

JAVA

十大经典案例

张恒汝 编著

100001111001010001010010001000111101010101
1100011110010100010100100010001111010101
01000111100101000101001000100011110101
111001111001010001010010001000111101

010001111001010001010010001
01000111100101000101001000
0100011110010100010100



科学出版社

www.sciencep.com

Java 十大经典案例

张恒汝 编著

科学出版社

北京

前 言

Java 语言由 C++ 衍生而来，其语言风格与 C++ 十分类似。Java 语言将面向对象、与平台无关、安全稳定、多线程等特点集于一身，其跨平台的特点特别适用于分布式计算机环境，Java 语言“编写一次、到处运行”的跨平台优势给整个网络世界带来了巨大变革。目前 Java 语言已经广泛地应用于 Internet 各种应用开发。以 Java 语言为基础的 J2EE 体系结构（注：J2EE 主要应用于商务开发）和 J2ME 体系结构（注：J2ME 主要应用于手机开发）是目前计算机行业中的热点，学习和掌握 Java 语言已经成为计算机类专业人才的一种迫切需要。

本书以十个经典的 Java 案例为主线，通过案例驱动的方式引导读者从零开始、一步步成为 Java 语言方面的专业人才。本书共分为十章：

第一章 以 JCreator 开发工具为例引导读者组建一个学习和开发 Java 程序的编程环境，并利用该环境编写基本的 Java Application 和 Java Applet；

第二章 以点、圆、圆柱体为例一步步引导读者理解 Java 语言面向对象的基本概念：类、对象、继承、多态性、包、接口等；

第三章 以求阶乘的异常处理为例逐步引导读者理解 Java 语言的异常处理机制：try-catch-finally 机制、throw 机制、throws 机制等；

第四章 以简单的计算器程序为例引导读者学习利用 Java 语言进行图形界面编程，包括 AWT 组件编程、Swing 组件编程、Applet 编程等；

第五章 以八皇后问题为例引导读者学习利用 Java 语言进行数据结构方面的编程，如利用 Java 语言实现堆栈、队列、链表、树等；

第六章 以文件加密为例引导读者理解 Java 语言对文件处理的支持，这些支持包括字符流和字节流的处理、随机文件的处理等；

第七章 以时钟显示为例引导读者逐步学习 Java 语言的多线程机制，包括如何利用 Java 语言创建线程、Java 线程生命期的描绘，以及如何实现 Java 多线程的同步等；

第八章 以 SMTP 客户端程序为例引导读者逐步学习利用 Java 语言进行网络程序的开发，包括利用 Java 语言开发 URL 连接的程序、基于套接字的各种网络程序等；

第九和十章 这两章是综合性应用章节，介绍了目前网络游戏中应用较普遍的“五子棋”和“俄罗斯方块”游戏，一方面读者可以了解“五子棋”和“俄罗斯方块”两种游戏的算法和源代码，另一方面通过学习这两种游戏的开发过程，

初步培养大中型软件的研究和开发能力；

书中各章都配有相应的习题，以利于读者对本章内容的巩固。本章的案例可作为读者学习时参考，也可作为读者开发具体项目时借鉴。

在本书编写过程中，科学出版社的杨岭老师、罗敏、邹荃等对本书提出了很多很好的建议，电子科技大学的刘睿等参与了本书的校对工作，另外要感谢我的家人，本书的出版离不开他们的支持和帮助。

