“十三五”普通高等教育规划教材

Java程序设计及应用开发

宋　晏　杨国兴　主　编
胡倩茹　陈晓美　副主编

提供电子教案和习题解答
http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

“十三五”普通高等教育规划教材

Java 程序设计及应用开发

宋　晏　杨国兴　主　编
胡倩茹　陈晓美　副主编

机 械 工 业 出 版 社

本书以 Java SE 6 为基础，按照从面向对象的语言走进面向对象的思想、利用图表增强文字的表现力、注重知识的原理性的编写思想，详细叙述了 Java 语言的基础知识，面向对象的封装、类、继承、多态性，Java 常用工具类、集合、异常处理，及图形用户界面、多线程、输入/输出流、JDBC 等内容。

本书配备了丰富的实例，并在“综合实践”部分引入较大规模的案例，通过“习题”和“实验指导”环节，为读者提供拓展思维、提升实践能力的训练。各章习题参照了 SCJP 考试模式，实验题目丰富、实用，有的放矢地提供编程训练。

本书可以作为大学本科、专科计算机及相关专业的教材，也可作为 Java 爱好者、工程技术人员的自学参考书。

本书配有电子教案，需要的教师可登录 www. cmpedu. com 免费注册，审核通过后下载，或联系编辑索取（QQ：2966938356，电话：010 - 88379739）。

图书在版编目（CIP）数据

Java 程序设计及应用开发/宋晏，杨国兴主编 . —北京：机械工业出版社，2016. 8
“十三五”普通高等教育规划教材
ISBN 978-7-111-54291-9

Ⅰ. ①J… Ⅱ. ①宋… ②杨… 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（2016）第 161139 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：和庆娣 责任编辑：和庆娣
责任校对：张艳霞 责任印制：李 洋
北京宝昌彩色印刷有限公司印刷
2016 年 8 月第 1 版 · 第 1 次印刷
184mm × 260mm · 21. 5 印张 · 530 千字
0001—3000册
标准书号：ISBN 978-7-111-54291-9
定价：49.90元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：(010)88379833
读者购书热线：(010)88379649

网络服务
机 工 官 网：www. cmpbook. com
机 工 官 博：weibo. com/cmp1952
教育服务网：www. cmpedu. com
金 书 网：www. golden - book. com

前　言

Java 语言的生命力毋庸置疑，1991 年由 Sun 公司（后被 Oracle 公司收购）开发。2014 年，Java 发布了 Java SE 8 版本，它以优秀的性质驰骋在各个领域，以开源建设的方式不断地为其注入新鲜血液。

思考了许久怎样写此书才能打好 Java 的根基，为今后 Struts、Spring、EJB 等的学习奠定坚实的基础；如何能让年轻的学习者们轻松、高效地完成学习，不会感觉代码是枯燥冰冷的字符，而是悦动在指尖的一串串的音符…。带着让学习者以享受的姿态步入 Java 程序员行列的希冀，最终确定了如下编写思想。

（1）从面向对象的语言走进面向对象的思想

任何一门计算机语言的学习都不仅仅是熟知语法的过程。计算机语言的语法就如音乐中的音符，它们会在不同人的笔下诞生奇妙的乐谱，那是作曲家赋予音符的灵魂。面向对象的思想就是面向对象语言的灵魂。

本书在讲述 Java 语法知识的同时，更注重面向对象思想的学习和贯彻。从面向对象分析出发，使用面向对象工具 UML 类图描述类结构及类与类之间的关系；在系统设计和组织程序架构时，引入面向对象设计中的经典原则和设计模式。从学习伊始就培养面向对象的视角和规范的编程方式，不仅要写出代码，而且要写出专业、漂亮的代码。

（2）使用图表增强文字的表现力

相对于文字而言，图可以更形象、立体地展示知识及彼此间的联系，表可以梳理、对比相关、相似的知识点。相信读者间都会有一种共识，如果面对一份长篇大论，那么你的关注点首先会集中到穿插在文字中的图或表，因为从图表中可以快速提取到文字的主旨、脉络和精华；而且我也在猜想，从小看漫画长大的年轻一代会对图表具有更高的敏感度。

