



高职高专**立体化教材**计算机系列

Java 程序设计

(第2版)

高晓黎 编著



赠送电子课件及
其他立体化资源



清华大学出版社

高职高专立体化教材 计算机系列

Java 程序设计(第2版)

高晓黎 编著

常州大学图书馆
藏书章

清华大学出版社
北京

内 容 简 介

本书在保持第一版教材原创风格的基础上,按照案例化、实例化的设计原则,解析了原有的知识体系,重新构建了新的知识结构与技术框架,增加了大量具有原创性的案例和实例,并将知识的讲授与案/实例的剖析讲解有机地结合起来,力图将 Java 面向对象的实质和特色淋漓尽致地展示给读者,帮助读者扎实而深入地掌握 Java 编程的技能与技巧,培养读者灵活运用知识进行应用开发的能力和素养。

全书共分 12 章,包含了 75 个精心创作或改编的案例和实例。主要内容包括:Java 程序开发环境的搭建、面向对象的 Java 编程、Java 编程基础、抽象与封装、字符串与数组、继承与多态、异常处理、图形化用户界面设计、事件处理、GUI 应用开发、输入输出流、Java 数据库应用开发等。每章后面都配有丰富的习题,附录的内容为主要习题的参考答案。

本书可作为培养应用型、技能型人才的 Java 课程教材,亦可作为相关专业的大中专学生及从事 Java 开发的技术人员的参考书。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计/高晓黎编著. --2 版. --北京:清华大学出版社,2015

(高职高专立体化教材 计算机系列)

ISBN 978-7-302-39540-9

I. ①J… II. ①高… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 034001 号

责任编辑:桑任松

封面设计:刘孝琼

责任校对:周剑云

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:22.75 字 数:552 千字

版 次:2008 年 10 月第 1 版 2015 年 8 月第 2 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~2500

定 价:41.00 元

产品编号:062473-01

“高职高专立体化教材 计算机系列”丛书序

一、编写目的

关于立体化教材,国内外有多种说法,有的叫“立体化教材”,有的叫“一体化教材”、有的叫“多元化教材”,其目的都是一样的,就是要为学校提供一种教学资源的整体解决方案,最大限度地满足教学需要,满足教育市场需求,促进教学改革。我们这里所讲的立体化教材,其内容、形式、服务都是建立在当前技术水平和条件基础上的。

立体化教材是“一揽子”式的,包括主教材、教师参考书、学习指导书、试题库在内的完整体系。主教材讲究的是“精品”意识,既要具备指导性和示范性,也要具有一定的适用性,喜新不厌旧,内容愈编愈多,本子愈编愈厚的低水平重复建设在“立体化”的世界中将被扫地出门。和以往不同,“立体化教材”中的教师参考书可不是千人一面的,教师参考书不只是提供答案和注释,而是含有与主教材配套的大量参考资料,使得老师在教学能做到“个性化教学”。学习指导书更像一本明晰的地图册,难点、重点、学习方法一目了然。试题库或习题集则要完成对教学效果进行测试与评价的任务。这些组成部分采用不同的编写方式,把教材的精华从各个角度呈现给师生,既有重复、强调,又有交叉和补充,相互配合,形成一个教学资源有机的整体。

除了内容上的扩充,立体化教材的最大突破还在于表现形式上走出了“书本”这一平面媒介的局限,如果说音像制品让平面书本实现了第一次“突围”,那么电子和网络技术的大量运用就让躺在书桌上的教材真正“活”了起来。用 PowerPoint 开发的电子教案不仅大大减少了教师案头备课的时间,而且也让学生的课后复习更加有的放矢。电子图书通过数字化使得教材的内容得以无限扩张,使平面教材更能发挥其提纲挈领的作用。

CAI 课件将动画、仿真等技术引入课堂,让课程的难点和重点一目了然,通过生动的表达方式达到深入浅出的目的。在科学指标体系控制之下的试题库既可以轻而易举地制作标准化试卷,也能让学生进行模拟实战的在线测试,提高了教学质量评价的客观性和及时性。网络课程更厉害,它使教学突破了空间和时间的限制,彻底发挥了立体化教材本身的潜力,轻轻敲击几下键盘,你就能在任何时候得到有关课程的全部信息。

最后还有资料库,它把教学资料以知识点为单位,通过文字、图形、图像、音频、视频、动画等各种形式,按科学的存储策略组织起来,大大方便了教师在备课、开发电子教案和网络课程时的教学工作。如此一来,教材就“活”了。学生和书本之间的关系不再像领导与被领导那样呆板,而是真正有了互动。教材不再是只为老师规定什么重要、什么不重要,而是成为教师实现其教学理念的最佳拍档。在建设观念上,从提供和出版单一纸质教材转向提供和出版较完整的教学解决方案;在建设目标上,以最大限度满足教学要求为根本出发点;在建设方式上,不是单纯以现有教材为核心,简单地配套电子音像出版物,