由于作者水平有限，疏漏之处敬请广大读者指正。

作 者
2005 年 1 月

目 录

第一章 构建 Java 编程环境	(1)
1.1 引言和本章任务	(1)
1.2 基础知识链接	(1)
1.2.1 Java 开发工具包 (JDK)	(1)
1.2.2 Application 和 Applet 的编辑、编译和运行	(2)
1.2.3 Java 语言的一些规定	(5)
1.2.4 软件打包 (jar) 和软件发布	(6)
1.3 案例 1 及案例分析: 构建 Java 编程环境	(6)
1.3.1 安装 JDK	(6)
1.3.2 安装 JCreator	(8)
1.3.3 Java 工具使用	(11)
1.4 扩展知识	(15)
1.4.1 Java 语言的特点	(15)
1.4.2 基本数据类型	(16)
1.4.3 控制语句	(16)
1.4.4 数组	(21)
1.4.5 Java 集成开发环境	(23)
1.5 本章小结	(24)
1.6 能力训练题	(24)
1.7 上机实验	(24)
第二章 Java 核心思想: 面向对象	(26)
2.1 引言和本章任务	(26)
2.2 基础知识链接	(27)
2.2.1 面向对象思想	(27)
2.2.2 类和对象	(27)
2.2.3 类的 Java 实现	(30)
2.2.4 继承	(32)
2.2.5 访问控制	(33)
2.2.6 多态性	(35)
2.3 案例 2 及案例分析: 点、圆、圆柱体	(37)
2.4 扩展知识	(41)
2.4.1 包	(41)
2.4.2 非访问控制符 (static, final, abstract)	(41)

2.4.3	接口	(44)
2.4.4	this 和 super	(46)
2.4.5	内部类和匿名类	(47)
2.4.6	Object 类和 System 类	(48)
2.5	本章小结	(49)
2.6	能力训练题	(49)
2.7	上机实验	(50)
第三章	Java 异常处理	(51)
3.1	引言和本章任务	(51)
3.2	基础知识链接	(53)
3.2.1	异常的概念	(53)
3.2.2	异常的分类	(53)
3.2.3	捕获和处理异常	(54)
3.2.4	抛出异常	(55)
3.3	案例 3 及案例分析: 求阶乘的异常处理	(56)
3.4	扩展知识	(58)
3.4.1	自定义异常类	(58)
3.4.2	异常类提供的常用方法	(59)
3.5	本章小结	(60)
3.6	能力训练题	(60)
3.7	上机实验	(61)
第四章	Java 图形界面编程	(62)
4.1	引言和本章任务	(62)
4.2	基础知识链接	(62)
4.2.1	基本的 AWT 组件	(62)
4.2.2	Java 布局管理器	(68)
4.2.3	Java 事件模型	(77)
4.3	案例 4 及案例分析: 简单的计算器程序	(81)
4.4	扩展知识	(88)
4.4.1	Swing 编程	(88)
4.4.2	Applet 编程	(94)
4.5	本章小结	(96)
4.6	能力训练题	(96)
4.7	上机实验	(97)
第五章	Java 数据结构编程	(98)
5.1	引言和本章任务	(98)
5.2	基础知识链接	(98)

5.2.1 线性数据结构	(98)
5.2.2 排序和查找	(108)
5.3 案例 5 及其分析：八皇后问题	(111)
5.3.1 问题描述和分析	(111)
5.3.2 具体实现分析	(112)
5.4 扩展知识	(114)
5.4.1 基本数据类型的包装类	(114)
5.4.2 Math 类	(115)
5.4.3 树	(116)
5.4.4 散列表	(117)
5.4.5 迭代和递归	(120)
5.5 本章小结	(121)
5.6 能力训练题	(121)
5.7 上机练习	(122)
第六章 Java 文件和流	(123)
6.1 引言和本章任务	(123)
6.2 基础知识链接	(123)
6.2.1 Java 文件与流概述	(123)
6.2.2 标准输入输出	(124)
6.2.3 字节流	(126)
6.2.4 字符流	(130)
6.3 案例 6 及其分析：对文件进行加密	(130)
6.4 扩展知识	(135)
6.4.1 随机访问文件	(135)
6.4.2 File 类	(137)
6.5 本章小结	(138)
6.6 能力训练题	(138)
6.7 上机实验	(139)
第七章 Java 多线程编程	(140)
7.1 引言和本章任务	(140)
7.2 基础知识链接	(140)
7.2.1 线程与进程	(140)
7.2.2 Java 线程创建方式	(141)
7.2.3 Java 线程状态与生命周期	(145)
7.2.4 Java 线程优先级	(146)
7.2.5 Java 线程同步机制	(146)
7.3 案例 7 及其分析：时钟显示	(151)

