATH 应设置为：C:\ProgramFiles\Java\JDK1.7.0\bin。

CLASSPATH 变量指示 Java 应用程序和 JDK 工具去查找第三方和自定义 Java 类文件的路径，即 Java 源代码编译后生成的字节码文件。这两个环境变量设置的步骤为：

(1) 环境变量的设置：我的电脑→属性→高级→环境变量。

(2) 在系统变量下的列表中，单击“新建”，变量名输入 CLASSPATH，变量值以“path1;path2;...”的方式输入 JDK 的类，以及第三方和自定义 Java 类文件所在的多个路径，确认完成。

(3) 在系统变量下的列表中，双击选择 PATH 变量，在所出现编辑框中的变量值尾部加入“;C:\ProgramFiles\Java\JDK1.7.0\bin”，这里用“;”分割，然后确认完成。

在 JDK 1.5 版本之前，Windows 系列操作系统需要程序员手动设置 PATH 环境变量，JDK 1.6 版本后，JDK 安装程序在安装完成后自动设置 PATH 路径；在 Linux 操作系统中，假设 JDK 安装在/usr/local/JDK1.6.0_04 目录下，需要在系统的启动文件中进行设置，加入 PATH=/usr/local/JDK1.6.0_04/bin 语句。

Java 的集成开发工具有很多，现在比较流行的是 Eclipse、NetBeans 和 JBuilder。Eclipse 是一个开源的绿色软件平台。

1.1.2 编译 Java 程序

在 Java 应用程序的开发过程中, Java 源程序代码扩展名为“.java”,以纯文本的方式存储。源文件由 Java 编译器编译成 Java 虚拟机可执行的字节码,生成扩展名为“.class”的字节码文件。这些字节码都由 JVM 的机器语言组成,由 JVM 装载,可以跨平台执行。javac 是 Java 语言编程编译器,用于将 Java 源程序编译成字节码的 class 文件。

通过命令行的方式, javac 可以执行编译的操作, 其命令行的执行语法为:

```
javac <options><sourcefiles>
```

编译器有一批标准选项,目前的开发环境支持这些标准选项,将来的版本也将支持它。还有一批附加的非标准选项是目前的虚拟机实现所特有的,将来可能要有变化。非标准选项以-X 打头,其中主要的标准选项包括:

> -classpath 类路径

设置用户类的路径,它会覆盖 CLASSPATH 环境变量中的用户类路径。若既未指定 CLASSPATH 又未指定-classpath,则用户类路径由当前目录构成。有关详细信息,请参阅设置类路径。若未指定-sourcepath 选项(见下文),则将在用户类路径中查找类文件和源文件。

> -d 目录

设置输出类文件的位置。如果某个类是一个包的组成部分,则 javac 将把该类文件放入反映包名的子目录中,必要时创建目录。例如,类名叫 com.mypackage.MyClass,如果指定-d c:\myclasses,那么类文件的路径为 c:\myclasses\com\mypackage\MyClass.class。若未指定-d 选项,则 javac 将把类文件放到与源文件相同的目录中。

-d 选项指定的目录不会被自动添加到用户类路径中。

> -deprecation

显示每种不鼓励使用的 API(包含成员和类)的使用或覆盖情况说明。没有给出 -deprecation 选项的话, javac 将显示这类源文件的名称:这些源文件使用或覆盖了不鼓励使用的 API。

> -encoding

设置源文件编码名称,例如 EUCJIS/SJIS。若未指定-encoding 选项,则使用平台缺省的转换器。

> -g

生成所有的调试信息,包括局部变量。缺省情况下,只生成行号和源文件信息。

> -g:none

不生成任何调试信息。

-g:{关键字列表}

只生成某些类型的调试信息，这些类型由逗号分隔的关键字列表所指定。有效的关键字有：**source** 表示源文件调试信息；**lines** 表示行号调试信息；表示 **vars** 局部变量调试信息。

➤ **-nowarn**

禁用警告信息。

➤ **-o**

优化代码，缩短执行时间。使用 **-o** 选项可能使编译速度下降，生成更大的类文件并使程序难以调试。在 JDK 1.2 以前的版本中，**javac** 的 **-g** 选项和 **-o** 选项不能一起使用。在 JDK 1.2 中，可以将 **-g** 和 **-o** 选项结合起来，但可能会得到意想不到的结果，如丢失变量或重新定位代码或丢失代码。**-o** 选项不再自动打开 **-depend** 或关闭 **-g** 选项。同样，**-o** 选项也不再允许进行跨类内嵌。

➤ **-sourcepath** 源路径

指定用来查找类或接口定义的源代码路径。与用户类路径一样，多个源路径项用分号“;”进行分隔，它们可以是目录、JAR 归档文件或 ZIP 归档文件。如果使用包，那么目录或归档文件中的本地路径名必须反映包名。（注意：通过类路径查找的类，如果找到了其源文件，则可能会自动被重新编译。）

➤ **-verbose**

冗长输出。它包括了每个所加载的类和每个所编译的源文件的有关信息。

➤ **-target** 类文件的版本。

如果已经指定了 **-source** 选项，那么 **-target** 版本不能低于 **-source** 选项。表 1-1 默认 **source** 与 **target** 选项值列出不同版本的 **javac**，默认 **source** 与 **target** 选项值。