而是以课程为核心,整合已有资源并聚拢新资源。

网络化、立体化教材的出版是我社下一阶段教材建设的重中之重,作为以计算机教材出版为龙头的清华大学出版社确立了“改变思想观念,调整工作模式,构建立体化教材体系,大幅度提高教材服务”的发展目标。并提出了首先以建设“高职高专计算机立体化教材”为重点的教材出版规划,通过邀请全国范围内的高职高专院校的优秀教师,在2008年共同策划、编写了这一套高职高专立体化教材,利用网络等现代技术手段实现课程立体化教材的资源共享,解决国内教材建设工作中存在教材内容的更新滞后于学科发展的状况。把各种相互作用、相互联系的媒体和资源有机地整合,形成立体化教材,把教学资料以知识点为单位,通过文字、图形、图像、音频、视频、动画等各种形式,按科学的存储策略组织起来,为高职高专教学提供一整套解决方案。

二、教材特点

在编写思想上,以适应高职高专教学改革的需要为目标,以企业需求为导向,充分吸收国外经典教材及国内优秀教材的优点,结合中国高校计算机教育的教学现状,打造立体化精品教材。

在内容安排上,充分体现先进性、科学性和实用性,尽可能选取最新、最实用的技术,并依照学生接受知识的一般规律,通过设计详细的可实施的项目化案例(而不仅仅是功能性的小例子),帮助学生掌握要求的知识点。

在教材形式上,利用网络等现代技术手段实现立体化的资源共享,为教材创建专门的网站,并提供题库、素材、CAI课件、案例分析,实现教师和学生更大范围内的教与学互动,及时解决教学过程中遇到的问题。

本系列教材采用案例式的教学方法,以实际应用为主,理论够用为度。教程中每一个知识点的结构模式为“案例(任务)提出→案例关键点分析→具体操作步骤→相关知识(技术)介绍(理论总结、功能介绍、方法和技巧等)”。

该系列教材将提供全方位、立体化的服务。网上提供电子教案、文字或图片素材、源代码、在线题库、模拟试卷、习题答案、案例动画演示、专题拓展、教学指导方案等。

在为教学服务方面,主要是通过教学服务专用网站在网络上为教师和学生提供交流的场所,每个学科、每门课程,甚至每本教材都建立网络上的交流环境,可以为广大教师信息交流、学术讨论、专家咨询提供服务,也可以让教师发表对教材建设的意见,甚至通过网络授课。对学生来说,则在教学支撑平台上所提供的自主学习空间来实现学习、答疑、作业、讨论和测试,当然也可以对教材建设提出意见。这样,在编辑、作者、专家、教师、学生之间建立起一个以网络为纽带、以数据库为基础、以网站为门户的立体化教材建设与实践的体系,用快捷的信息反馈机制和优质的教学服务促进教学改革。

前言

自 2008 年 10 月教材的第一版出版以来, Java 技术正以日新月异的速度在不断地发展着。Java 语言以自身诸多优秀的品格、跨平台的特性, 以及开源的理念, 为越来越多的人士所喜爱、所使用, 成为网络编程、手机开发等领域的主流编程语言。

感谢本教材第一版的忠实读者们。他们中有大中专院校的学生, 有高校或培训机构的教师, 有科研院所的研究人员, 还有从事 Java 开发的技术人员。感谢他们对本教材的选择与肯定, 感谢他们在使用本教材的过程中, 向作者提出了许多中肯的意见与合理的建议。这些宝贵的意见和建议, 使我在修订教材的进程中受益无穷。毫不夸张地讲: 没有读者一贯的支持与鼓励, 就没有教材第二版的问世。

新教材采用版本较新的 JDK7 和广为流行的 MyEclipse 作为开发平台, 对原有教材的知识体系作了重大的调整, 删除了部分不合时宜的内容, 增加了数据库开发等章节, 并进一步加大了面向对象知识的讲解力度, 使新教材的结构与内容更为合理, 更适合初学者使用。

与第一版相比, 本书在知识安排与讲解体例上都有重大的改进。本书彻底采用案例、实例教学法展开讲解, 75 个精心编制的案例和实例贯穿全书。每个案例与实例都紧扣核心知识点, 通过解题思路点拨与代码剖析, 向读者传授编程的方法与技巧, 揭示编程的要义与规律, 以期提升读者抽象思维和逻辑思维的素养, 培养读者分析问题与解决问题的能力。