7.4	扩展知识	(156)
	管程机制	(156)
7.5	本章小结	(159)
7.6	能力训练题	(159)
7.7	上机练习	(161)
第八章	Java 网络编程	(162)
8.1	引言和本章任务	(162)
8.2	基础知识链接	(162)
	8.2.1 Java 与网络	(162)
	8.2.2 Java 基本的网络功能	(163)
	8.2.3 Java 套接字编程	(167)
8.3	案例 8 及分析: SMTP 邮件传送客户端	(179)
8.4	扩展知识	(183)
	编写多客户服务器程序	(183)
8.5	本章小结	(186)
8.6	能力训练题	(186)
8.7	上机实验	(186)
第九章	综合应用案例一: 五子棋游戏	(187)
9.1	引言和本章任务	(187)
9.2	基础知识链接	(187)
	9.2.1 五子棋简介	(187)
	9.2.2 五子棋算法分析	(189)
9.3	案例 9 及其分析: 五子棋游戏	(190)
	9.3.1 案例 9 的类层次结构	(190)
	9.3.2 案例 9 源代码分析	(191)
9.4	本章小结	(213)
9.5	课程设计	(213)
第十章	综合应用案例二: 俄罗斯方块	(214)
10.1	引言和本章任务	(214)
10.2	基础知识链接	(214)
	10.2.1 俄罗斯方块简介	(214)
	10.2.2 俄罗斯方块算法分析	(215)
10.3	案例 10 及案例分析: 俄罗斯方块	(215)
	10.3.1 案例 10 类层次结构	(215)
	10.3.2 案例 10 源代码分析	(217)
10.4	本章小结	(250)
10.5	课程设计	(251)

第一章 构建 Java 编程环境

1.1 引言和本章任务

在学习其它语言如 C 语言，通常都是以一个简单的程序入手的，就是在终端上打印出字符串“Hello world”。那么在 Java 编程环境下如何实现在终端上打印“Hello world”的字符串呢？

为了编辑、编译、调试的方便，其它语言通常都有各种平台的开发环境。那么如何构建适合自己的 Java 开发环境，以及在这样的开发环境中如何编辑、编译、运行自己编写的 Java 代码呢？

通过本章的案例分析和相关知识介绍，读者可以很轻松地完成如下任务：

1. 构建 Java 编辑、编译、运行环境；
2. 了解 Java 编译、运行机制以及相关工具包和获取开发帮助；
3. 可以开发最基本的 Java Application 和 Java Applet。

1.2 基础知识链接

在构建 Java 编程环境之前，读者必须了解相关的基础知识。下面分章节一一介绍这些相关的基础知识。

1.2.1 Java 开发工具包 (JDK)

JDK 是 Java 语言的基本工具包，它是 SUN 公司免费提供的开发、运行 Java 程序的基本软件。该软件可以从 <http://java.sun.com> 网站免费下载，读者可以根据自己的操作平台选取相应的 JDK 软件。JDK 安装后的目录结构如下：

- bin：存放运行程序；
- demo：存放一些示例程序；
- include：存放与 C 相关的头文件；
- jre：存放 Java 运行环境相关的文件；
- lib：存放程序库；
- Src.zip：含有 Java 库程序的源程序。

JDK 为编程人员提供了 Java 基础类库 (JFC), JFC 包含很多包, 在每个包中都有若干个具有特定功能和相互关系的类和接口:

1. java.lang 包: Java 基本语言包, 包含了运行 Java 程序必不可少的系统类, 每个 Java 程序运行时, 系统都会自动地引入 java.lang 包;

2. java.io 包: Java 语言的标准输入/输出包;

3. java.util 包: Java 语言的一些低级实用工具, 如处理时间的 Date 类等;

4. java.awt 包: Java 语言图形工具包;

