



新世纪应用型高等教育
软件专业系列规划教材

Java 核心编程技术

新世纪应用型高等教育教材编审委员会 组编

主编 张屹 蔡木生

大连理工大学出版社



新世纪应用型高等教育
软件专业系列规划教材

新世纪

Java 核心编程技术

新世纪应用型高等教育教材编审委员会 组编

主编 张屹 蔡木生

副主编 聂常红 黄玲玲

吴向荣 邹立杰

林若钦

大连理工大学出版社

图书在版编目(CIP)数据

JAVA 核心编程技术 / 张屹, 蔡木生主编. — 大连 :
大连理工大学出版社, 2010.9
新世纪应用型高等教育软件专业系列规划教材
ISBN 978-7-5611-5767-1

I. ①J… II. ①张… ②蔡… 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(2010)第 170849 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:http://www.dutp.cn

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:21.25 字数:544 千字 光盘 1
印数:1~1500

2010 年 9 月第 1 版

2010 年 9 月第 1 次印刷

责任编辑:潘弘喆

责任校对:姜 伊

封面设计:张 莹

ISBN 978-7-5611-5767-1

定 价:45.00 元

前 言

1995年5月23日Java语言诞生,1996年1月JDK 1.0发布,屈指算来,Java尚属“少年”。由于Java语言具有面向对象、跨平台、分布式、安全、易用、开发周期短等诸多优点,顺应了Internet的发展,获得了巨大成功,迅速成为当今主流编程语言之一,在业界得到广泛使用,学习Java的人数与日俱增,许多高校已将Java语言列为必修课程。然而,由于Java包含的知识点较多,涉及领域较广,“教什么?怎么学?”是教师与学生需要面对的问题。

本书较好地回答上述两个问题:

(1)本书内容的选取突出了基础知识、经典内容,并与时俱进地增添了一些新技术、新特性。面向对象知识(如:类、对象、接口、继承与多态性、异常处理等)和Java经典内容(如:文件与输入输出流、数据库编程、多线程、网络编程等)是Java编程的核心技术,它们构成了本书的主体,对于图形用户界面(GUI)、Applet等传统内容进行了弱化处理。Java在其发展、应用过程中涌现了许多新知识、新技术,我们进行了筛选,增补了一些实用内容:如第3—6章中的UML图、第8章的Java泛型与Java集合、第9章的Word、Excel、PDF文件内容的读取等,目的是为了拓展读者视野;

(2)以初学者角度、根据人的认知规律,循序渐进地组织教学内容,并提供了一系列的教学资源进行“助学”。本书适合作为应用型本科教材,应用型软件人才首先应该掌握一定的理论知识,不是只会“照葫芦画瓢”的工匠;其次要注重实践能力的培养。为此,本书采用这样的编排顺序:先提出问题,再用平实的语言叙述语法及理论,之后用一些例子进行说明,并给出运行结果,以增强学生的感性认识。书中共有170多个例题,这些例题代码都经过上机调试,可以直接运行。为了让学生更好地把握教材内容,每一章的开始都有教学目标,正文之后是小结,涵盖了本章的主要知识点。第1章介绍了Java当今最流行的开发工具——Eclipse的安装与使用,与本书配套的还有一本实验指导教程,每章都有对应的实验内容,包含:实验目的、主要知识点、实验内容、实验总结四部分,这保证了实践环节的配套实施。之所以提供这些资源,是为了让学生快速掌握相关知识点;

(3)在知识学习过程中,注重能力的培养是本书的又一个着力点。“怎样培养学生的逻辑思维、编程能力”是教师、教材应该关注的问题,本书在这方面做了大量努力,例如:文中有意识地给出一些小问题让学生思考,以加深对知识的理解;鼓励学生查阅 API 文档可培养其自主学习能力;通过阅读程序代码、运行例题、来提高编程能力。

广州大学华软软件学院是一所应用型本科学院,以“培养、造就 IT 人才”为己任,Java 作为该校的主要技术方向之一,历经 10 年时间,形成了包含 Java SE、Java EE、Java ME 的多专业方向课程体系。本书是几位多年从事 Java 教学老师的辛劳之作,其中:第 1、12 章由张屹编写,第 2、4、9 章由蔡木生编写,第 3、5、6 章由聂常红编写,第 7 章由林若钦编写,第 8 章由邹立杰编写,第 10、11 章由吴向荣编写,第 13、14 章由黄玲玲编写。

由于编写时间仓促,编者水平有限,书中错误之处在所难免,恳请读者谅解、指正。

所有意见和建议请发往:gzjckfb@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411-84707492 84706104