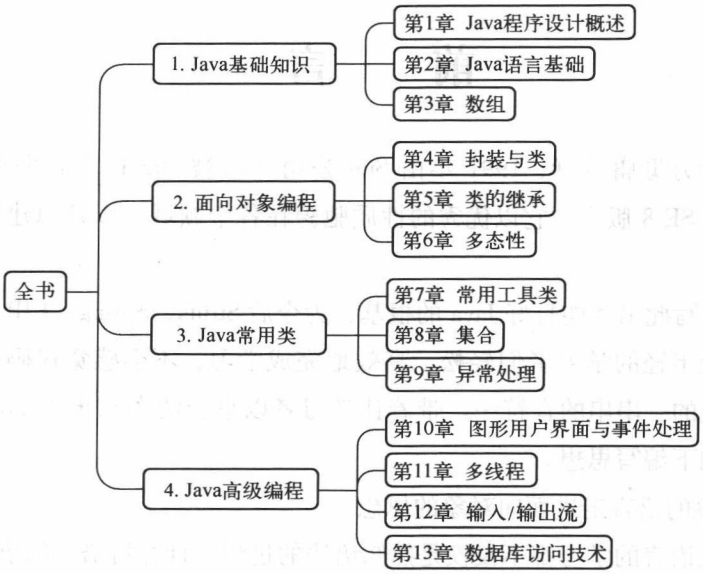
本书尽可能地为抽象、不易单纯通过语言表述清楚、信息量大、知识庞杂的部分设计了图表，力求简明扼要地展示知识结构。

另外，本书各章都使用**思维导图**从更高的角度对整章知识、案例进行了梳理，将看似零散的文字浓缩在一张图中，提纲挈领，将知识从点连接成线，再构建为面，最终立体化，达到读书“从物理上将书变厚，从逻辑上将书变薄”的效果。

（3）知其然亦知其所以然

坚实的基础是进阶的基石。本书注重知识背后隐藏的原理和细节，培养读者从 why 和 how 的角度构建学习的习惯，使学习不仅知其然，更能做到知其所以然，以扎实的基本功为后续的学习打好根基。

本书架构如下：



本书由宋晏、杨国兴主编，胡倩茹（河北大学）、陈晓美副主编，参加编写工作的还有刘勇、严婷、吕东艳、朱红、张子萍、张小静。

由于时间仓促，书中难免有疏漏和不足之处，敬请广大读者批评指正。

编　者

目　录

第1章　Java程序设计概述

本章按照了解Java发展历史、平台结构、特性、程序设计环境，以及学习编写简单的Java应用程序的路径，开始Java语言的学习。

1.1　Java程序平台

Java是由Sun公司（后被Oracle公司收购）于1991年开发的编程语言，初衷是为家用消费类电子产品开发一个分布式代码系统。为了使整个系统与平台无关，采用了虚拟机器码方式，所以，Java从诞生之日起就成为了平台无关的语言。

Java分为3个体系：Java SE（Java2 Platform Standard Edition，Java平台标准版），Java EE（Java 2 Platform Enterprise Edition，Java平台企业版），Java ME（Java 2 Platform Micro Edition，Java平台微型版）。

（1）Java SE

Java SE（以前称为J2SE）是允许开发和部署在桌面、服务器、嵌入式环境、实时环境中使用的Java应用程序。Java SE包含了支持Java Web服务开发的类，并为Java EE提供基础。

（2）Java EE

Java EE（以前称为J2EE）是帮助开发和部署可移植、健壮、可伸缩且安全的服务器端Java应用程序。Java EE是在Java SE的基础上构建的，它提供Web服务、组件模型、管理和通信API，可以用来实现企业级的面向服务体系结构和Web 2.0应用程序。

（3）Java ME

Java ME（以前称为J2ME），为在移动设备和嵌入式设备（比如手机、PDA、打印机等）上运行的应用程序提供一个健壮且灵活的环境。Java ME包括灵活的用户界面、健壮的安全模型、内置的网络协议以及对可动态下载的应用程序的支持。

1.2　Java的特性

Java是一种编程语言，拥有跨平台的特性，并以开源的方式得到众多开发者的支持。Java语言具有简单的（simple）、面向对象的（object - oriented）、网络的（network - savvy）、健壮的（robust）、安全的（secure）、可移植的（portable）、解释型的（Interpreted）、高性能的（high - performance）、多线程的（multithreading）、动态的（dynamic）等特性。