5. java.applet 包: Java 语言小应用程序包;

6. java.net 包: Java 语言网络处理包。

另外 JDK 还提供了其它包, 如 java.sql (Java 数据库处理包), java.rmi (Java 远程方法调用包), java.security (Java 安全控制和管理包)。

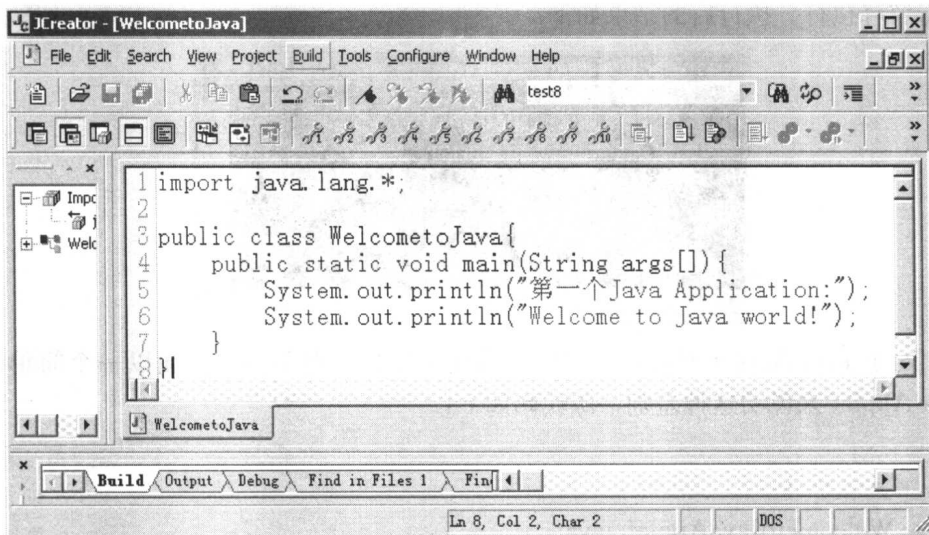
1.2.2 Application 和 Applet 的编辑、编译和运行

Java 程序包括两种基本的程序, 一种是 Java Application 程序 (又称为应用程序), 另一种是 Java Applet 程序 (又称为小应用程序)。

下面以一个简单的 Java Application 程序来说明 Java Application 的编辑、编译和运行, 该程序的源代码如下:

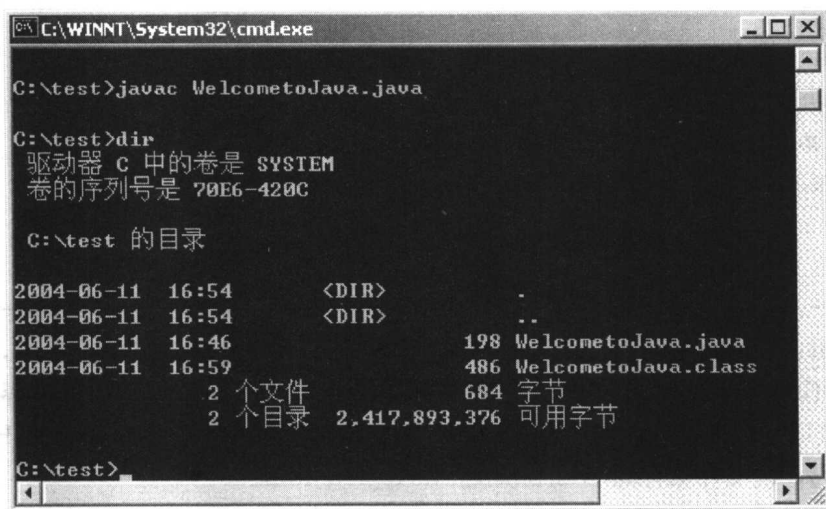
```
1  /* *
2   * WelcometoJava.java
3   */
4  import java.lang.*;
5
6  public class WelcometoJava {
7      public static void main (String args []) {
8          System.out.println (" 第一个 Java Application:");
9          System.out.println (" Welcome to Java world!");
10     }
11 }
```