表 1-1 默认 **source** 与 **target** 选项值

JDK/J2SDK	DefaultSource	SourceRange	DefaultTarget	TargetRange
1.0.x	1.0	—	1.1	—
1.1.x	1.1	—	1.1	—
1.2.x	1.2	—	1.1	1.1–1.2
1.3.x	1.2/1.3	—	1.1	1.1–1.3
1.4.x	1.2/1.3	1.2†–1.4	1.2	1.1–1.4
5	1.5	1.2–1.5	1.5	1.1–1.5
6	1.5	1.2–1.6	1.6	1.1–1.6
7	1.7	1.2–1.7	1.7	1.1–1.7

默认情况下,类是根据与 `javac` 一起发行的 JDK 自举类和扩展类来编译。但 `javac` 也支持联编,在联编中,类是根据其他 Java 平台实现的自举类和扩展类来进行编译的。联编时, `-bootclasspath` 和 `-extdirs` 的使用很重要。

➤ `-extdirs` 目录

根据指定的扩展目录进行联编。目录是以分号分隔的目录列表。在指定目录的每个 JAR 归档文件中查找类文件。

1.1.3 运行 Java 程序

在编译完成的类文件后,可以通过 JVM 来装载和执行的类文件,JVM 通过 SDK 的 `java` 命令来启动。`java` 命令既可以运行字节码.class 文件,也可以运行字节码文件打包后的 jar 文件。其用法为:

```
java [-options] class [args...]  
java [-options] -jar jarfile [args...]
```

其中 `-options` 为虚拟机参数,通过这些参数可对虚拟机的运行状态进行调整,掌握参数的含义可对虚拟机的运行模式有更深入理解。虚拟机参数分为基本和扩展两类,在命令行中输入 `java` 就可得到基本参数的说明,这些选项包括:

➤ `-client` 和 `-server`

这两个参数用于设置虚拟机使用何种运行模式, `client` 模式启动比较快,但运行时性能和内存管理效率不如 `server` 模式,通常用于客户端应用程序。相反, `server` 模式启动比 `client` 慢,但可获得更高的运行性能。在 Windows 上,默认的虚拟机类型为 `client` 模式。在 Linux、Solaris 上默认采用 `server` 模式。

➤ `-hotspot`

与 `-client` 相同, `jdk1.4` 以前使用的参数, `jdk1.4` 开始不再使用,代之以 `client`。

➤ `-classpath` 和 `-cp`

虚拟机在运行一个类时,需要将其装入内存,虚拟机搜索类的方式和顺序如下:自举类,扩展类,用户类。自举类是虚拟机自带的 jar 或 zip 文件,扩展类是位于 `jre\lib\ext` 目录下的 jar 文件,虚拟机在搜索完自举类后就搜索该目录下的 jar 文件,用户类的路径搜索顺序为当前目录、环境变量 `CLASSPATH`。参数 `-classpath` 用来设置虚拟机搜索目录名、jar 文件名和 zip 文件名,路径之间用分号分隔。如果 `-classpath` 和 `CLASSPATH` 都没有设置,则虚拟机使用当前路径“.”作为类搜索路径。推荐使用 `-classpath` 来定义虚拟机要搜索的类路径,以减少多个项目同时使用 `CLASSPATH` 时存在的潜在冲突。

➤ `-D<propertyName>=value`

在虚拟机的系统属性中设置属性名和对应值。

➤ **-version**

显示可运行的虚拟机版本信息然后退出。

➤ **-showversion**

显示版本信息以及帮助信息。

➤ **-ea[:<packagename>...[:<classname>]]**和

-enableassertions[:<packagename>...[:<classname>]]

从 JDK1.4 开始, java 可支持断言机制, 用于诊断运行时问题。通常在测试阶段使断言有效, 在正式运行时不需要运行断言。上述参数就用来设置虚拟机是否启动断言机制。缺省时虚拟机关闭断言机制, 用 **-ea** 可打开断言机制, 不加 **<packagename>** 和 **classname** 时运行所有包和类中的断言, 如果希望只运行某些包或类中的断言, 可将包名或类名加到 **-ea** 之后。例如要启动包 **com.foo.util** 中的断言, 可用命令 **-ea:com.foo.util**。

➤ **-da[:<packagename>...[:<classname>]]**和

-disableassertions[:<packagename>...[:<classname>]]

用来设置虚拟机关闭断言处理, **packagename** 和 **classname** 的使用方法和 **-ea** 相同。

➤ **-esa|-enablesystemassertions**

设置虚拟机显示系统类的断言。

➤ **-dsa|-disablesystemassertions**

设置虚拟机关闭系统类的断言。

➤ **-agentlib:<libname>[=<options>]**

是新引入 JDK5 参数, 用于虚拟机装载本地代理库。Libname 为本地代理库文件名, 虚拟机的搜索路径为环境变量 **PATH** 中的路径, options 为传给本地库启动时的参数, 多个参数之间用逗号分隔。

➤ **-agentpath:<pathname>[=<options>]**

设置虚拟机按全路径装载本地库, 不再搜索 **PATH** 中的路径。其他功能和 **agentlib** 相同。

1.2 实 验

实验 1 了解 Java SDK 的安装和使用

➤ **实验目的**

- (1) 掌握 Java 运行环境的安装和基本配置;
- (2) 了解 JDK 的组成与其提供的基本工具;