1. 简单性和健壮性

C/C ++语言具有非常强的生命力，然而C/C ++中一些功能却耗费了相当的学习成本、开发成本和维护成本，有些特性带来的麻烦远远多于其带来的好处，诸如著名的指针运算、运算符重载、多重继承、内存管理等。Java的设计者舍弃了C/C ++中较少使用、难以掌握或可能不安全的功能，在许多常用特性上加以简化，并提供丰富的类库。

以指针和内存管理为例，Java 从语法的角度屏蔽了指针的概念，它保留了指针的原理，允许使用地址（通过引用类型的变量），但不允许操作指针重写内存，消除了损坏数据的可能性。

2. 面向对象

Java 语言是纯面向对象的语言，即便是只有一个 main()方法也需要用一个类封装。Java 摒弃了 C++中的多继承，取而代之的是“接口”。用面向对象的方式设计和解决问题并不是一件简单的事情，在本书的第 4～6 章将详细学习 Java 面向对象的特性和设计方法。

3. 网络特性

Java 的网络能力非常强大且易于使用，它提供的类库可以便捷地处理 HTTP 和 FTP 等 TCP/IP。Java 应用程序能够通过 URL 打开和访问网络上的对象，就如同访问本地文件一样。Java 应用最广泛的领域就是网络服务。

4. 安全性

Java 适用于网络/分布式环境。Java 的垃圾回收机制用更安全的方式解决了资源的回收问题。异常处理架构使开发人员可以掌控程序中各种突发的异常状况。final、synchronized 等关键字的使用也都在共享方面加强了安全性。

5. 可移植性

程序跨平台并不是一件容易的事情。平台可以指计算机体系结构（Architecture），可以指操作系统（Operating System），也可以指开发平台（编译器、链接器等）。

不同的计算机体系结构有不同的指令集，可以识别的机器指令格式是不同的。以 C 程序为例，因为开发人员不清楚 C 程序未来的使用环境，所以在编译时要利用不同操作系统下的不同体系结构的计算机的 C 编译器，把 C 程序编译成各种不同的机器指令，由客户选择不同的版本去执行。例如，操作系统中的硬件驱动程序多数由 C 语言编写，编好后会针对不同的平台进行编译，用户在使用时找到与当前操作系统相匹配的版本进行安装。

这意味着用 C 语言编写的程序只需稍加修改甚至不用修改就可以在各种不同的计算机上编译运行，C 语言实现的是源代码级的跨平台。

Java 利用 Java 虚拟机（Java Visual Machine，JVM）机制实现了 Java 程序在操作系统和体系结构级别的跨平台，做到了“一次编译，到处执行”。

如图 1-1 所示，Java 程序在编译时并不直接生成与机器相关的指令，而是编译生成 Java 虚拟机可以读懂的字节码（Bytecode）文件。Java 虚拟机位于操作系统之上，可以理解为一个以字节码为机器指令的软件 CPU，它屏蔽了底层操作系统的差异，使 Java 应用程序可以在安装了 JVM 的任何计算机系统上运行，实现了良好的可移植性。字节码文件因为是由二进制代码组成，所以传播也更加安全。

如图 1-1，Java 虚拟机的体系结构主要包括类加载器（ClassLoader）、执行引擎（Execution engine）、垃圾收集器（Garbage Collector）3 部分。JVM 在工作时操作运行时数据区，包括多线程共享的方法区和堆内存区，以及每个线程都具备的程序计数器、JVM 栈内存区和本地方法栈（使用 Java 语言以外的其他语言编写的方法的工作区）等。

JVM 类加载器的工作流程是：首先负责找到二进制字节码并加载至 JVM 中，然后由链接过程对二进制字节码的格式进行校验、初始化装载类中的静态变量以及解析类中调用的接口、类，最后按需完成类中的静态初始化代码、构造方法代码等初始化工作。

