



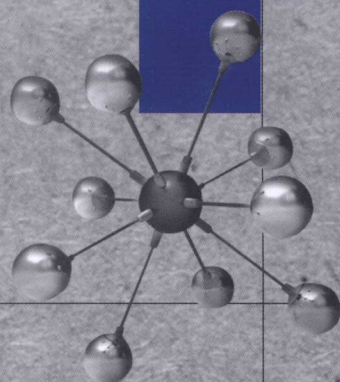
新世纪应用型高等教育软件专业系列规划教材

(第二版)

Java核心编程技术

JAVA HEXIN BIANCHENG JISHU

主 编 张 屹 蔡木生



大连理工大学出版社



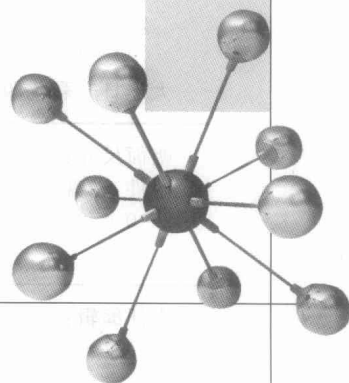
新世纪应用型高等教育软件专业系列规划教材

(第二版)

Java核心编程技术

JAVA HEXIN BIANCHENG JISHU

主 编 张 屹 蔡木生
副主编 谭翔纬 聂常红 潘正军
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. — 2 版
— 大连: 大连理工大学出版社, 2015.7
新世纪应用型高等教育软件专业系列规划教材
ISBN 978-7-5611-9880-3

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(2015)第 115417 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:0411-84708943 传真:0411-84701466

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

大连日升彩色印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm	印张:23	字数:589 千字
附件:光盘 1 张		印数:1~2000
2010 年 9 月第 1 版		2015 年 7 月第 2 版
	2015 年 7 月第 1 次印刷	

责任编辑:王晓历

责任校对:姜 伊

封面设计:张 莹

ISBN 978-7-5611-9880-3

定 价:49.80 元

《Java 核心编程技术》(第二版)是新世纪应用型高等教育软件专业系列规划教材之一。

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,所以获得了巨大成功,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。

本教材较好地回答了上述两个问题:

(1)本教材内容的选取突出了基础知识和经典内容,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、继承和多态性、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、多线程与网络编程等)是 Java 编程的核心技术,它们构成了本教材的主体,本教材还对图形用户界面(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,本教材进行了筛选,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的视野。

(2)本教材以初学者的角度、根据人的认知规律,循序渐进地组织教学内容,并提供一系列的教学资源进行“助学”。本教材适合作为应用型本科计算机专业、软件专业教材。应用型软件人才首先应该掌握一定的理论知识,不能做只会“照葫芦画瓢”的工匠;其次要注重实践能力的培养。为此,本教材采用这样的编排顺序:先提出问题,再用平实的语言叙述理论,之后用一些例子进行说明,并给出运行结果,以增强学生的认识。教材中共有 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,每章的开始都有学习目标,每章的结尾处都有本章小结,涵盖了本章的主要知识点。与本教材配套的是《Java 核心编程技术实验指导教程》(第二版),



每章都有对应的实验内容,包括:实验目的、相关知识点、实验内容与步骤、实验总结四部分,有效地保证了实践环节的配套实施,使学生快速掌握相关知识点。

(3)在知识学习的同时,注重能力培养是本教材的又一个着力点。“怎样培养学生的逻辑思维能力、编程能力”是教师、教材应该关注的问题,本教材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,以加深学生对知识的理解;鼓励学生查阅 API 文档可培养其自主学习的能力;通过阅读程序代码、运行例题来提高编程能力。

本教材由张屹、蔡木生任主编,负责拟订全书大纲,并进行最终的统稿;谭翔纬、聂常红、潘正军、吴向荣、邹立杰、林若钦任副主编。具体编写分工如下:张屹编写第 1 章、第 12 章,蔡木生编写第 2 章、第 4 章和第 9 章,聂常红编写第 3 章、第 5 章和第 6 章,潘正军编写第 13 章、第 14 章,吴向荣编写第 10 章、第 11 章,邹立杰编写第 8 章,林若钦编写第 7 章,谭翔纬协助统稿。