本书适合理论与实践一体化的教学模式。总学时建议不宜低于 120 课时。各章节的教学内容与建议学时如下表所示。教学过程中, 教师可酌情调整。

教学内容与学时建议参考表

章 节	教学内容	建议学时	章 节	教学内容	建议学时
第 1 章	Java 程序开发环境的搭建	6	第 7 章	异常处理	6
第 2 章	面向对象的 Java 编程	6	第 8 章	图形化用户界面设计	14
第 3 章	Java 编程基础	14	第 9 章	事件处理	10
第 4 章	抽象与封装	10	第 10 章	GUI 应用开发	8
第 5 章	字符串与数组	10	第 11 章	输入输出流	8
第 6 章	继承与多态	14	第 12 章	Java 数据库应用开发	14

建议总学时: 120

本书在编著与出版的过程中, 还得到了“山东省重大水利科研与技术推广专项资金”的资助, 在此一并表示感谢。

本书所有的附加资源, 包括电子课件、图表素材、实例代码、模拟试卷、实训指导书等, 读者可以从清华大学出版社的官网上自由下载, 也可以发邮件到作者的电子信箱(wishuluck@126.com)来索取。

每一部书稿,从交稿的那一天起,都注定要成为我人生中的又一次遗憾。总是在完稿的那一刻,才感觉到自己脑海中一直激荡着的许多绝妙的设想和构思,却没有最终完美地体现在书稿中。这也许要归咎于本人学艺不精,抑或是笔力不逮。由于知识与水平所限,书中的错误与疏漏在所难免,还请各位读者与专家予以谅解。随时欢迎您的批评、指正和建议,编者将不胜感激。

最后,恭祝每位与本书有缘的读者,能够诸事遂意,心想事成!

编 者

目 录

第 1 章 Java 程序开发环境的搭建	1
1.1 案例 1: 基于 JDK 的 Java 程序验证	1
1.1.1 必备知识	1
1.1.2 案例综述	2
1.1.3 解决步骤	4
1.1.4 拓展提升	10
1.2 案例 2: 基于 MyEclipse 的 Java 程序验证	10
1.2.1 必备知识	11
1.2.2 案例综述	11
1.2.3 解决步骤	12
1.2.4 拓展提升	20
习题	20
第 2 章 面向对象的 Java 编程	23
2.1 案例 1: 输出单行信息的 Java 应用程序	23
2.1.1 必备知识	23
2.1.2 案例综述	24
2.1.3 案例解决	24
2.1.4 拓展提升	28
2.2 案例 2: 体现面向对象特征的 Java 程序	31
2.2.1 必备知识	31
2.2.2 案例综述	33
2.2.3 案例解决	33
2.2.4 拓展提升	37
2.3 案例 3: Java Applet 程序	38
2.3.1 必备知识	39
2.3.2 案例综述	39
2.3.3 案例解决	40
2.3.4 拓展提升	42
习题	43
第 3 章 Java 编程基础	45
3.1 单元 1: 数据表达	45

3.1.1 Java 语言符号集	45
3.1.2 Java 数据类型	47
3.1.3 常量与变量	47
3.1.4 数据类型转型	50
3.2 单元 2: 运算符与表达式	51
3.2.1 运算符与表达式	51
3.2.2 运算符的优先级	56
3.3 单元 3: 程序的结构	57
3.3.1 语句与顺序结构	58
3.3.2 分支结构	59
3.3.3 循环结构	64
3.3.4 循环嵌套	71
3.3.5 跳转语句	74
习题	78
第 4 章 抽象与封装	80
4.1 单元 1: 类的构成要素	80
4.1.1 面向对象的基本特征	80
4.1.2 类的定义	82
4.1.3 属性声明	85
4.1.4 构造方法定义	87
4.1.5 成员方法定义	89
4.2 单元 2: Java 的封装机制	93
4.2.1 Java 的修饰符	93
4.2.2 访问修饰符	94
4.2.3 特征修饰符	99
4.3 单元 3: 成员方法的应用	102
4.3.1 成员方法的分类	102
4.3.2 成员方法中的数据	103
4.3.3 读/写属性的成员方法	104
习题	109
第 5 章 字符串与数组	112
5.1 单元 1: 常用的系统内部类	112
5.1.1 Java 的系统类包	112