编 者

2010 年 9 月



第 1 章 Java 概述	1
1.1 Java 语言的诞生与发展	1
1.2 Java 语言的特点	1
1.3 Java 程序运行环境的配置与使用	3
1.4 Java 程序的类型	13
第 2 章 Java 编程基础	17
2.1 Java 的数据类型	17
2.2 Java 的关键字、标识符、常量和变量	20
2.3 Java 的运算符、表达式及语句	23
2.4 Java 的流程控制	28
2.5 Java 程序的基本结构及常用的输入输出格式	37
2.6 数组	40
第 3 章 类与对象	47
3.1 OOP 设计概述	47
3.2 类	48
3.3 对象	61
3.4 对象数组	71
3.5 包	72
第 4 章 常用类	77
4.1 String 类	77
4.2 StringBuffer 类	83
4.3 Math 类	86
4.4 包装类	89
4.5 日期日历类	91
第 5 章 继承和多态性	97
5.1 继承的概念	97
5.2 子类的创建	97
5.3 访问修饰符和继承性	98
5.4 is-a 和 has-a 之间的联系	104
5.5 成员变量的隐藏和方法重写	106
5.6 super 关键字	107
5.7 继承的层次性	111
5.8 final 关键字	114
5.9 多态性	115
5.10 Object 类	119
第 6 章 抽象类与接口	124
6.1 抽象类	124
6.2 接口	129
第 7 章 异常处理	138
7.1 异常与异常类型	138
7.2 异常处理	140

7.3	重新抛出异常	145
7.4	finally 子句	146
7.5	自定义异常	146
第 8 章	Java 泛型与 Java 集合	150
8.1	Java 泛型	150
8.2	Java 集合	155
8.3	Java 泛型和 Java 集合综合实例	160
第 9 章	文件与输入输出流	162
9.1	File 类与文件操作	162
9.2	输入输出流	167
9.3	字节流	170
9.4	字符流	181
9.5	对象序列化	190
9.6	随机存取文件	192
9.7	Word、Excel、PDF 文件的操作(选学)	194
第 10 章	图形用户界面设计	206
10.1	图形用户界面(GUI)	206
10.2	AWT 容器类	208
10.3	AWT 独立组件类	214
10.4	AWT 菜单类	224
10.5	Swing 组件	228
10.6	布局管理器	234
10.7	字体和颜色的使用	241
10.8	图形绘制和图像显示	243
第 11 章	事件处理	248
11.1	事件委托模型	248
11.2	事件类和事件对象	251
11.3	事件侦听器与侦听接口	253
11.4	事件处理	256
第 12 章	数据库编程	267
12.1	JDBC 设计	267
12.2	安装 JDBC	269
12.3	JDBC 编程的基本概念	271
12.4	执行查询操作	276
12.5	滚动和更新结果集	277
12.6	事务	280
第 13 章	多线程	284
13.1	线程的概念	284
13.2	创建线程的两种方法	287
13.3	线程的生命周期	290
13.4	线程同步	295
13.5	多线程的应用	299
第 14 章	网络编程	301
14.1	网络编程的基本概念	301
14.2	TCP 编程	307
14.3	UDP 编程	322
参考文献	333

从本章起,我们将开始 Java 语言的学习,先介绍 Java 语言的诞生与发展,接着说明 Java 语言的特点,然后讲述 Java 的开发环境的安装、配置、使用,最后讨论 Java 程序的分类和基本结构。

● 学习目标

- 了解 Java 语言的诞生与发展;
- 了解 Java 语言的特点;
- 掌握 Java 程序运行环境的配置与使用;
- 熟悉 Java 程序的类型及基本结构。

1.1 Java 语言的诞生与发展

1996 年 Java 1.0 第一次发布就引起人们的极大兴趣,它是一种应用于分布式网络环境中的程序设计语言,由 Sun 公司开发。Java 语言广泛流行得益于 Internet 的迅速发展。特别是 Applet 在 Web 上的应用吸引了更多人们对 Java 语言的关注,人们开始使用这种语言。Java 语言从它诞生起就显现出强大的威力与优越性。一般的应用程序、Applet、Web 服务器乃至嵌入式系统,Java 都足以胜任,并且表现十分出色。特别是 Java 对网络提供了强有力的支持。正是因为它集多种优势于一身,所以对广大的程序设计人员来说有着不可抗拒的吸引力。

在推出 JDK1.0 后,Sun 公司在 1997 年初发布了 JDK 1.1。其相对于 JDK 1.0 最大的改进就是为 JVM 增加 JIT(即时编译)编译器。JIT 会将经常用到的指令保存在内存中,在下次调用时就不需要再编译,这样 JDK 在效率上就会得到提升。1998 年 12 月 Sun 公司发布新的版本 JDK 1.2,标志着 Java 进入 Java2 时代。在这一时期 Sun 公司发布了 JSP/Servlet、EJB 规范,将 Java 分成了 J2SE、J2EE 和 J2ME,标志着 Java 向企业、桌面和移动 3 个领域进军。2000 年 5 月 Sun 公司对 JDK 1.2 进行升级,推出 JDK 1.3,增加 DNS 及 JNI 的支持,使得 Web 容器得到广泛的应用。Sun 在 2002 年 2 月发布 JDK 最为成熟的版本:JDK 1.4,性能上获得极大的提高,已经可以使用 Java 实现大多数应用了。但是它又面临一些新问题,比如它不支持泛型、增强的 for 语句和 Java 相关的技术 EJB2. x,由于比较复杂很少有人使用。2004 年 10 月 Sun 公司发布了 JDK 1.5 后改名为 J2SE 5.0,也就是将版本号 1.5 改为 5.0。其中增加了泛型、增强 for 语句、注解、自动拆箱和装箱等功能,同时更新了企业级规范,改善了 EJB 的复杂性,推出了 EJB 3.0 规范。2007 年 Sun 推出 J2SE 6.0 正式版,在性能、易用性方面得到极大提高,在脚本和 API 上获得了全新的支持。本书会详细介绍 JDK 1.6,即 J2SE 6.0。