本教材在编写过程中,参考、借鉴了许多专家、学者的相关著作,对于引用的段落、文字尽可能一一列出,谨向各位专家、学者一并表示感谢。

限于水平,书中也许仍有疏漏和不妥之处,敬请专家和读者批评指正,以使教材日臻完善。

编 者

2015 年 7 月

所有意见和建议请发往:dutpbk@163.com

欢迎访问教材服务网站:<http://www.dutpbook.com>

联系电话:0411-84708445 84708462



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	2
1.3 Java 程序运行环境的配置与使用	4
1.4 Java 应用程序的结构	13
1.5 API 文档的下载、安装与使用	14
第 2 章 Java 编程基础	16
2.1 Java 的数据类型	16
2.2 Java 的关键字、标识符、常量和变量	19
2.3 Java 的运算符、表达式及语句	22
2.4 Java 的流程控制	27
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	48
3.1 OOP 设计概述	48
3.2 类	49
3.3 对 象	61
3.4 静态变量与静态方法	71
3.5 包	75
第 4 章 Java API 实用类	80
4.1 String 类	80
4.2 StringBuffer 类和 StringBuilder 类	86
4.3 Math 类	92
4.4 包装类	95
4.5 日期日历类	97
第 5 章 继承和多态性	104
5.1 继承的概念	104
5.2 子类的创建	104
5.3 访问修饰符和继承性	105
5.4 is-a 和 has-a 之间的联系	111
5.5 成员变量的隐藏和方法重写	112
5.6 super 关键字	114
5.7 继承的层次性	118
5.8 final 关键字	121
5.9 多态性	122
5.10 Object 类	126
第 6 章 抽象类与接口	132
6.1 抽象类	132
6.2 接 口	137
6.3 抽象类与接口的区别	144
6.4 自动注解	148
第 7 章 异常处理	157
7.1 异常与异常类型	157
7.2 异常处理	159

7.3	重新抛出异常	164
7.4	finally 子句	165
7.5	自定义异常	165
第 8 章	Java 泛型与 Java 集合	169
8.1	Java 泛型	169
8.2	Java 集合	174
8.3	Java 泛型与 Java 集合综合实例	179
第 9 章	文件与输入输出流	181
9.1	File 类与文件操作	181
9.2	输入输出流	186
9.3	字节流	188
9.4	字符流	200
9.5	对象序列化和反序列化	209
9.6	随机存取文件	211
9.7	Word、Excel、PDF 文件的操作(选学)	213
第 10 章	图形用户界面设计	225
10.1	图形用户界面(GUI)	225
10.2	AWT 容器类	227
10.3	AWT 独立组件类	233
10.4	AWT 菜单类	243
10.5	Swing 组件	247
10.6	布局管理器	252
10.7	字体和颜色的使用	259
10.8	图形绘制和图像显示	261
第 11 章	事件处理	266
11.1	事件委托模型	266
11.2	事件类和事件对象	269
11.3	事件侦听器与侦听接口	271
11.4	事件处理	274
第 12 章	数据库编程	284
12.1	JDBC 设计	284
12.2	安装 JDBC	286
12.3	JDBC 编程的基本概念	288
12.4	执行查询操作	293
12.5	滚动和更新结果集	294
12.6	事务及存储过程的调用	297
第 13 章	多线程与网络编程	303
13.1	线程的概念	303
13.2	创建线程的方式	306
13.3	线程的生命周期	309
13.4	线程同步	314
13.5	多线程的应用	318
13.6	网络编程的基本概念	318
13.7	TCP 编程	324
13.8	UDP 编程	340
第 14 章	JUnit	351
14.1	JUnit 简介及安装	351
14.2	编写 JUnit 测试代码	352
14.3	JUnit 的套件(Suite)	357
14.4	参数化测试	359
参考文献		362

第 1 章

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 开发环境的安装、配置、使用,最后讨论 Java 程序的基本结构与 JDK API。

● 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 程序的基本结构;
- 熟悉 Java 开发的 API 文档。

1.1 Java 语言的诞生与发展

1996 年 Java 1.0 第一次发布就引起了人们的极大兴趣,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对 Java 语言的关注,人们开始使用这种语言。Java 语言从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,并且表现得十分出色。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 JDK 1.0 后,Sun 公司在 1997 年初发布了 JDK 1.1。其相对于 JDK 1.0,JDK 1.1 最大的改进就是为 JVM 增加了 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、桌面和移动三个领域进军。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。但是它又面临着一些新问题,比如它不支持泛型、增强的 for 语句和 Java 相关的技术 EJB 2.x,而且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、易用性方面得到了极大提高,在脚本和 API 上获得了全新的支持。2009 年 4 月 7 日,Google App Engine 开始支持 Java,同年,甲骨文(Oracle)公司以 74 亿美元收购 Sun,取得 Java 的版权。2011 年,代号为 Dolphin(海豚)的 Java 7.0 正式版发布,对核心 API 进行了改进。2014 年 3 月 19 日,甲骨文公司发布代号为 Spider(蜘蛛)的 Java 8.0 正式版,这个版本使 Java 更易于为多核处理器编写高效的代码,同时为与 Javascript 进行交互提供了更好的支持。