5.1.2	Math 类.....	113
5.1.3	基本数据类型的封装.....	116
5.2	单元 2: 字符串 String.....	121
5.2.1	String 类.....	121
5.2.2	StringBuffer 类.....	125
5.3	单元 3: 数组.....	127
5.3.1	一维数组.....	127
5.3.2	多维数组.....	131
5.3.3	数组的应用.....	136
	习题.....	140
第 6 章	继承与多态.....	142
6.1	单元 1: 继承机制.....	142
6.1.1	继承的概念与实现.....	142
6.1.2	构造方法的继承.....	144
6.1.3	继承的特征与意义.....	148
6.2	单元 2: Java 的多态.....	149
6.2.1	属性隐藏与方法重写.....	149
6.2.2	方法重载.....	153
6.2.3	重载与重写的不同.....	158
6.3	单元 3: 抽象类与最终类.....	159
6.3.1	抽象类与抽象方法.....	159
6.3.2	最终类与最终方法.....	163
6.4	单元 4: 接口.....	165
6.4.1	接口的定义与实现.....	165
6.4.2	深入理解接口.....	171
6.5	单元 5: 包.....	176
6.5.1	包及包的创建.....	176
6.5.2	包的引用.....	178
	习题.....	181
第 7 章	异常处理.....	184
7.1	单元 1: 异常处理.....	184
7.1.1	异常类.....	184
7.1.2	异常处理机制.....	187
7.2	单元 2: 定制异常处理.....	190
7.2.1	抛出异常.....	191
7.2.2	自定义异常.....	193
7.2.3	异常处理机制的特点.....	197

习题.....	197
第 8 章 图形化用户界面设计	199
8.1 单元 1: AWT	199
8.1.1 AWT 的类与组织	199
8.1.2 设置颜色与字体	203
8.1.3 AWT 的应用程序窗体	205
8.2 单元 2: Swing 的容器组件	208
8.2.1 框架窗体 JFrame.....	209
8.2.2 其他的容器类	213
8.3 单元 3: Swing 的基本组件	215
8.3.1 基本组件类	215
8.3.2 标签与命令按钮	216
8.3.3 文本类组件	218
8.3.4 选择类组件	221
8.3.5 列表类组件	223
8.4 单元 4: 布局管理器.....	227
8.4.1 AWT 的布局管理器	228
8.4.2 BorderLayout.....	228
8.4.3 GridLayout.....	230
8.4.4 FlowLayout.....	232
8.4.5 CardLayout	235
习题.....	237
第 9 章 事件处理	240
9.1 单元 1: 事件驱动编程.....	240
9.1.1 重要概念	240
9.1.2 Java 的事件	241
9.1.3 事件处理原理	243
9.2 单元 2: 监听器接口与适配器.....	247
9.2.1 事件监听器接口	247
9.2.2 事件适配器类	251
9.2.3 内部类	254
9.2.4 匿名类	257
习题.....	260
第 10 章 GUI 应用开发	262
10.1 单元 1: 菜单系统与工具栏.....	262
10.1.1 菜单系统设计	262
10.1.2 工具栏	270

10.2 单元 2: Swing 的高级组件	273	11.3.3 字符文件流类 FileReader 和 FileWriter	303
10.2.1 选项卡窗格集 JTabbedPane	273	11.3.4 字符缓冲流类 BufferedReader 和 BufferedWriter	305
10.2.2 表格 JTable	277	11.3.5 系统类 System	308
10.2.3 层次树 JTree	280	习题	312
习题	284	第 12 章 Java 数据库应用开发	314
第 11 章 输入输出流	286	12.1 单元 1: JDBC	314
11.1 单元 1: 流与文件	286	12.1.1 JDBC 概述	315
11.1.1 流的类别	286	12.1.2 JDBC 驱动程序的类别	317
11.1.2 文件类 File	289	12.1.3 JDBC 的类与接口	318
11.2 单元 2: 字节流的输入输出	293	12.1.4 为 MyEclipse 配置 JDBC 运行环境	324
11.2.1 抽象字节流类 InputStream 与 OutputStream	294	12.2 单元 2: JDBC 操纵数据库中的 数据	327
11.2.2 字节文件流类 FileInputStream 和 FileOutputStream	295	12.2.1 编写 JDBC 数据库应用 程序的步骤	327
11.2.3 字节缓冲流类 BufferedInputStream 和 BufferedOutputStream	297	12.2.2 创建数据库表	333
11.2.4 字节数据流类 DataInputStream 和 DataOutputStream	298	12.2.3 查询数据	336
11.3 单元 3: 字符流的输入输出	301	12.2.4 增加数据	339
11.3.1 抽象字符流类 Reader 和 Writer	301	12.2.5 修改数据	342
11.3.2 转换流类 InputStreamReader 和 OutputStreamWriter	302	12.2.6 删除数据	344
		习题	346
		各章课后习题参考答案	348
		参考文献	354