JVM 执行引擎的主要技术包括解释、即时编译（Just In Time，JIT）、自适应优化等。解释

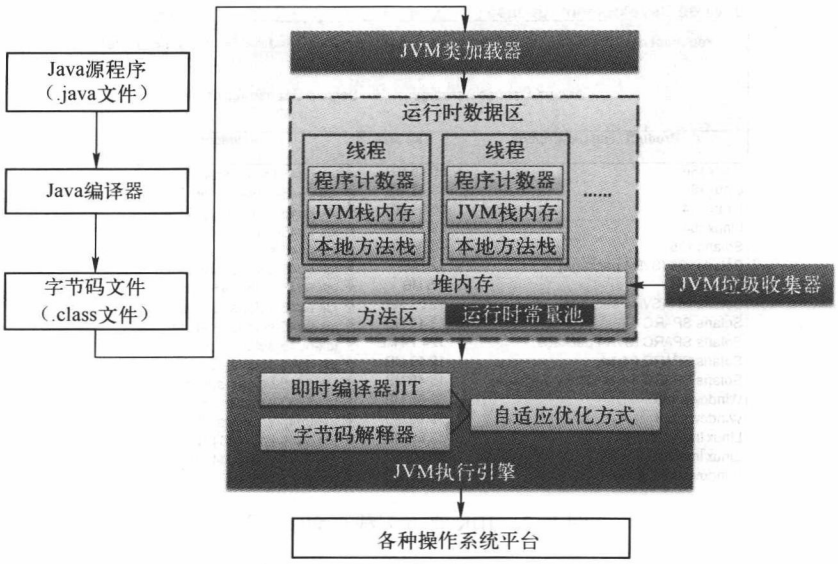


图 1-1　Java 程序的运行过程

执行属于第一代 JVM，现场解释执行，不生成目标程序；即时编译属于第二代 JVM，在运行时将字节码翻译为机器码；自适应优化则吸取第一代 JVM 和第二代 JVM 的经验，采用两者结合的方式，是目前 Sun 的 HotspotJVM 采用的技术。

自适应优化执行引擎开始对所有的代码都采取解释执行的方式，并监视代码执行情况，然后对那些经常调用的方法启动一个后台线程，将其编译为本地代码，并进行优化。若方法不再频繁使用，则取消编译过的代码，仍对其进行解释执行。

1.3　Java 程序设计环境

Java 开发环境大体分成两种方式：一种方式是使用 JDK 的命令行方式，另一种是使用集成开发环境。

1.3.1　下载、安装和了解 JDK

用户在网站 http://www.oracle.com/technetwork/java/index.html 可以免费下载适合于不同计算机操作系统的 Java 开发工具包（Java Development Kit，JDK），本书使用 Java SE 6 版本。下载 JDK 时要选择与当前操作系统对应的安装文件，如图 1-2 所示。

下载并按照向导安装 JDK 后，Java SE 的体系结构如图 1-3 所示。

主要生成如下目录。

\bin：Java 开发工具所在目录，包括 Java 编译器（javac.exe）、解释器（java.exe）等。

\demo：在该目录下 Sun 提供了一些实例程序。

\lib：该目录下存储了 Java 开发工具要用的类库，例如包含了支持 JDK 工具的类库 tool.jar 等。

\jre：Java 自己附带的运行环境（Java Runtime Environment，JRE），包括 Java 虚拟机、运行类库等，向用户编写的 Java 应用程序提供运行环境。

Java SE Development Kit 6u45

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

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	65.46 MB	jdk-6u45-linux-i586-rpm.bin
Linux x86	68.47 MB	jdk-6u45-linux-i586.bin
Linux x64	65.69 MB	jdk-6u45-linux-x64-rpm.bin
Linux x64	68.75 MB	jdk-6u45-linux-x64.bin
Solaris x86	68.38 MB	jdk-6u45-solaris-i586.sh
Solaris x86 (SVR4 package)	120 MB	jdk-6u45-solaris-i586.tar.Z
Solaris x64	8.5 MB	jdk-6u45-solaris-x64.sh
Solaris x64 (SVR4 package)	12.23 MB	jdk-6u45-solaris-x64.tar.Z
Solaris SPARC	73.41 MB	jdk-6u45-solaris-sparc.sh
Solaris SPARC (SVR4 package)	124.74 MB	jdk-6u45-solaris-sparc.tar.Z
Solaris SPARC 64-bit	12.19 MB	jdk-6u45-solaris-sparcv9.sh
Solaris SPARC 64-bit (SVR4 package)	15.49 MB	jdk-6u45-solaris-sparcv9.tar.Z
Windows x86	69.85 MB	jdk-6u45-windows-i586.exe
Windows x64	59.96 MB	jdk-6u45-windows-x64.exe
Linux Intel Itanium	53.89 MB	jdk-6u45-linux-ia64-rpm.bin
Linux Intel Itanium	56 MB	jdk-6u45-linux-ia64.bin
Windows Intel Itanium	51.72 MB	jdk-6u45-windows-ia64.exe