本书案例使用的 JDK 为 1.8 版本,即 J2SE 8.0 版。

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,几乎所有的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++ 的语法相比较更为简单。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 还保留了 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++ 就能很快将它转换成 Java。另一方面,Java 的开发包很小,基本的解释器以及类支持仅 40 KB,再加上基础的标准类库和对线程的支持也只大约需要增加 175 KB。

1.2.2 面向对象

Java 面向对象的特性与 C++ 旗鼓相当,但两者主要的不同点在于多继承。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,到处运行”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,这个类就可以在用户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,如图 1-1 所示。

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。

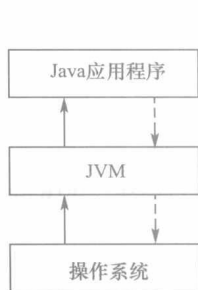


图 1-1 JVM 充当 Java 程序与系统平台的桥梁

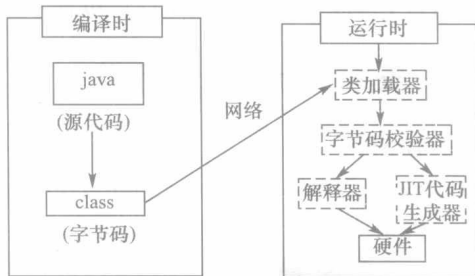


图 1-2 Java 虚拟机的运行过程

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。我们可以把线程看作是进程中的小进程。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这些垃圾回收,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

访问网址 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载最新版本的 JDK。如图 1-3 所示。

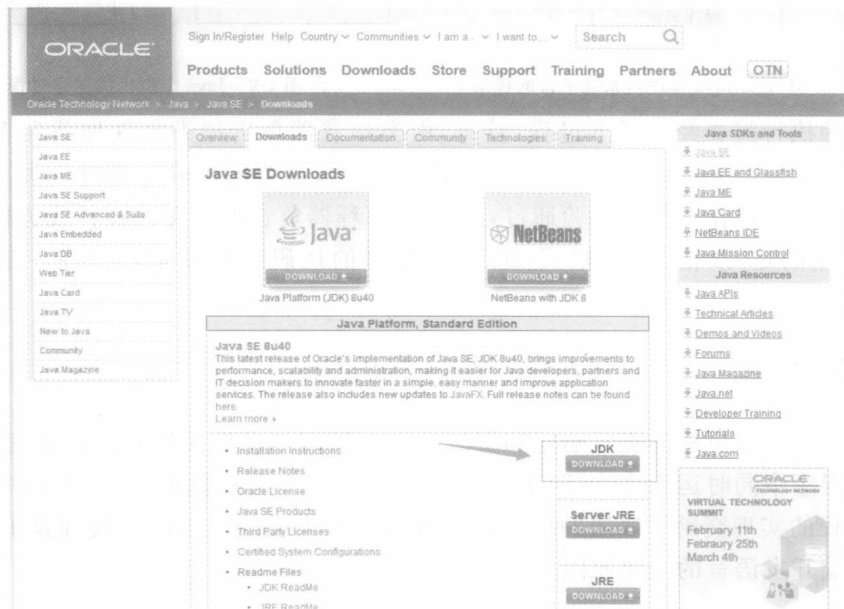


图 1-3 下载 JDK 选择页面

单击 JDK DOWNLOAD 按钮后进入选择页面,如图 1-4 所示。

在下载窗口中,选择对应的操作系统版本,直接单击相应版本的 Download 链接就可以下载。这里下载的是 Windows 64 位版本的 JDK 安装文件。下载完毕后可以看到一个名为 jdk-8u40-windows-i586.exe 文件。

2. 安装 JDK

在本地磁盘找到下载的安装文件,具体安装步骤如下:

(1) 双击打开 jdk-8u40-windows-i586.exe 文件,打开“许可协议”对话框,单击“接受”按钮,打开“定制安装”对话框。

(2) 根据自己的需要,更改安装路径和组件,此处演示将 JDK 安装到 D:\Java\jdk1.8.0_05\目录下,并安装所有的组件,如图 1-5 所示。

(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

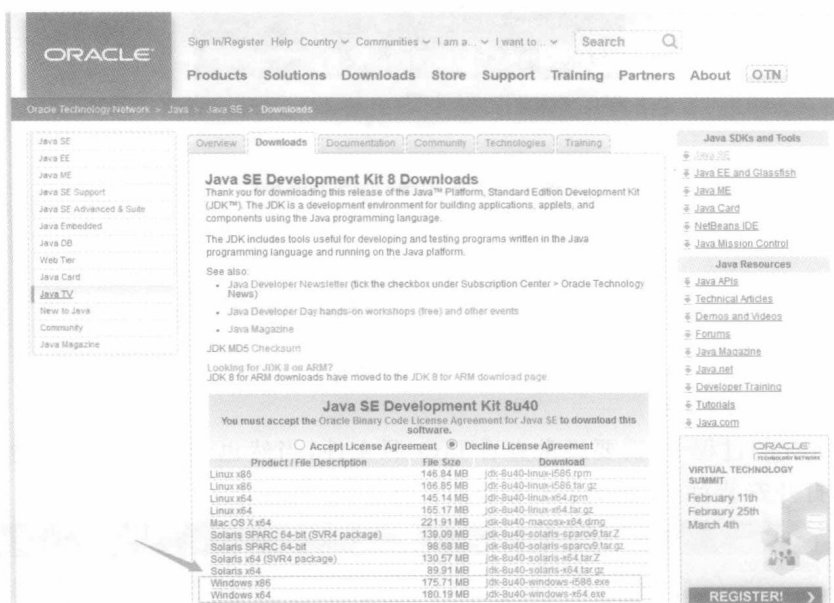


图 1-4 下载适合操作系统的 JDK 页面

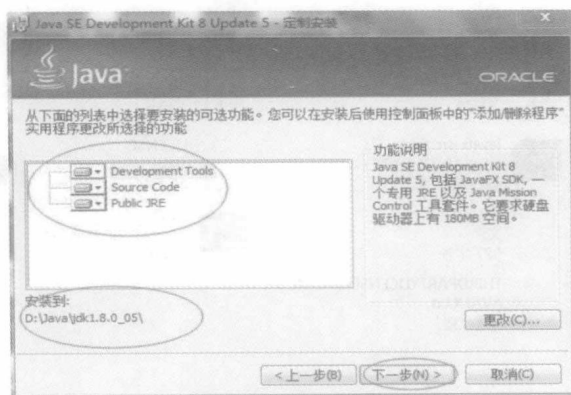


图 1-5 “定制安装”对话框

装的路径。单击“下一步”按钮,开始安装 JRE,设置安装目录为 D:\Java\jre8\,如图 1-6 所示。

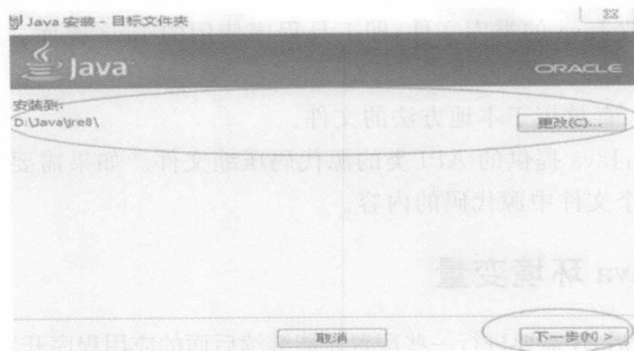


图 1-6 安装 JRE 运行环境

(5)安装成功后显示如图 1-7 所示对话框。单击“关闭”按钮,结束安装过程。

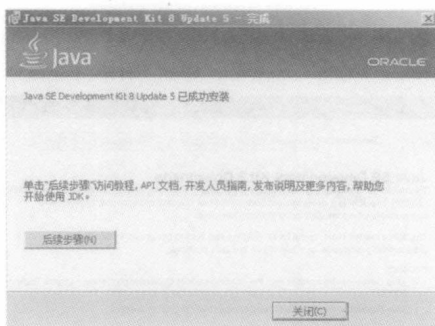


图 1-7 安装 JDK 成功

3. 了解 Java 安装目录

在学习 Java 的过程中,需要经常查看 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 1-8 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)jre 目录:存放 Java 运行环境文件。
- (3)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。
- (4)include 目录:存放用于本地方法的文件。
- (5)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,还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“D:\java\jdk1.8.0_05\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,变量值为:

```
. ; D:\java\jdk1.8.0_05\lib\dt.jar;D:\java\jdk1.8.0_05\lib\tools.jar;
```