编辑: 对于 Java Application 程序的编辑来说, 可以利用一些集成开发工具, 在这些集成开发工具中进行编写, 也可以直接利用 Windows 平台所提供的记事本 (Notepad) 软件。本书以 JCreator 为例 (关于 JCreator 的安装和使用见本章的相关介绍), 编辑后的界面如下:



假定该文件保存为 C:\test\WelcometoJava.java。

编译：可以直接在 Dos 命令提示符下利用 Java 的编译工具（javac）来编译源程序，主要命令为：javac WelcometoJava.java，下面是编译的 Dos 界面：

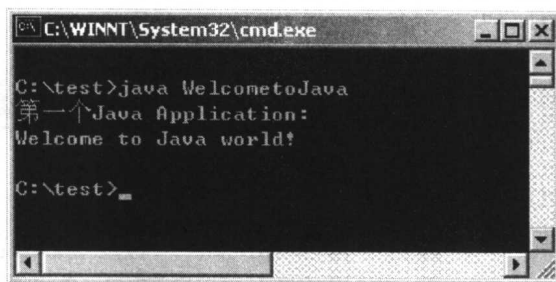


在 JCreator 集成开发环境中，也可以利用该集成环境所提供的编译工具：打开“Build”菜单，然后选中“Compile File”，该工具可以调用配置好的 Java 编译工具进行编译。

运行：可以直接在 Dos 命令提示符下利用 Java 的运行工具（java）来解释执行编译好的程序，主要命令为：java WelcometoJava（注意：不要写成 java WelcometoJava.class）。

也可以利用 JCreator 集成开发环境中的运行工具来解释执行：打开“Build”菜单，然后选中“Execute File”，该工具可以调用配置好的 Java 解释执行工具

进行解释执行。执行后的结果如下：



对于 Java Applet 而言，其编辑、编译和运行略有不同，下面以一个简单的 Java Applet 来说明这种差别，源代码如下：

```
1  /* *
2  * WelcometoJavaApplet.java
3  * /
4  import java.applet.*;
5  import java.awt.*;
6
7  public class WelcometoJavaApplet extends Applet {
8      public void paint (Graphics g) {
9          g.drawString (" 第一个 Java Applet:", 20, 20);
10         g.drawString (" Welcome to Java world!", 20, 40);
11     }
12 }
```

编辑：Applet 除了编写小程序外，还需要编写对应的 HTML 页面，因为 Applet 不像 Application 那样支持字符界面的处理，Applet 是基于图形界面的，它需要 HTML 容器来存放 Applet 应用程序，下面是该例子代码对应的 HTML 页面代码，该文件保存为 WelcometoJavaApplet.html：

```
1  <HTML>
2  <HEAD>
3  <TITLE>第一个 Applet 程序</TITLE>
4  </HEAD>
5  <BODY>
6  <APPLET CODE = WelcometoJavaApplet Width= 300 Height =200
7  ></APPLET>
8  </BODY>
```

编译：Applet 的编译只编译 Applet 源代码，不需要编译 HTML 页面代码，Applet 的编译和 Application 的编译相同，都是使用 Java 编译工具（javac）进行编译。

运行：运行 Applet，可以使用小程序查看器 appletviewer，在 Dos 命令行下输入如下命令：appletviewer WelcometoJavaApplet.html，运行的结果如下：



也可以利用浏览器直接点击 WelcometoJavaApplet.html 来运行 Applet。

1.2.3 Java 语言的一些规定

Java 语言与其它语言相比，有一些较为特殊的规定：

1. Java 应用程序必须有一个类中定义 main () 方法，而对 applet 来说，它必须作为 Applet 的一个子类；

2. Java 语言规定在一个 .java 源文件中，最多只能有一个被声明为 public 的类，如果一个 java 源文件中包含了一个 public 类，则该文件名必须与类名一致；

3. 一个 .java 源文件可以包含：

(1) 一个可选的包声明，如 package java.io；

(2) 任意多个 import 语句，如 import java.io.*；

(3) 类和接口的定义，如 public class HelloWorld。这三项出现的顺序必须是固定的，如果没有在第一句指定 package，则该类将归属一个缺省包；