图 1-2　JDK 官方下载页面示例

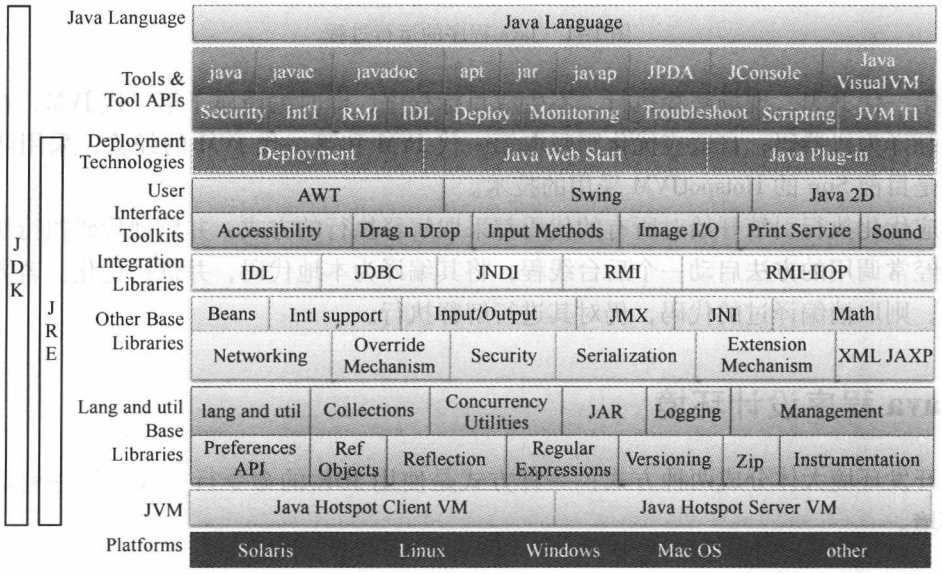


图 1-3　Java SE 体系结构

【注意】在安装目录中有一个 src. zip 文件，解压后得到 src 文件夹，里面存放了所有的类库源文件，通过它就可以看到所有的 Java 提供类库的源代码，这里既是走近 Java 的途径，也是非常好的学习资源。

这些目录中，\bin 目录非常重要，因为编写完 Java 程序后，无论是编译还是执行程序，都会用到\bin 目录下提供的工具程序。Java 开发工具主要包括以下几种。

- javac. exe：Java 编译器，用来将 Java 程序编译为字节码文件。
- java. exe：Java 解释器，执行已经转化为字节码的 Java 应用程序。
- jdb. exe：Java 调试器，用来调试 Java 程序。
- javap. exe：Java 反编译器，将字节码文件还原为源文件。
- javadoc. exe：文档生成器，创建 HTML 文件。

JDK 安装完毕后，虽然安装者知道 JDK 的工具程序位于\bin 目录下，但是操作系统并不

知道这件事。为了方便使用工具程序，通常将开发工具路径（\bin 文件夹的完整路径）加入到操作系统的环境变量 Path 中。Path 变量告诉操作系统都可以到哪些目录下尝试找到当前要使用的工具程序。

在“计算机”上右击，从快捷菜单中选择“属性→高级属性”命令，打开“系统属性”对话框。单击“环境变量”按钮，在弹出的“环境变量”对话框中编辑 Path 变量，在“变量值”文本框中加入\bin 目录的完整路径，如图 1-4 所示，新加入的路径与原 Path 变量中的路径以分号“;”连接。

