

21 世纪高等教育计算机规划教材

Java 编程技术 与项目实战

(第 2 版)

*JAVA PROGRAMMING TECHNOLOGY
AND PROJECT TRAINING
(2nd edition)*

王诚 梅霆 李琴 王峰 朱书眉 ◆ 编著



TP312JA
1738-2

21 世纪高等教育计算机规划教材

Java 编程技术 与项目实战

(第 2 版)

*JAVA PROGRAMMING TECHNOLOGY
AND PROJECT TRAINING
(2nd edition)*

王诚 梅霆 李琴 王峰 朱书眉 ◆ 编著

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

Java编程技术与项目实战 / 王诚等编著. -- 2版

. -- 北京 : 人民邮电出版社, 2015. 12

21世纪高等教育计算机规划教材

ISBN 978-7-115-40095-6

I. ①J… II. ①王… 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174140号

内 容 提 要

本书介绍了 Java 语言编程基础及 Java Web 开发的基本知识, 主要包括 Java 语言编程基础、面向对象编程基础、线程和网络编程、Java 图形用户界面、Java 数据库编程、JSP、Servlet、JavaBean 编程基础、MVC 模式及 Struts2 编程框架, 最后给出两个具体实例。本书基本覆盖了 Java 编程及 Java Web 编程的大部分实用技术。本书编排结合具体真实项目实例展开教学, 由浅入深, 重点突出。

本书在选择编排内容时注重实用性, 强调实践技能的提高, 同时也注重系统性、科学性和学习性, 每章末都有练习。本书层次分明, 理论联系实际, 用典型的案例来演示 Java 编程技术的魅力, 有利于培养工程应用及编程能力。本书既可作为高等学校计算机软件及应用、电子信息工程、通信工程、软件工程等相关专业的本科生教材, 也可供广大 Java 软件开发工程师学习参考。

◆ 编 著 王 诚 梅 霆 李 琴 王 峰 朱书眉

责任编辑 武恩玉

责任印制 沈 蓉 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本: 787×1092 1/16

印张: 16.5

2015 年 12 月第 2 版

字数: 435 千字

2015 年 12 月北京第 1 次印刷

定价: 39.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

第2版前言

当前, Internet 正以惊人的速度发展, 计算机技术、软件技术、物联网技术、网络技术、通信技术等 IT 领域新技术不断涌现, 使现在的生活出现了革命性的变化, 特别是物联网技术的兴起, 推动了各行业和部门的智能化、智慧化应用研究。在这些信息技术的应用中, Java 技术凭借其独有的、与平台无关的、与网络紧密结合的特点, 在企业级应用及移动终端的应用编程中发挥越来越重要的作用。

本书在介绍 Java 基本知识的基础上, 侧重介绍 Java Web 的框架知识, 并给出项目的原型实例。本书从最基本的下载配置和安装开始教学, 从最基础的数据类型、表达式、语句讲起, 引导读者进入面向对象的编程环节中; 中间贯穿了线程技术、网络技术、数据库技术、Java Web 的基本框架等方面的编程知识; 最后结合两个具体的实例来进行应用编程的总结。

本书并不要求读者逐章阅读，每一章内容都可独立学习，读者可以根据应用需要进行选择。本书的实例都是可以运行的，建议读者在学习过程中录入书中的例子以加强实践，以便更好、更快地掌握技术要领。

本书的先修课程包括 C/C++ 语言程序设计、计算机编程基础、网络技术、数据库技术等。如果读者学习并掌握 C/C++ 语言的使用,则学习本书可能会轻松一些,因为它们的语法结构是相似的。如果读者具备面向对象编程的基础,则更容易掌握 Java 编程技术。

本书在编写过程中得到了很多同行和读者的帮助，在此感谢南京邮电大学的陈杰、许晓、范向阳在整理资料和校对等方面所做的大量工作，以及宋文广、赵振文工程师提供的项目代码实例，在此一并表示感谢！

由于时间仓促和编者的知识所限，书中难免存在错误与不足之处，欢迎读者批评指正。作者联系方式为：wangc@njupt.edu.cn。

编者

2015 年 5 月

第 1 章 Java 语言概述及编程基础.....1

1.1 Java 语言概述.....1	
1.1.1 Java 语言发展.....1	
1.1.2 Java 语言的特点.....2	
1.1.3 Java 平台.....4	
1.2 Java 语言开发环境.....4	
1.2.1 Jdk 下载及安装.....4	
1.2.2 Java 开发环境设置.....6	
1.2.3 Java 程序的基本结构.....7	
1.2.4 MyEclipse 开发工具及使用.....8	
1.2.5 Tomcat 服务器的安装与配置.....11	
1.3 Java 语言编程基础.....13	
1.3.1 Java 基本数据类型.....13	
1.3.2 Java 标识符与关键字.....14	
1.3.3 运算符.....15	
1.3.4 程序控制语句.....18	
1.3.5 Java 异常处理.....21	
思考与习题.....24	