1.2 Java 语言的特点

Java 是一种被广泛使用的网络编程语言。它定位于网络计算,几乎所有特点都是服从于这一点。同时 Java 语言也集中体现并充分利用了许多软件技术的新成果。

1.2.1 简单性

Java 语法与 C++ 的语法相比较为简单。它没有头文件、指针运算、结构、联合、操作符重载、虚基类等。然而,Java 还保留 C++ 的一些特性,如:switch 语句的语法在 Java 中没有改变。如果熟悉 C++ 就会很快将它转换成 Java。另一方面 Java 的开发包是很小的,基本的解释器以及类支持大约 40 KB,再加上基础的标准类库和对线程的支持大约需要增加 175 KB。

1.2.2 面向对象

Java 面向对象特性与 C++ 旗鼓相当,但主要不同点在于多继承。在 Java 中,取而代之的是简单的接口概念,以及 Java 的元类模型、反射机制。对象序列化特性使得 Java 更容易拥有持久对象并进行 GUI 构建。

1.2.3 可移植性

在 Java 中数据类型具有固定的大小,这是消除代码移植时的主要问题。二进制数据以固定的格式存储和传输,消除了字节顺序的困扰。字符串是用标准的 Unicode 格式存储的。Java 语言的可移植性使得编写的应用程序可在任何平台上良好运行,体现了 Java 语言“一次编译,处处运行”的优点。

1.2.4 安全性

Java 语言对网络提供了强有力的支持,这就不得不考虑网络安全的问题。网络上的应用程序必须具有较高的安全性和可靠性,Java 特有的 Sandbox 机制是其安全性的保障;同时它删除了 C++ 语言中复杂而且容易造成错误的指针,保证了 Java 程序运行的安全可靠。Java 小程序 Applet 在浏览器中运行时,语言功能受限于浏览器本身,这使得 Applet 对用户是安全的。同时,许多安全特性相继地加入 Java 中,比如 Java 中的数字签名类,通过它可以确定类的作者。如果信任该类的作者,这个类就可以在您的机器上拥有更多的权限。

1.2.5 虚拟机 JVM

JVM 是 Java Virtual Machine 的缩写,即 Java 虚拟机。事实上它并非一种机器,而是一种运行 Java 程序的软件实现,是虚拟的机器。通常 Windows 应用程序只能在 Windows 平台上运行,Linux 应用程序只能在 Linux 平台上运行。前面说过,Java 程序与平台无关,它直接在 JVM 中运行。见图 1-1 所示:

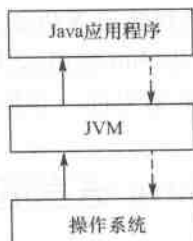


图 1-1 描述 JVM 充当 Java 程序与系统平台的桥梁

运行 Java 程序,必须首先安装 JVM。事实上,每个平台需要安装不同版本的 JVM,比如 OS/2、Windows、Linux、Unix 等不同版本的 JVM。

Java 语言使用 JVM 屏蔽了与具体平台相关的信息,使得 Java 语言编译程序只需生成能够在 JVM 上运行的目标代码即字节码,就可以在不同平台上不加修改地运行。JVM 在执行字节码时,将其解释成具体平台上的机器指令执行。图 1-2 为 Java 虚拟机的运行过程。

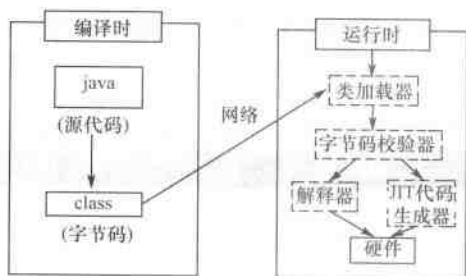


图 1-2 Java 虚拟机的运行过程

字节码的执行需要经过以下过程:首先由类加载器负责把类文件加载到 JVM 中,在此过程中需要检验该类文件是否符合类文件规范。其次字节码校验器检查该类文件的代码中是否存在某些非法操作。如果校验通过,由 Java 解释器负责把类文件解释成为机器码进行执行。JVM 采用的是“沙箱”运行模式,把 Java 程序的代码和数据都限制在一定内存空间里执行。