第 1 章 Java 程序开发环境的搭建

章节导读

对于初学者，通过验证 Java 案例，可以感性地认识 Java 程序的结构与特点，初步了解 Java 程序开发的环境与流程。本章给出两个验证性的案例，其功能分别是绘制四叶玫瑰曲线，以及模拟显示唱片图案。通过对第一个案例的验证，读者能够掌握 JDK 平台的下载、安装与配置方法，学会编写、编译、调试与运行 Java 程序的操作步骤。通过对第二个案例的验证，读者能够掌握 MyEclipse 的安装与配置方法，学会创建项目和构造类的操作步骤。两个案例的学习，使读者初步领悟到 Java 程序的本质与特征，为后续学习奠定基础。

英文缩略词

本章重要的英文缩略词如表 1.1 所示。

表 1.1 本章重要的英文缩略词索引表

缩 略 词	英文全称	汉译名称
Java EE	Java Enterprise Edition	Java 2 企业版本
Java SE	Java Standard Edition	Java 2 标准版本
Java ME	Java Micro Edition	Java 2 微型版本
JDK	Java Development Kit	Java 开发工具包
JRE	Java Runtime Environment	Java 运行时环境
JAR	Java Archive File	Java 压缩文件
JVM	Java Virtual Machine	Java 虚拟机
IDE	Integrated Development Environment	集成开发环境
AFC	Application Foundation Class Library	应用基本类库

1.1 案例 1：基于 JDK 的 Java 程序验证

作为初学者，也许你还不具备编写 Java 程序的能力，但是你完全能够按照操作指示，输入一段 Java 代码，然后对其进行编译、运行，观察这段代码的运行结果。通过对这种程序的验证，你可以初步认识 Java 程序“长”什么样子，了解编写 Java 程序的基本流程。这些经验，将有助于你更顺利地进行课程的学习。

1.1.1 必备知识

作为面向对象的编程语言，作为最主流的网络编程语言，同时也作为开源理念的代表，Java 无疑是广大编程爱好者的首选语言，它为软件开发领域带来了一场革命。

1. 三类 Java 系统平台

根据应用领域的不同,可将 Java 系统平台分为 Java SE、Java EE 和 Java ME 三种不同的版本。

在 Java 的三类平台系统中,Java SE 支持 Java 所有的核心类库与基本数据类型,定位于客户端(Client-side)程序的应用;Java EE 则在 Java SE 的基础之上,增加了支持企业复杂计算的扩充类库,定位于服务器端(Server-side)程序的应用;Java ME 支持的核心类库只是 Java SE 的一部分,但增加了支持嵌入式系统的扩充类库,定位于嵌入式系统的应用。三类系统的功能范畴如图 1.1 所示。

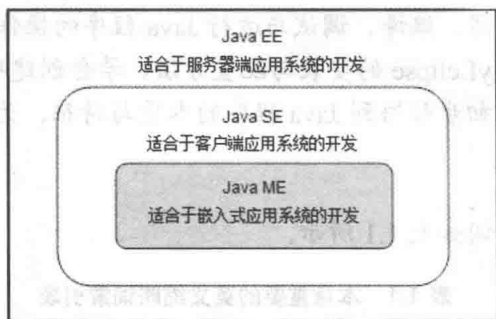


图 1.1 Java 平台系统的三种类型

以上三类系统中,Java EE 的优势在于支持网络应用开发和支持服务器级的复杂系统开发,成为电子商务、企业应用的热门技术;Java SE 适用于客户端及 PC 桌面系统的应用开发,成为个人、小型企业开发应用的新宠;而 Java ME 则主要用于消费性电子产品的开发领域。Java SE 是 Java 平台的核心,提供了 Java 语言的执行环境。同时它也是 Java EE 功能扩展的基础。

2. JDK 概述

JDK(Java Development Kit, Java 开发工具包),是 Oracle(甲骨文)公司免费提供的 Java 程序开发与运行的工具包,是 Java 语言系统的核心,也是其他 Java 开发工具的底层支撑平台,所有的 Java 开发工具必须遵循与支持 JDK 的标准。

JDK 的内容非常丰富,包含 Java 运行时环境 JRE(Java Runtime Environment)、Java 语言的基础类库源码,以及开发应用程序与 Applet 程序所用的编译、测试、调试、解释执行等一系列命令行程序。

建议:对于 Java 初学者,强烈推荐直接使用 JDK 来调试运行简单的程序,这样能够让用户更好地领悟 Java 的工作机制,理解各个类之间、各代码块之间的关系,帮助用户们建立起面向对象的编程思路。

1.1.2 案例综述

案例 1 为验证性案例。该案例的正确性将在 JDK 平台上加以验证。

1. 案例任务

要求使用 Windows 记事本, 编辑输入下面所给的案例 1 的源代码, 并保存为文本格式的 Java 程序文档; 然后在 JDK 平台上编译、调试和运行, 从而验证程序的输出结果是否为四叶玫瑰曲线图案。