4. 在 Java 语言中，有三种格式的注释信息：

(1)//：用于单行注释，只对当前行有效，可以出现在一行的开始、中间乃至该行的结束；

(2) /* */：用于多行注释，在此之间的字符将被编译器忽略；

(3) /* * 文档注释注释信息 */：用于 Java 文档的注释；

5. 在 Java 语言中，必须在语句结尾处标记一个分号表示语句的结束，一个程序段必须用一对大括号括起来，程序段可以嵌套；

6. 在 Java 中, 标识符以字母、下划线或 \$ 符号起始, 后面可以是上述三种符号或数字。

1.2.4 软件打包 (jar) 和软件发布

对于一个较大的工程软件来说, 该工程可能包括多个 .class 文件及其相关的文件 (如图像文件等), 在进行软件发布时, 基于安全和压缩存储空间的考虑, 需要对这些文件打包并压缩成一个文件, 这时候需要一个 Java 打包工具 (jar)。

下面举一个例子来说明该工具的使用:

```
jar cvf myJar.jar Hello.class Person.class
```

上面的例子将 Hello.class 和 Person.class 打包成 myJar.jar 文件, 要执行 jar 文件, 可利用 java 命令中的一 jar 选项, 如:

```
java -jar myJar.jar
```

1.3 案例 1 及案例分析: 构建 Java 编程环境

1.3.1 安装 JDK

为了鼓励更多的人使用 Java 语言开发软件, SUN 公司向大家免费提供了 Java 语言的软件开发工具包 (Java Develop Kit, JDK), 它包含了所有编写、运行 Java 程序所需要的工具: Java 基本组件、库、Java 编译器、Java 解释器、小应用程序浏览器、以及一些用于开发 Java 应用程序的程序等。

JDK 提供了一个"没有任何装饰"的编程引擎, 它既没有一步步的说明, 更没有华丽的引擎, 你需要在 Dos 窗口下一个字符一个字符地敲入命令行完成整个工具。

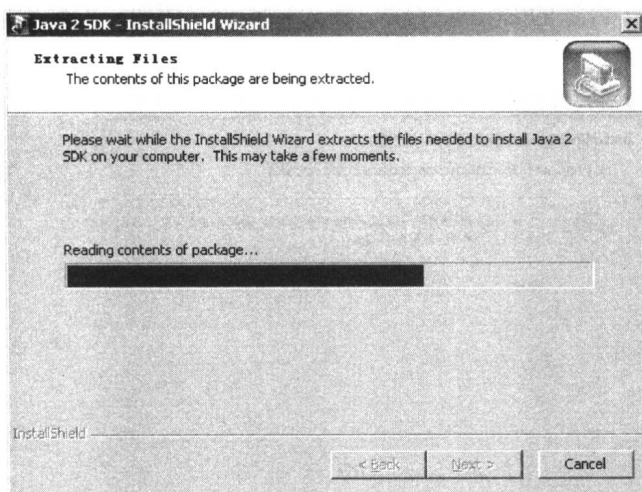
准备 Java 的编程环境, 也就是找到 Java 语言的软件开发工具包 JDK, 然后安装到系统中去。

SUN 公司提供了 Macintosh、Solaris、Windows 平台的 JDK, 在本书中我们以 Windows 平台为例。一般来说, 可以通过两种途径找到 JDK:

1. 从光盘中获取;
2. 通过 Internet 下载, 网址为: <http://java.sun.com>。

准备好 JDK 软件后, 用户就可以安装 JDK 了, 下面以 Windows 平台为例来说明 JDK 的软件的安装, 安装步骤如下:

- (1) 双击 Windows 下的 JDK 软件: j2sdk-1_4_2_03-windows-i586.exe, 开始安装;
- (2) 安装程序首先进行解压缩, 如下图所示:



(3) 然后，安装程序开始准备安装；

(4) 然后将出现一个欢迎安装 JDK 的界面，单击“Next”按钮继续；

(5) 安装程序提示读者是否认可“软件许可协议”，单击“Yes”按钮继续；

(6) 接下来，为 JDK 指定安装目录，缺省是 c:\j2sdk1.4.2_03，如果读者想修改，就单击“Browse”按钮，否则按“Next”按钮继续；

(7) 接着，安装程序将要求读者选择要安装的组件；如下图所示，在想安装的组件前面的方框上打上一个勾。其中包括：

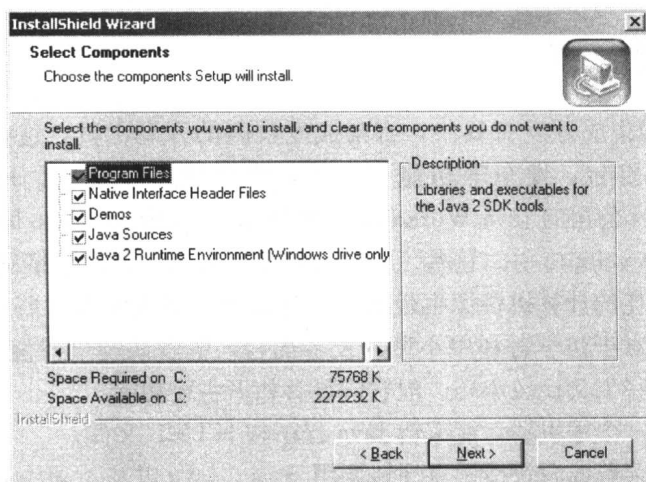
Program Files：这是 JDK 的主体文件，必须选中；

Native Interface Header Files：本地接口头文件，必须选中；

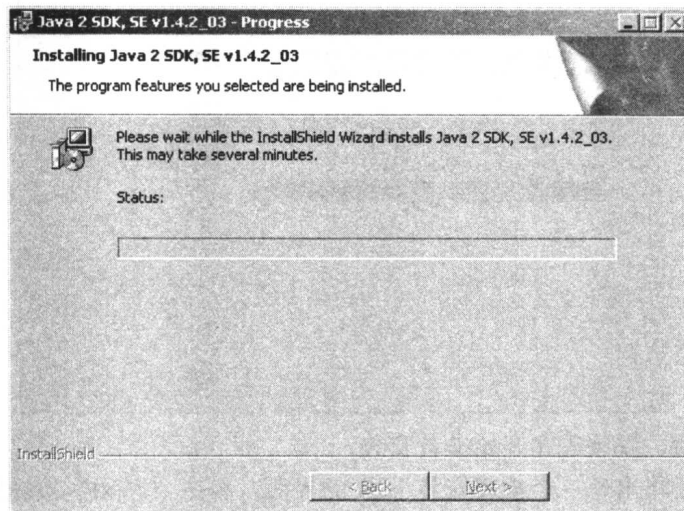
Demos：一些关于 Java 的演示，可以不选；

Java Sources：一些 Java 的示例源程序，可以不选；

Java 2 Runtime Environment (Windows drive only)：Java 运行时间环境，建议选中。



(8) 选择了合适的安装组件后，单击“Next”按钮，安装程序就开始复制文件了，如下图所示；



(9) 最后，按“Finish”按钮，结束安装。

安装完毕后，读者可以在 `c:/j2sdk1.4.2_03` 看到如下目录文件：

bin：该目录存放运行程序（如 `java`、`javac`、`appletviewer` 等）；

demo：该目录存放一些 Java 的示例文件；

include：该目录存放与 C 相关的头文件；

jre：该目录存放 Java 运行环境相关的文件；

lib：该目录存放程序库；

src.zip：该文件是压缩文件，读者可以将该文件解压缩，该文件经过解压缩后可以看到 Java 库程序的源代码。

整个 JDK 安装完后，将占用 80M 左右的磁盘空间。

1.3.2 安装 JCreator