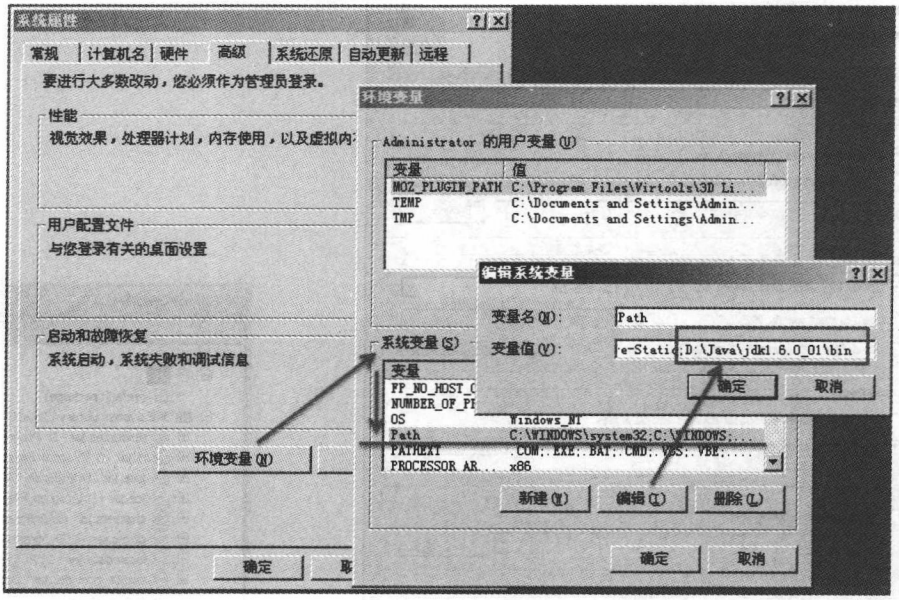


图 1-4　向环境变量 Path 加入 Java 开发工具路径的方法

设置 Path 变量后，要重新打开命令行模式才能重新读入 Path 变量的取值。在命令行的任意提示符下输入“javac”并按〈Enter〉键，如果都能看到关于 javac 命令的使用说明则说明 Path 的设置成功。

1.3.2　集成开发环境 Eclipse

Java 的集成开发环境有很多种，目前使用最广泛的是 Eclipse。Eclipse 可以从官方网站 www. eclipse. org 进行下载，与 JDK 的选择相同，下载 Eclipse 时也要与当前的操作系统对应（JDK 是 Eclipse 的运行环境，在安装 Eclipse 前必须要先安装好 JDK），下载好的 Eclipse 的安装包是一个压缩文件，解压缩之后，双击 eclipse. exe 文件启动 Eclipse。

（1）工作区

启动 Eclipse 时，系统将询问用户选择一个工作区（Workspace），如图 1-5 所示。工作区实际就是一个用来存储项目的文件夹，即启动 Eclipse 后建立的项目都将存储在此文件夹下。

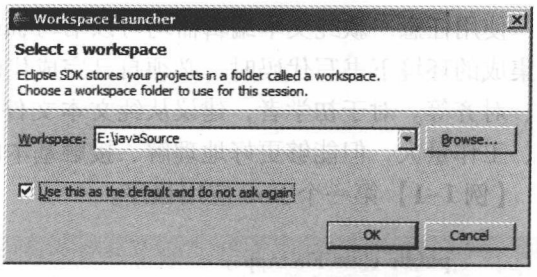


图 1-5　工作区的选择

（2）项目

Eclipse 用项目的方式管理文件，J2SE 方式下的 Java 应用程序对应的项目是“Java Project”。

在 Eclipse 窗口中，单击“File→New→Java Project”命令，打开新建项目向导，如图 1-6 所示，输入项目名称，单击“Finish”按钮完成项目的创建，项目将出现在“Package Explorer”（资源浏览器）窗口中，Eclipse 用树型结构展示项目下的所有资源，系统提供的“src”文件夹用于存储用户建立的所有 Java 程序，如图 1-7 所示。