1.2.6 多线程

多线程可以带来更好的交互响应和实时行为。我们可以把线程看做是进程中的小进程。可以在一个程序中同时运行多个不同的小程序,像是多个程序同时在一个程序中运行一样。Java 把多线程的实现交给了底层的操作系统或线程库来完成。因此多线程是 Java 成为有魅力的服务器端开发语言的主要原因之一。

1.2.7 动态性

当需要将某些代码添加到正在运行的程序中时,动态性将是非常重要的。目前 Java 的版本允许程序员了解对象的结构和行为,这对于必须在运行时分析对象的系统来说非常有用。这些系统有:Java GUI 构建器、智能调试器、可插入组件以及对象数据库。

1.2.8 垃圾回收机制

Java 语言提供了垃圾回收机制,用来自动回收内存垃圾。这使得程序设计人员在编写 Java 程序时不必特别考虑内存管理的问题。在程序设计中,会出现内存垃圾。自动垃圾回收功能将其回收,并释放相应的内存。

1.3 Java 程序运行环境的配置与使用

要运行 Java 程序,只需下载一个 JDK 开发工具包就可以。JDK 开发工具包中包含完整的 JRE(Java Runtime Environment,Java 运行环境)、各种类库和示例程序。JDK 是 Java 程序

员最初使用的开发环境,由一个标准类库和一组可用于建立、测试 Java 程序及创建文档的实用工具组成,其核心是 Java API,它包含一些重要的语言结构以及基本图形、网络和文件 I/O。

1.3.1 安装 Java 开发环境

1. 下载 JDK

访问下面的网址 <http://java.sun.com/javase/downloads/index.jsp> 下载最新版本的 JDK。如图 1-3 所示。

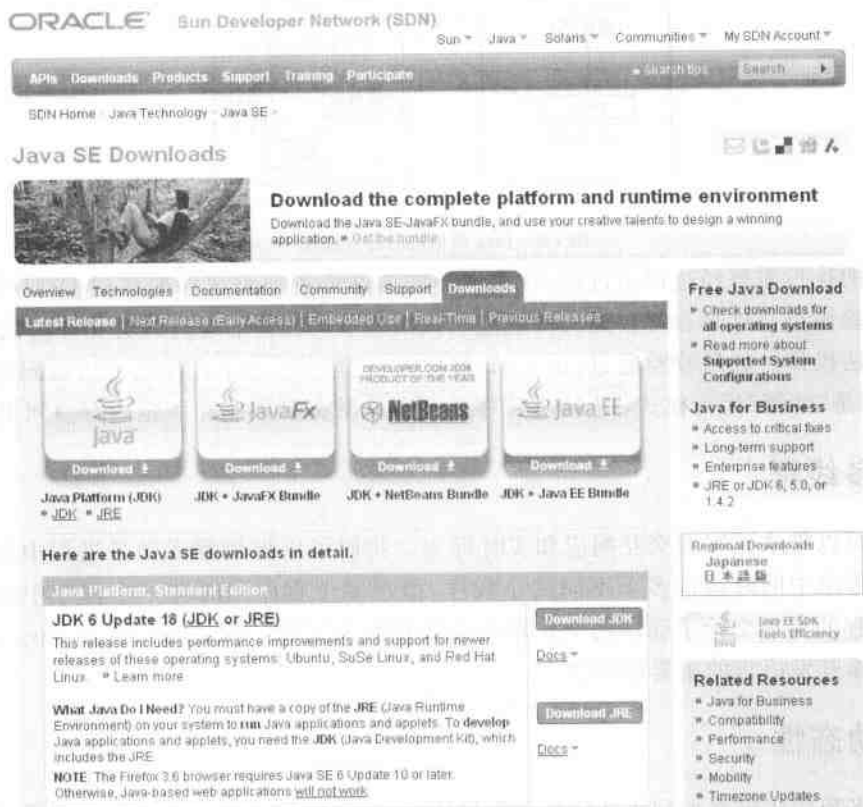


图 1-3 下载 JDK 选择页面

单击 Download JDK 按钮后进入下载选择页面,如图 1-4 所示。

在下载窗口中,默认选择下载的操作系统是 Windows,直接点击 Download 就可以下载。这里下载的是 JDK 6 UPDATE 18。下载完毕后可以发现一个名为 `jdk-6u18-windows-i586-p.exe` 文件。

2. 安装 JDK

找到下载的 JDK 文件 `jdk-6u18-windows-i586-p.exe`,就可以开始安装了。具体步骤如下:

(1) 双击打开 `jdk-6u18-windows-i586-p.exe` 文件,打开【许可协议】窗口,单击【接受】按钮,打开【自定义安装】窗口。

(2) 在这个窗口中可以根据自己的需要,更改文件的安装路径以及选择是否安装某些组件。这里演示把 JDK 安装到 `C:\Program Files\Java\jdk1.6.0_18` 目录下,并安装所有的组件,如图 1-5 所示。