在 Java 的集成开发工具中，有集成度较高的开发工具，这些工具集编辑、编译、调试、打包、发布等功能于一体，常见这些工具有：Borland 公司的 Jbuilder，SUN 公司的 Java Workshop，IBM 公司的 Visual Age for Java，Oracle 公司的 Java Developer 等。这些工具集成度虽然较高，但通常情况下运行这些集成开发工具消耗的计算机资源也较大，并且在短时间内掌握这些开发工具较难。通常情况下，对于初学者和中小型开发者而言，笔者认为最好使用一些 Java 的辅助工具，这些工具比较小巧，但它们具有如下主要功能：

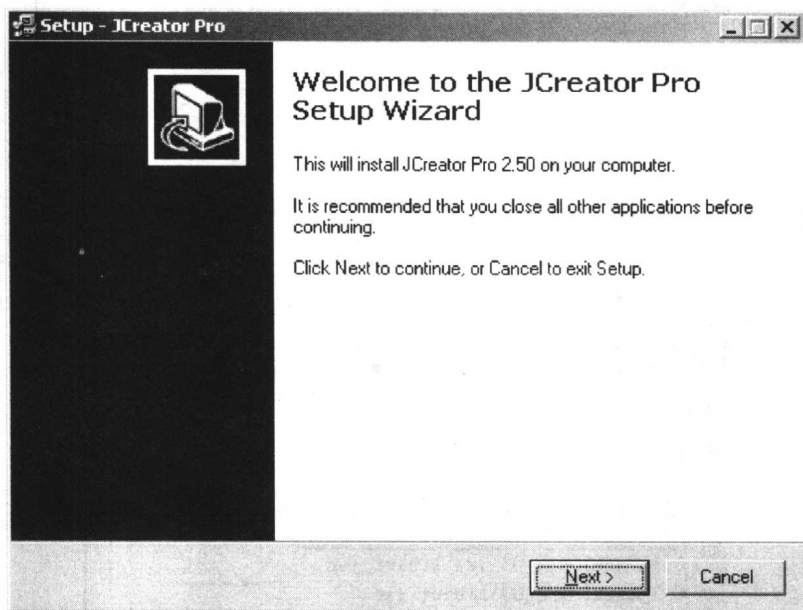
(1) 提供一个编辑器，能编辑 Java 程序和 HTML 文件；

(2) 通过菜单或者快捷键方便地调用 `javac`、`java` 以及 `appletviewer` 来编译和

运行 Java 程序。

在这些辅助工具中，笔者推荐使用 JCreator，读者可以通过光盘或者 Internet 获取该软件，该软件有免费版和共享版两个版本，对于共享版而言，它相对于免费版而言多提供了很实用的功能，即能即时提示各个类的方法名以及相关的参数，这样可以大大提高程序输入的速度并减少输入的错误。下面介绍该软件的安装步骤和使用。

(1) 通过光盘或者 Internet 获取该软件 JCreator Pro2.50，双击该文件夹下的 setup.exe，开始安装，如下图所示；



(2) 安装程序提示读者是否认可“软件许可协议”，单击“I accept the agreement”按钮继续；

(3) 紧接下来，为 JCreator 指定安装目录，缺省是“C:\Program Files\Xinox Software\JCreator Pro”，如果读者想修改，就单击“Browse”按钮，否则按“Next”按钮继续，安装程序就开始复制文件了；

(4) 最后，按“Finish”按钮，结束安装。

下面介绍 JCreator 基本的使用：

(1) 在使用 JCreator 编辑 Java 程序之前，读者必须进行一些基本的配置，首先应该配置的是 JDK Profiles，配置方法如下：

打开 JCreator，选择 Configure 菜单中的 Options 菜单项，如下图所示。