案例 1 的程序代码(程序文档名称: Case_1_1.java)如下:

```
1: //引入标准的类库
2: import java.awt.*;
3: import java.awt.event.*;
4: import java.awt.Color;
5: //定义主类 Case_1_1
6: public class Case_1_1 extends Frame {
7:     private static final double PI=3.1415926; //定义圆周率为符号常量 PI
8:     private int appWidth,appHeight; //定义应用程序窗口的宽度与高度属性
9:     //定义主类的构造方法
10:    public Case_1_1(String title){
11:        super(title); //调用父类的构造方法显示窗口标题
12:        this.setSize(380,250);
13:        //实现关闭窗口功能
14:        this.addWindowListener(new WindowAdapter(){
15:            public void windowClosing(WindowEvent e) { System.exit(0); }
16:        });
17:    }
18:    //定义绘图方法 paint
19:    public void paint(Graphics g) {
20:        super.paint(g); //调用父类的绘制方法
21:        //获取当前窗口的宽度与高度值
22:        appWidth=this.getWidth();
23:        appHeight=this.getHeight();
24:        int ox,oy,x,y; //定义原点坐标变量 ox 与 oy
25:        //设置原点(ox,oy)的坐标为当前窗口中心
26:        ox =appWidth/2;
27:        oy =appHeight/2;
28:        g.setColor(Color.BLUE); //设置绘制坐标轴的画笔颜色为蓝色
29:        //绘制水平与垂直坐标轴
30:        g.drawLine(ox,10,ox,appHeight);
31:        g.drawLine(0,oy,appWidth,oy);
32:        //定义控制角度与半径变化的变量
33:        int i,j=10;
34:        double angle,r;
35:        g.setColor(Color.RED); //设置绘制四叶玫瑰曲线的画笔颜色为红色
36:        while (j<100){
37:            for(i=0;i<1024;i++){
38:                //计算最新的角度
39:                angle = i*PI/512;
40:                r = j*Math.sin(2*angle);
41:                //计算最新的轨迹点坐标
```

```

42: x =(int)Math.round(r*Math.cos(angle)*2);
43: y =(int)Math.round(r*Math.sin(angle));
44: g.fillOval(ox+x,oy+y,1,1); //画轨迹点
45: }
46: j = j + 5; //增进的步长为 5 个像素
47: }
48: }
49: //定义入口方法 main
50: public static void main(String[ ] args) {
51: Case_1_1 app=new Case_1_1("绘制四叶玫瑰曲线示例"); //创建应用实例
52: app.setVisible(true); //设置应用程序可见
53: }
54: }

```

2. 程序说明

程序的功能是绘制平面坐标轴及美丽的四叶玫瑰曲线。坐标轴与曲线的绘制由绘图方法 paint 完成,核心算法由以下两部分代码组成。

(1) 第一部分代码完成以下功能:获取绘图窗口的中心点坐标,调用 JDK 的 drawLine 方法绘制出水平坐标轴和垂直坐标轴。

(2) 第二部分代码完成以下功能:用正弦方法和余弦方法计算出四叶玫瑰曲线上每个像素点的逻辑坐标,然后调用 JDK 的 fillOval 方法,用红颜色逐点渲染出玫瑰曲线的所有像素点,最终形成综合的视觉效果。

3. 案例剖析

案例 1 的任务可分解为以下六个子任务。

- (1) 从 Oracle 官网上下载当前较高版本的 JDK。
- (2) 安装 JDK。
- (3) 为安装成功的 JDK 配置环境变量。
- (4) 学习更改 DOS 命令窗口的显示模式。
- (5) 使用记事本,编辑输入案例 1 的 Java 代码,以指定的文档名称保存。
- (6) 使用 JDK 的相应命令,编译并运行程序文档,检验程序输出结果是否正确。

1.1.3 解决步骤

依据案例任务的分解结果,顺序完成以下每个子任务。

1. 下载 JDK7

Oracle 公司的官方网站提供了可自由下载的各个版本的 JDK 软件安装包,网址是:
<http://www.oracle.com/index.html>(英文)或 <http://www.oracle.com/cn/index.html>(中文)。

【Step 1】访问 Oracle 公司的官网主页,单击 Downloads(下载)栏目,然后找到 JDK 的下载点,下载相应的项目。推荐下载 Java 平台标准版 Java SE 最新版本的 JDK 安装包,本书采用的版本为 JDK7。