第 2 章 Java 面向对象技术基础.....25

2.1 类及对象.....25	
2.1.1 面向对象基本概念.....25	
2.1.2 类的定义.....26	
2.1.3 对象的创建和使用.....27	
2.1.4 构造方法.....28	
2.1.5 方法重载.....30	
2.1.6 类的成员和关键字 this.....32	
2.2 封装、继承与多态性.....33	
2.2.1 类的封装.....33	
2.2.2 类的继承.....34	
2.2.3 多态与方法重写.....35	
2.2.4 关键字 super.....36	
2.2.5 关键字 static.....37	

2.2.6 final 类和 abstract 类.....39	
----------------------------------	--

2.2.7 类的接口.....41	
-------------------	--

2.3 包.....43	
--------------	--

2.3.1 包的概念.....44	
-------------------	--

2.3.2 引入包.....44	
------------------	--

2.3.3 访问保护.....45	
-------------------	--

2.3.4 包的编译.....45	
-------------------	--

思考与习题.....46	
--------------	--

第 3 章 线程和网络编程.....47

3.1 线程概念.....47	
-----------------	--

3.1.1 Java 线程模型.....47	
------------------------	--

3.1.2 主线程.....49	
------------------	--

3.1.3 创建线程.....50	
-------------------	--

3.1.4 线程同步.....54	
-------------------	--

3.1.5 线程通信.....57	
-------------------	--

3.2 网络编程基础.....60	
-------------------	--

3.2.1 TCP/UDP.....60	
----------------------	--

3.2.2 端口.....60	
-----------------	--

3.2.3 套接字.....60	
------------------	--

3.2.4 客户机/服务器模式.....61	
------------------------	--

3.2.5 Java 和网络.....61	
-----------------------	--

3.2.6 InetAddress 类.....62	
----------------------------	--

3.2.7 URL.....63	
------------------	--

3.3 基于 TCP/UDP 的编程.....65	
---------------------------	--

3.3.1 TCP 编程模型与实例.....65	
--------------------------	--

3.3.2 UDP 编程模型与实例.....68	
--------------------------	--

思考与习题.....70	
--------------	--

第 4 章 Java 图形用户界面.....72

4.1 概述.....72	
---------------	--

4.1.1 图形用户界面.....72	
---------------------	--

4.1.2 组件.....73	
-----------------	--

4.2 Swing 组件.....74	
---------------------	--

4.2.1 文本组件	74	6.2.3 Servlet 生命周期	138
4.2.2 按钮组件	75	6.2.4 Servlet 表单数据	139
4.2.3 列表组件	75	6.3 创建 HttpServlet	142
4.3 布局管理器	76	6.4 JavaBean 技术	144
4.3.1 顺序布局	76	6.4.1 JavaBean 基础	144
4.3.2 边框布局	77	6.4.2 JavaBean 开发模式	145
4.3.3 网格布局	77	思考与习题	147
4.3.4 布局实例	78		
4.4 事件	80	第 7 章 MVC 模式和 Struts2	
4.4.1 事件处理机制	80	框架	148
4.4.2 鼠标和键盘事件	81	7.1 MVC 模式基础	148
4.4.3 事件源和监听器	83	7.1.1 MVC 模式简介	148
4.4.4 事件处理实现	86	7.1.2 模型、视图和控制器	149
4.5 界面编程实例	90	7.1.3 MVC 的实现	149
思考与习题	113	7.2 Struts2 框架基础	151
第 5 章 Java 数据库编程	114	7.2.1 Struts2 概述	151
5.1 数据库编程	114	7.2.2 Struts2 工作流程	152
5.1.1 MySQL 的安装	114	7.2.3 Struts2 配置文件	153
5.1.2 SQL 语言简介	117	7.2.4 Struts2 标签库	156
5.2 JDBC	118	7.3 Struts2 实现的 MVC 模式	158
5.2.1 JDBC 概念	119	7.3.1 Struts2 架构	158
5.2.2 系统编程模型	119	7.3.2 FilterDispatcher 核心过滤器	159
5.2.3 JDBC 接口及驱动	120	7.3.3 Action 详解	161
5.3 通过 JDBC 访问数据库	121	7.3.4 值栈与 OGNL 表达式	163
5.3.1 java.sql 包	121	7.3.5 结果与视图	165
5.3.2 编程模型及实例	123	7.4 Struts2 深入理解	169
5.3.3 解决中文乱码问题	128	7.4.1 拦截器	169
思考与习题	128	7.4.2 Struts2 验证框架	172
第 6 章 JSP、Servlet 和		7.5 Struts2 编程实例	174
JavaBean	129	7.5.1 Struts2 安装配置	174
6.1 JSP 基础	129	7.5.2 创建 Struts2 的 Web 应用	175
6.1.1 JSP 技术概述	129	思考与习题	178
6.1.2 JSP 基本语法	130		
6.1.3 JSP 内置对象	133	第 8 章 工业园区企业安全巡检	
6.2 Servlet 基础	136	系统	179
6.2.1 Servlet 简介	136	8.1 系统设计	179
6.2.2 Servlet 的类与接口	137	8.1.1 开发背景和需求分析	179
		8.1.2 系统目标与功能结构	180
		8.1.3 数据库设计	180