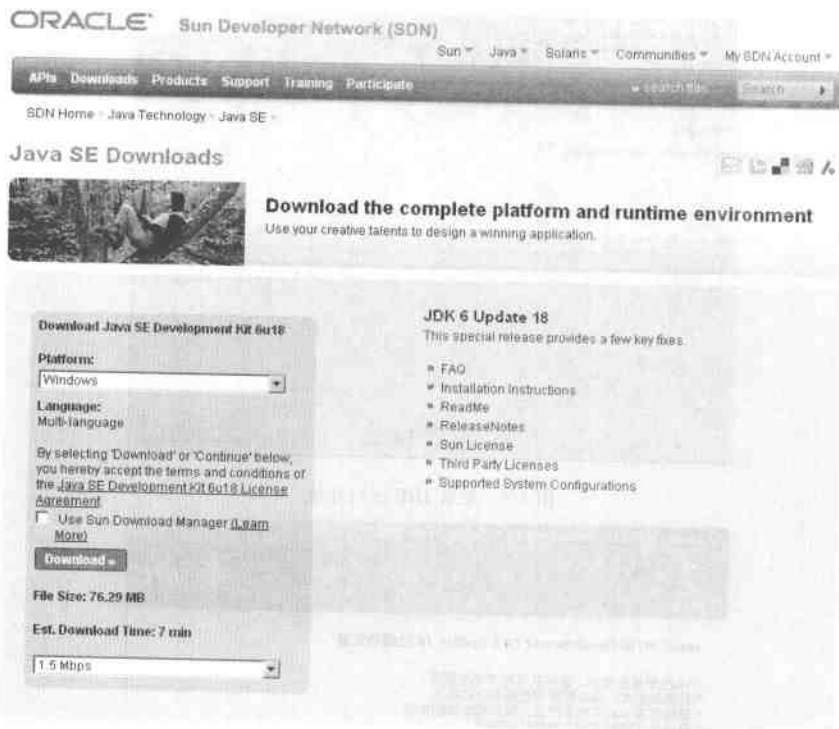


图 1-4 下载适合操作系统的 JDK 的页面

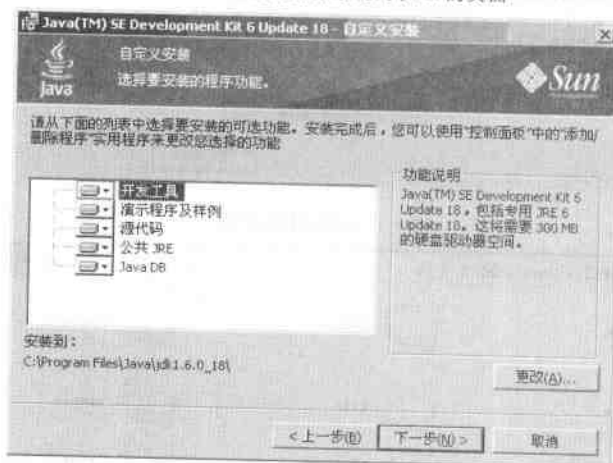


图 1-5 自定义安装对话框

(3) 设置完成后,单击【下一步】按钮开始进行安装。

(4) JDK 类库安装完成后,会提示安装 JRE 运行环境。用户可以根据自己的需要选择安装的路径。单击【下一步】按钮,开始安装 JRE,设置安装目录为 C:\Program Files\Java\jre1.6.0_18,如图 1-6 所示。