【Step 2】下载前,必须选中 Accept License Agreement 单选按钮,接受许可协议,否

则无法下载软件。网页上默认的选择为 Decline License Agreement (拒绝接受许可协议), 如图 1.2 所示。

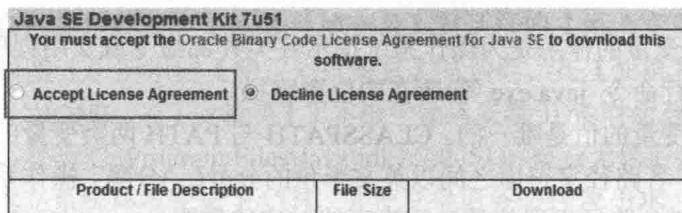


图 1.2 下载前必须接受许可协议

2. 安装 JDK

【Step 1】选择并运行下载的 JDK7 安装程序, 进入安装向导界面, 单击【下一步】按钮, 开始 JDK 的安装过程。

【Step 2】在弹出的【Java SE Development Kit 7 Update 51 - 自定义安装】对话框中, 可以对系统的功能模块有选择地进行安装; 对于用不到的功能模块, 可以取消安装, 从而避免存储资源的浪费。但“开发工具”模块属于必须安装项目, 用户无权取消安装。系统默认安装全部的功能模块。通过单击【更改】按钮, 可以改变系统默认的安装路径, 如图 1.3 所示。

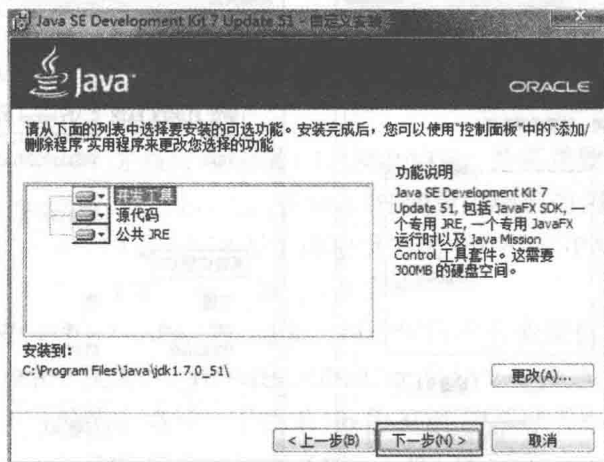


图 1.3 【Java SE Development Kit 7 Update 51 -自定义安装】对话框

【Step 3】单击【下一步】按钮, 弹出安装进度指示条。此时无须操作, 等待安装进度结束。

【Step 4】安装结束后, 弹出【完成】对话框。单击【关闭】按钮, 则退出 JDK 的安装。

3. 为 JDK 配置系统环境变量

使用 JDK 之前, 需要先将相关的系统环境变量设置好, 否则无法有效地运行 JDK 的命令。

配置 JDK 主要涉及以下三类系统环境变量。

(1) JAVA_HOME: 该变量的作用是指示 JDK 的安装根目录。

(2) CLASSPATH: 该变量的作用是为操作系统指示各类系统所包含的重要类所在的位置路径。

(3) PATH: 该变量的作用是为操作系统指示各类外部命令或实用程序(如 JDK 的编译命令 javac.exe、运行命令 java.exe 等)所在的位置路径。

JAVA_HOME 变量的值是唯一的。CLASSPATH 与 PATH 两类变量的值却可以包含多个路径指示字符串,各路径字符串之间以英文半角的分号(;)分隔。操作系统在检索类或命令的位置时,将依照各路径出现的顺序依次查找对应的位置。

不同版本的操作系统,环境变量的配置方法大致相同,只是某些细节会略有差别。下面以 Windows 7 为例,详细讲解配置 JDK 环境变量的方法与步骤。

【Step 1】在桌面上右击【计算机】图标,从弹出的快捷菜单中选择【属性】命令;或者选择 Windows 的【开始】|【控制面板】菜单命令,依次打开【系统和安全】和【系统】选项。这两种方法都将打开计算机系统。