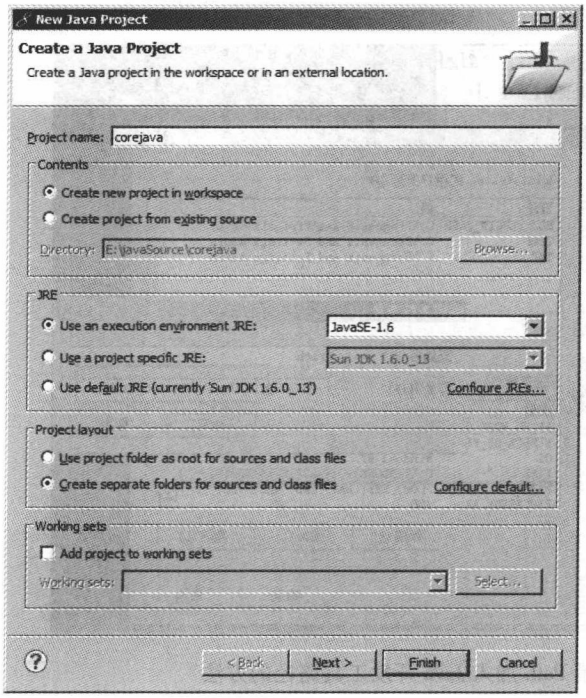


图 1-6　新建 Java Project

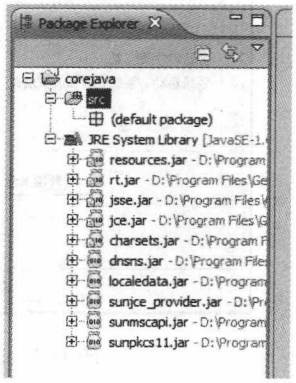


图 1-7　“Package Explorer”窗口

1.4　Java 应用程序

Java 应用程序是一个可以独立运行的程序，它只要有 Java 虚拟机就能运行。一个 Java 应用程序中一定要有一个类包含 main() 方法。

1.4.1　Java 应用程序的编写

使用任意一款纯文本编辑器均可以书写 Java 源程序（比如 Windows 下的记事本），只是在非集成的环境下书写代码时，必须自己完成代码格式的控制，包括缩进空格的输入、括号的匹配、对齐等。对于初学者，建议从纯文本文件的编写开始，不过早地使用 IDE 环境，虽然陌生、工作量大，但能够更好地理解、接触基本的概念，加深对 Java 的理解。

【例 1-1】第一个 Java 应用程序。

```
public class FirstApp {
    /**
```

```
        * 第一个 Java Application
        */
        public static void main(String[] args){
            //打印"Hello world!"
            System. out. println("Hello World!");
        }
}
```

本程序会在控制台打印一行信息：

```
Hello world!
```

Java 应用程序中包含如下要素：

1）类的声明。关键字 class 声明一个类，类名为 FirstApp。类体由{}括起来，用来封装类的属性和类的方法。

2）Java 应用程序的命名。一个 Java 应用程序文件可以由 n(n>0)个类组成，但这 n 个类中**只能有一个**类是 public 类（公共类）；并且应用程序的名字**必须**与公共类同名（Java 虚拟机要求公共类必须放在与其同名的源文件中），包括大小写。因此上面见到的应用程序只能有一个合法的命名 FirstApp. java（Java 源文件的扩展名为“java”）。

3）主类与 main()方法。main()方法是 Java 应用程序执行的入口。main()方法所在的类叫作主类，显然一个应用程序只能有一个主类。main()方法的签名（signature）：

```
public static void main(String[] args)
```

public 指明 main()是公共方法。

static 指明该方法是一个静态方法。

void 表示 main()方法没有返回值。

参数 String[] args 是 main()方法固定的参数，用一个 String 类型的数组接收运行应用程序时传递过来的参数。

4）控制台输出。在 Java 中向控制台输出文本的方式是使用 System. out 对象。

System. out. println()方法在控制台输出一行文本后回车换行。

System. out. print()方法在控制台输出一行文本后不回车换行。

除此之外，还可以使用 System. out. printf()方法进行格式化的控制输出，格式化方式与 C 语言相同。例如：

```
System. out. printf("% -20s\n","Hello world!");
```

5）注释。Java 中跨行代码段的注释使用“/*”“*/”，对于一行的注释使用“//”。

为了提高程序的可读性，还可以利用空行、空格进行区分。

1.4.2 命令行方式下的编译和运行

1. 编译

在命令行方式下，使用编译器 javac. exe 编译源文件。假设 FirstApp. java 文件存储在