注意:这是三个参数,第一个参数为“.”,参数间用分号间隔。

(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环境变量一般情况下不需要设置,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。如果配置成功,会出现当前JDK的版本号,如图1-10所示。

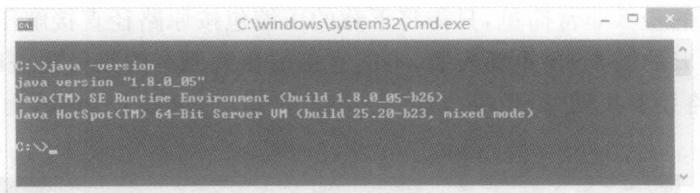


图 1-10 测试 JDK 是否成功

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。

祝贺你,已经成功地编译并运行了自己的第一个Java程序。

```
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

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 技术项目三个项目组成。

2. Eclipse 的获取与安装

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 所示界面。

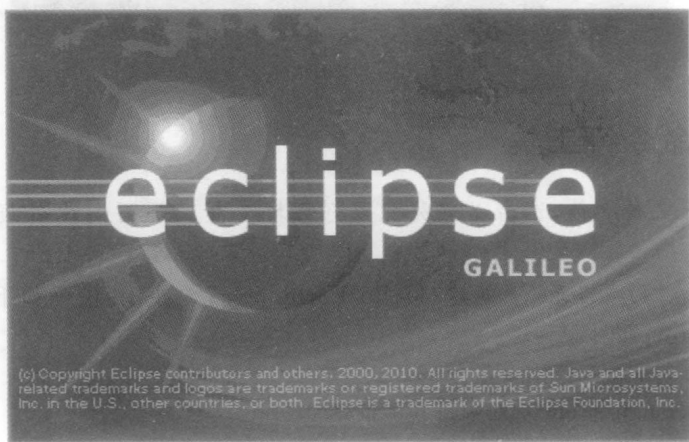


图 1-12 Eclipse 启动界面

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。

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。

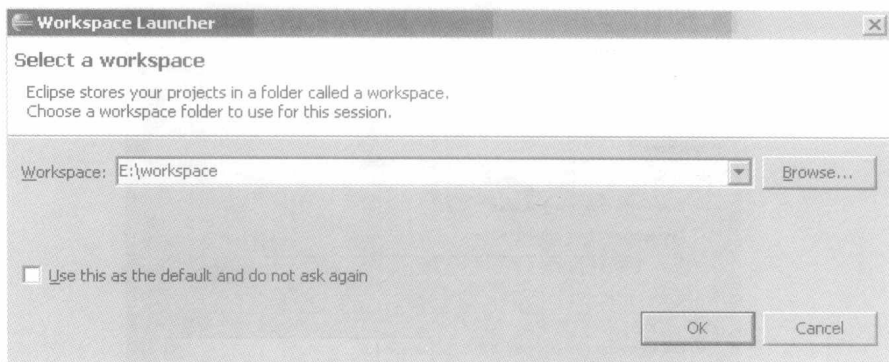


图 1-13 Eclipse 选择工作区路径

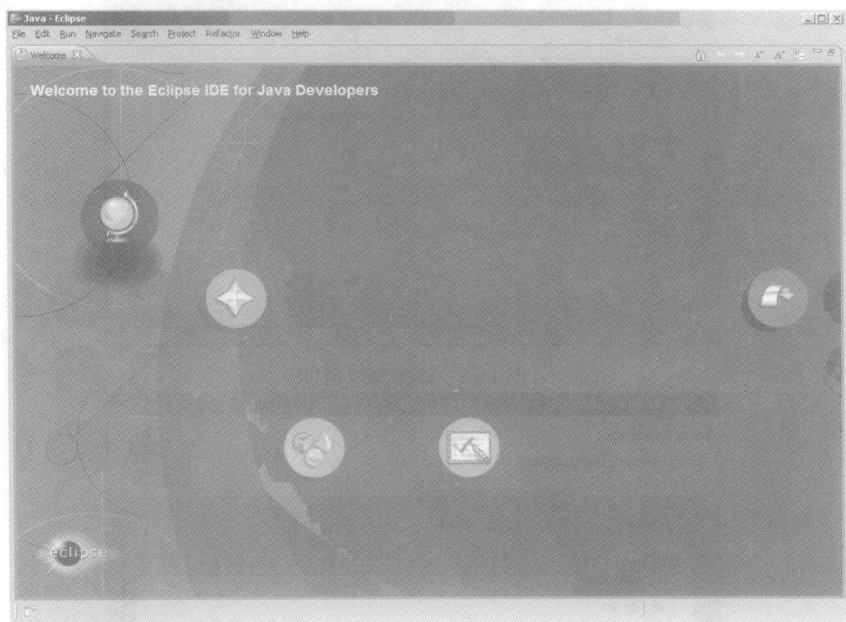


图 1-14 Eclipse 欢迎界面

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示项目的路径。使用的 JRE 环境以及项目布局,采用默认选项即可。输入项目名称后单击“Next”按钮,显示如图 1-16 所示窗口。

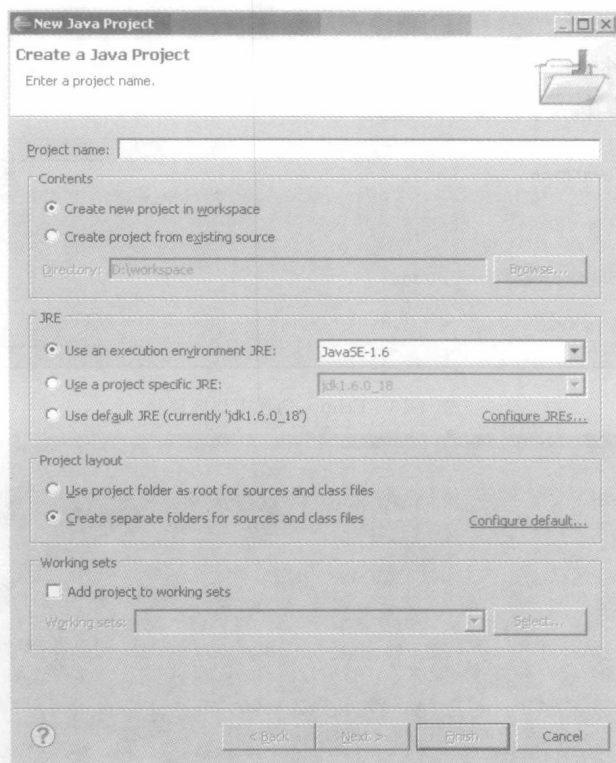


图 1-15 新建 Java 项目

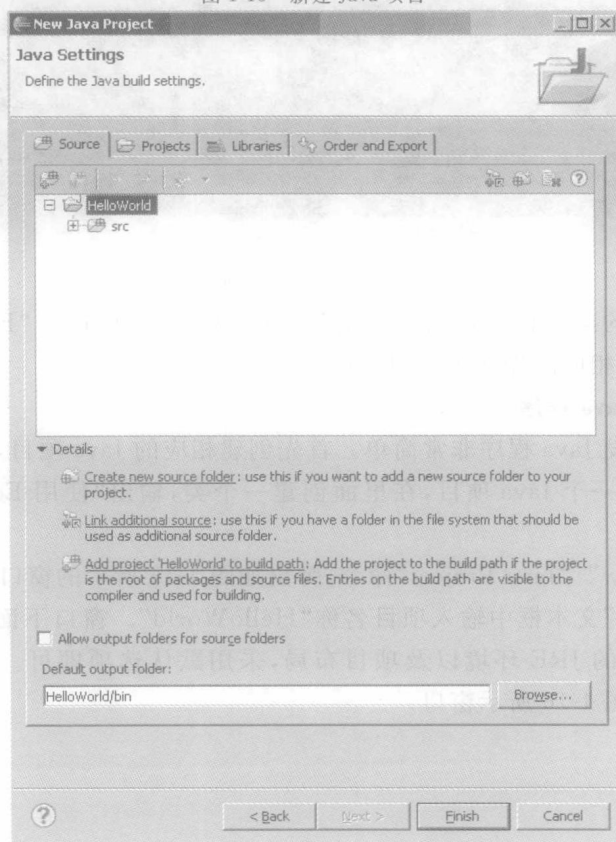


图 1-16 Java 项目设置