【Step 2】单击信息窗口右侧列表中的【高级系统设置】选项,弹出【系统属性】对话框,在该对话框中切换到【高级】选项卡,如图 1.4 所示。

【Step 3】打开【环境变量】对话框。方法如下:单击【高级】选项卡右下方的【环境变量】按钮,弹出如图 1.5 所示的【环境变量】对话框。

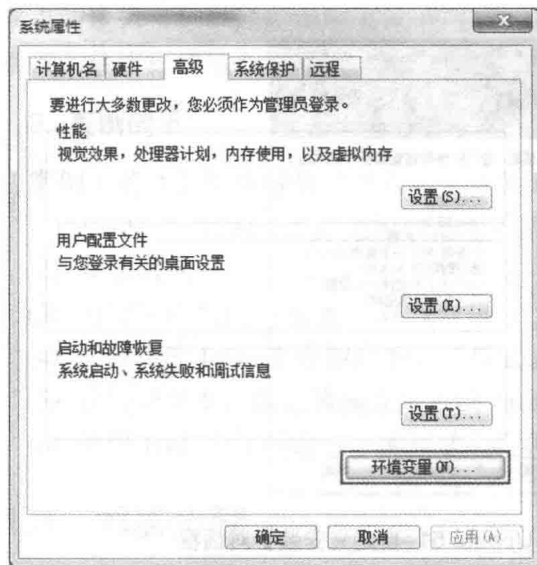


图 1.4 【系统属性】对话框



图 1.5 【环境变量】对话框

【环境变量】对话框提供了对环境变量进行编辑操作的手段。对话框分为上、下两个不同的编辑区：上方的编辑区能够对特定用户的环境变量进行新建、编辑与删除操作；下方的编辑区则针对所有用户的系统环境变量进行新建、编辑与删除操作。

对特定用户的环境变量进行配置,其效果是仅对以同样身份登录的用户起作用;对系统的环境变量进行配置,则对所有的用户都有效。因此,配置系统的环境变量比配置特定用户的环境变量影响的范围更为广泛。

提示：下面的操作将全部针对系统的环境变量。

【Step 4】创建系统环境变量 `JAVA_HOME`，指明 JDK 安装的根目录路径，用来简化后面两个环境变量的赋值操作。单击【环境变量】对话框的【系统变量】选项组中的【新建】按钮，打开如图 1.6 所示的【新建系统变量】对话框，在【变量名】文本框中输入环境变量名称 `JAVA_HOME`，在【变量值】文本框中输入安装 JDK 时所选择的根目录路径(笔者的安装根目录路径为 `C:\Program Files\Java\jdk1.7.0_51`)；检查无误后，单击【确定】按钮。

【Step 5】建立系统环境变量 `PATH`，并使变量值务必包含 JDK 安装根目录下的 `bin` 和 `jre\bin` 两个子目录，如图 1.7 所示。

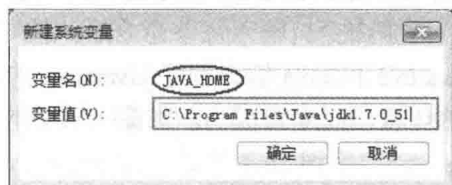


图 1.6 指定环境变量 `JAVA_HOME` 的值

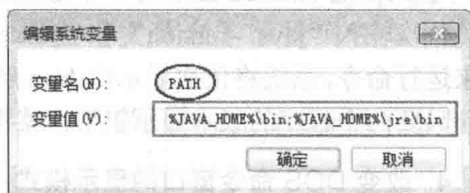


图 1.7 指定环境变量 `PATH` 的值

如果 `PATH` 变量已经存在，请不要破坏原有的变量值。此时只需单击【编辑】按钮，打开【编辑系统变量】对话框，修改 `PATH` 的值即可。修改时，在原有变量值的任意项位置后面添加 JDK 相应的两个子目录路径字符串“`%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;`”，两个路径项之间，以分号作为分隔符。其中“`%JAVA_HOME%`”是对环境变量 `JAVA_HOME` 内容的引用，代表了 JDK 安装根目录路径 `C:\Program Files\Java\jdk1.7.0_51`。

【Step 6】新建(或编辑)系统环境变量 `CLASSPATH`，并将重要的类库压缩文件包及类库的位置路径赋值给该变量。此处以 `CLASSPATH` 变量的编辑为例加以说明。

在【环境变量】对话框的【系统变量】选项组中选中 `CLASSPATH` 选项，单击【编辑】按钮，打开【编辑系统变量】对话框，如图 1.8 所示。

在该对话框中修改 `CLASSPATH` 的值。即在保留原有变量值的基础上，增加字符串“`.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;`”。

其中“`.`”代表系统的当前路径，子目录 `lib` 为 JDK 的基础类库与若干支撑文件的位置路径，`dt.jar` 和 `tools.jar` 为 JDK 的两个压缩文件，文件中封装着一些重要的系统类，它们是 JAR 格式的文件。

知识：JAR(Java Archive File)文件格式是一种与平台无关的压缩格式。JAR 文件允许将许多文件组合在一起，不仅用于压缩和发布，而且还用于部署和封装类库、组件和插件程序，并可以被 Java 编译器直接使用。

【Step 7】至此，环境变量已经完成配置，单击【确定】按钮，系统即将配置好的环境变量自动保存。

【Step 8】进入 DOS 命令窗口。方法如下：选择 Windows 的【开始】|【运行】菜单命令，在打开的【运行】对话框中输入 `CMD` 命令并按 `Enter` 键；或者选择 Windows 的【开始】|【所有程序】|【附件】|【命令提示符】菜单命令，打开如图 1.9 所示的 DOS 命令