(5) 安装成功后显示对话框如图 1-7 所示。点击【完成】按钮,结束安装过程。

3. 了解 Java 安装目录

在学习 Java 的过程中,经常需要查看 Java 源文件,当然也会频繁地使用类库文档。JDK 安装成功后,打开安装目录,如图 1-8 所示。

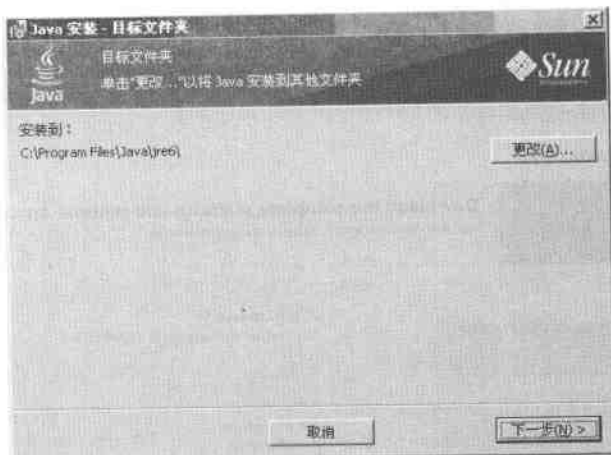


图 1-6 安装 JRE 运行环境

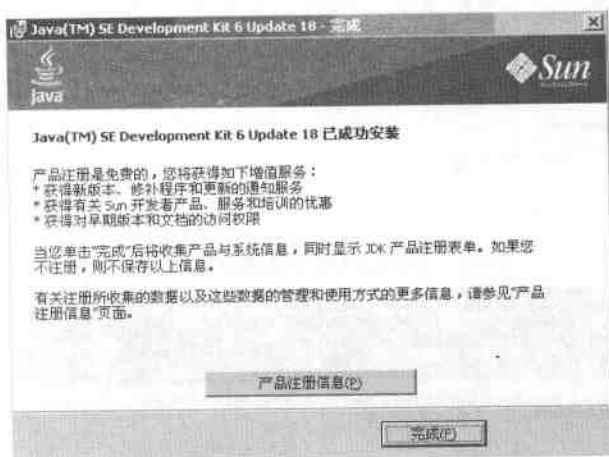


图 1-7 安装 JDK 成功

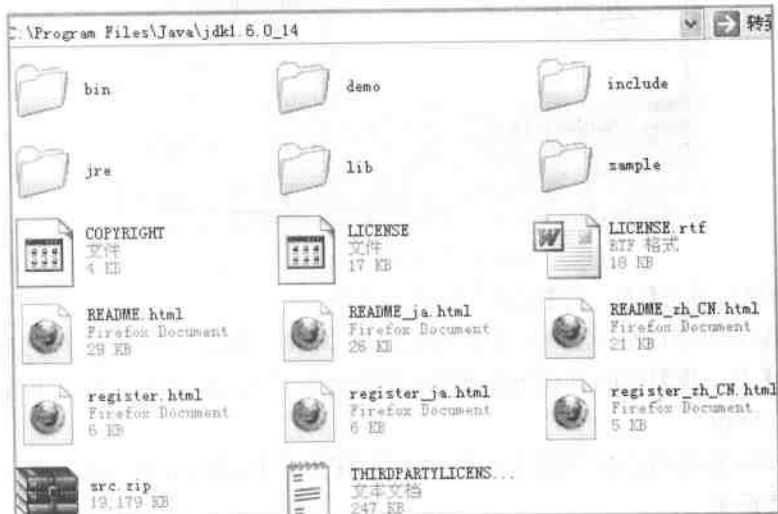


图 1-8 JDK 安装目录

从上图可知, JDK 安装目录具有以下多个目录和一些网页文件:

- (1) bin 目录: 提供 JDK 工具程序, 包括 javac、java、javadoc、appletviewer 等可执行程序。
- (2) demo 目录: Sun 公司为 Java 使用者提供的一些已经编写好的范例程序。
- (3) jre 目录: 存放 Java 运行环境文件。
- (4) lib 目录: 存放 Java 的类库文件, 即工具程序使用的 Java 类库。JDK 中的工具程序大多由 Java 编写而成。
- (5) include 目录: 存放用于本地方法的文件。
- (6) src.zip 文件: Java 提供的 API 类的源代码压缩文件。如果需要知道 API 的某些类如何实现, 可以查看这个文件中源代码的内容。

1.3.2 配置 Java 环境变量

在安装好 JDK 之后, 需要进行一些配置才能继续后面的应用程序开发。具体配置步骤如下:

(1) 在 Windows 桌面上, 右击【我的电脑】图标, 从弹出的菜单中选择【属性】命令, 弹出【系统属性】对话框。

(2) 在【系统属性】对话框中, 选择【高级】选项卡, 单击【环境变量】按钮, 弹出【环境变量】对话框。

(3) 在【环境变量】对话框的【系统变量】选项区域中, 选中变量 path, 单击【编辑】按钮, 在弹出的【编辑系统变量】对话框中, 加入“C:\Program Files\Java\jdk1.6.0_18\bin;”(即 JDK bin 目录所在路径, 注意路径后需要加“;”), 如图 1-9 所示。

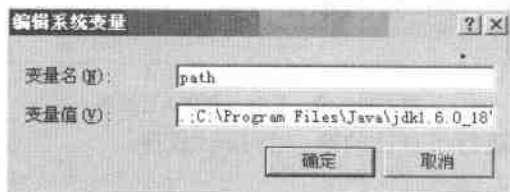


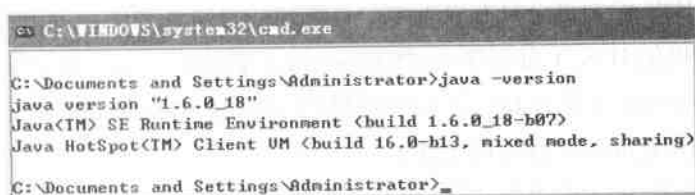
图 1-9 配置 path 变量

(4) 按照同样的方式编辑系统变量 classpath, 变量值为:

```
.;C:\ProgramFiles\jdk1.6.0_18\bin\da.jar;C:\Program  
Files\Java\jdk1.6.0_18\bin\tools.jar;
```

(5) 这样就完成了 JDK 在 Windows XP/2000/2003 操作系统上的安装与配置。path 变量必须要进行配置。classpath 环境变量一般情况不需要设置, 只有在计算机上安装了其他的 Java 开发工具时, 才需要配置。

为了检查 JDK 是否配置成功, 可以打开命令提示符窗口, 输入“java -version”命令。如果成功配置, 会出现当前 JDK 的版本号, 如图 1-10 所示。



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Administrator>java -version
java version "1.6.0_18"
Java(TM) SE Runtime Environment (build 1.6.0_18-b07)
Java HotSpot(TM) Client VM (build 16.0-b13, mixed mode, sharing)

C:\Documents and Settings\Administrator>_
```

图 1-10 测试 JDK 是否成功

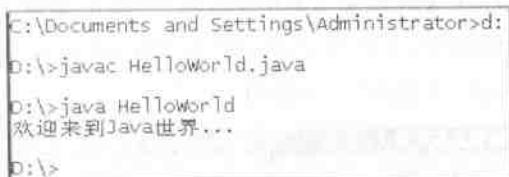
1.3.3 使用命令行工具

在使用命令行工具之前,读者必须先配置好环境变量。下面向读者演示如何编写第一个 Java 应用程序。

打开记事本,在“D:\目录”下创建“HelloWorld.java”文件,其代码如下:

```
public class HelloWorld {
    public static void main(String [] args) {
        System.out.println("欢迎来到 Java 世界...");
    }
}
```

首先选择【开始】→【运行】命令,输入 cmd 命令,切换路径至“D:\”;然后输入“javac HelloWorld.java”,编译程序,编译成功后无结果输出;接着输入“java HelloWorld”,运行程序,输出结果如图 1-11 所示。



```
C:\Documents and Settings\Administrator>d:
D:\>javac HelloWorld.java
D:\>java HelloWorld
欢迎来到Java世界...
D:\>
```

图 1-11 第一个 Java 程序运行结果

祝贺你!已经编译并运行了自己的第一个 Java 程序。

1.3.4 使用集成开发环境

Java 作为一门流行的网络语言,相应的图形化工具有很多,比较著名的有 IBM 公司的 Eclipse、Sun 公司的 NetBeans 和 Borland 公司的 JBuilder 等。IBM 的 Eclipse 以其开源和可扩展的优点深受广大程序员的喜爱。

本节将介绍 Eclipse 的安装与使用。

1. Eclipse 概述

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言,它只是一个框架和一组服务,用于通过插件组件构建开发环境。它专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台,主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目 3 个项目组成。

2. Eclipse 获取与安装

Eclipse 是一个开放源代码的项目,可以到其官方网站 www.eclipse.org 上免费下载

Eclipse 的最新版本。本书所使用的 Eclipse 为 Windows 平台下的 Eclipse 3.5 版本。

安装 Eclipse 的步骤非常简单,只需将下载的压缩包按原路径直接解压即可。如果有新的版本,需要先删除旧的版本,再重新安装,不能直接解压到原有的路径覆盖旧版本。解压后,可以到相应的安装路径下找到“Eclipse.exe”文件双击运行。启动 Eclipse,会出现如图 1-12 所示。

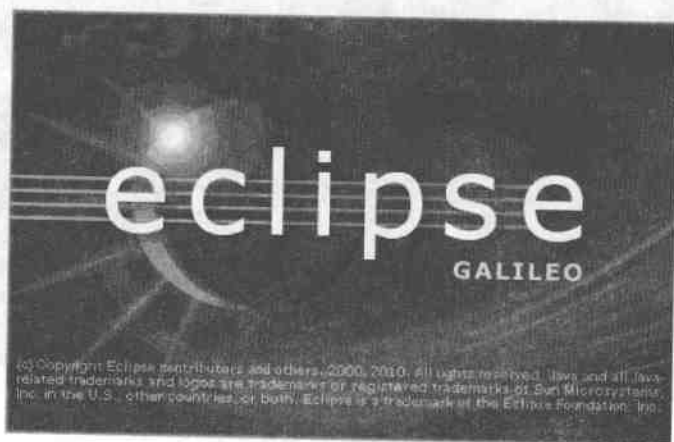


图 1-12 Eclipse 启动

随后出现一个如图 1-13 所示的选择工作区路径对话框。Eclipse 会将所有文件存放在工作区指定的路径下。

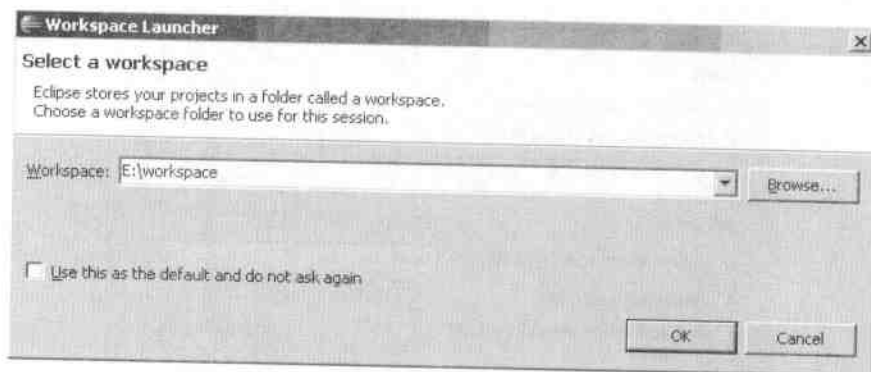


图 1-13 Eclipse 选择项目工作区

确定工作区路径后单击【OK】按钮,打开如图 1-14 所示的欢迎界面。如果未安装 JDK 就启动 Eclipse,系统会报告相应的错误信息,提示用户先安装 JDK,正确配置后再重新启动。

在 Eclipse 界面的菜单栏上,选择【Help】→【Software updates】→【Find and Install...】选项。读者可以按要求执行相应的操作来实现 Eclipse 的更新。

3. Eclipse 开发 Java 程序

使用 Eclipse 开发 Java 程序非常简单。首先创建相应的 Java 项目,然后创建 Java 源文件即可。下面演示创建一个 Java 项目,在里面创建一个类,输出“使用 Eclipse 开发 Java 程序”字符串。

选择【File】→【New】→【Java Project】命令,弹出一个如图 1-15 所示的窗口。

在【Project name】文本框中输入项目名称“HelloWorld”。窗口下面的信息主要是用来显

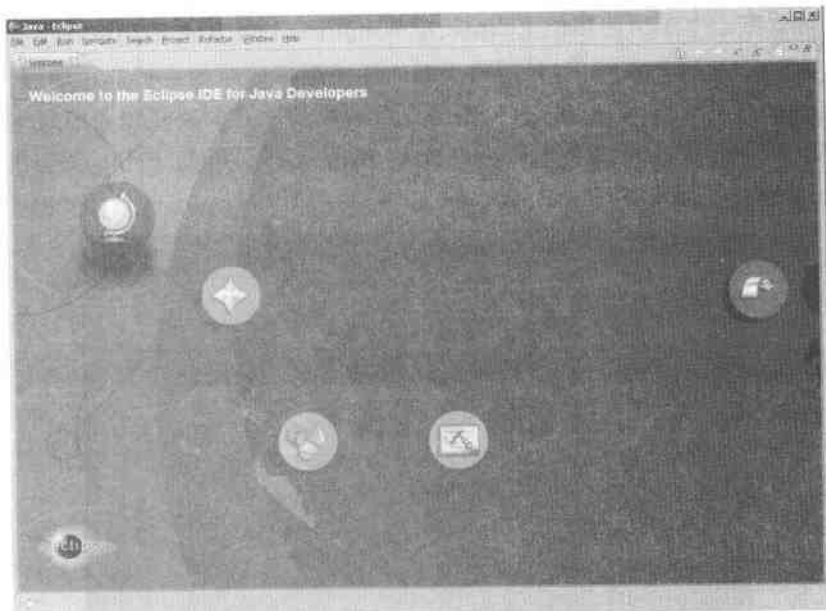


图 1-14 Eclipse 欢迎界面

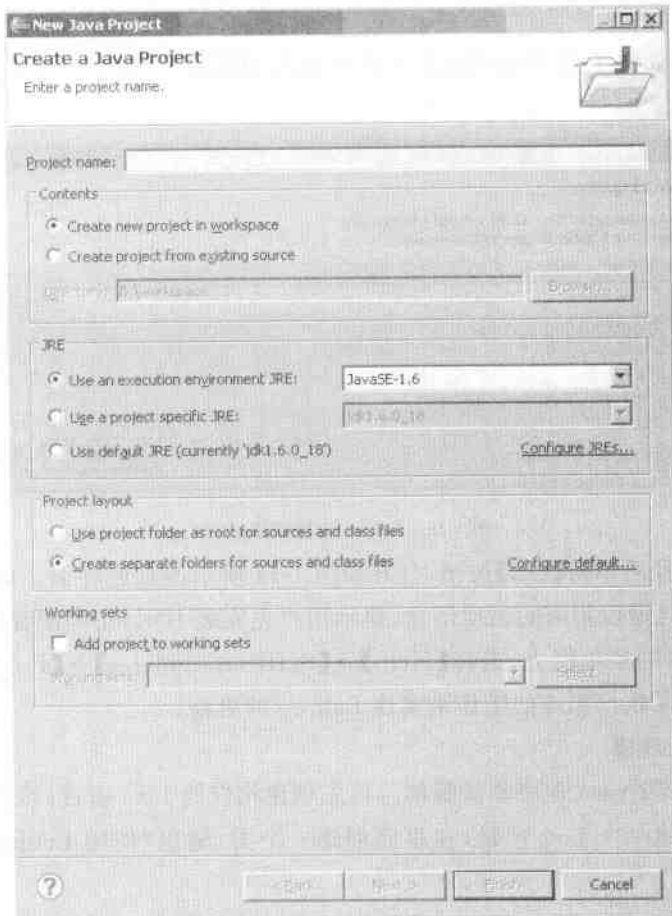


图 1-15 新建 Java 项目