8.1.4 系统预览图	182
8.2 Spring 框架介绍	183
8.2.1 Spring 基础	183
8.2.2 Spring 骨骼架构	184
8.2.3 Bean 的装配	186
8.2.4 IoC 介绍	188
8.2.5 BeanFactory、Application Context	191
8.3 DWR 框架介绍	192
8.3.1 配置 web.xml 文件	193
8.3.2 配置 dwr.xml 文件	193
8.3.3 页面配置	195
8.3.4 系统代码示例	196
8.4 系统编程实例	202

第9章 精细化物资与人员管理

平台

9.1 平台设计	220
9.1.1 开发背景和需求分析	220
9.1.2 系统目标与功能结构	221
9.1.3 数据库设计	221
9.1.4 系统预览图	223
9.2 Mybatis 框架介绍	224
9.2.1 Mybatis 概述	224
9.2.2 Mybatis 组件	224
9.3 系统编程实例	239

参考文献

第 1 章

Java 语言概述及编程基础

Java 语言是 Sun 公司（已被 Oracle 公司收购）于 1995 年正式推出的面向对象（Object-oriented）的程序设计语言。Java 语言由于具有安全、跨平台、面向对象、简洁等特点，一经推出即引起了广大软件公司和程序员关注，受到计算机界的普遍接受与欢迎，成为目前网络时代最为流行的程序设计语言。Java 问世以来，其技术发展非常快，在计算机、移动电话、家用电器等领域都得到了广泛的应用。

本章将简单介绍 Java 语言的概述、Java 环境的安装与配置，以及基本的 Java 语法知识。通过本章的学习，同学们可以对 Java 语言的发展、特点和语法知识有基础的理解，并且可以在自己的工作机上安装搭建 Java 开发环境，为后续的深入学习做准备。

1.1 Java 语言概述

1.1.1 Java 语言发展

1991 年 4 月，Sun 公司的 James Gosling 领导的绿色计划（Green Project）开始着力发展一种分布式系统结构，使其能够在各种消费性电子产品上运行。他们研发了一种新的语言，该语言以 C 和 C++ 为基础，James 根据他在 Sun 公司办公室外的一棵橡树，而称其为 Oak 语言。后来发现已有一种称为 Oak 的计算机语言。当一些 Sun 公司的员工到当地一家咖啡店时，有人提议将该语言命名为 Java，从而使 Java 这个名字一直延续至今。

不过，在当时市场不成熟的情况下，他们的项目没有获得成功。市场对智能型电子装置需求的上升率并不像 Sun 公司所期盼的那样快，更糟的是 Sun 公司参加竞争的一个重要的销售合同被另一公司得去了。此时，Green 项目几乎处于被取消的境地。但很幸运的是，1993 年万维网（WWW）疯狂地流行起来，由于 Internet 的迅猛发展和 WWW 的快速增长，第一个全球信息网络浏览器 Mosaic 诞生了。此时，工业界非常迫切地需求一种适合在网络异构环境下使用的语言。James Gosling 决定改变绿色计划的发展方向，对 Oak 进行了小规模改造。就这样给该项目重新注入了生机，1995 年，Oak 语言更名为 Java 语言。Java 的诞生标志着互联网时代的开始。

Sun 公司于 1995 年 5 月在一个重要会议上正式发布了 Java。这样的事通常不会引起广泛的注意，但是由于万维网的商业利益，Java 立即引起了商业界的极大兴趣。目前，Java 被广泛应用于创建具有动态的、交互内容的 Web 网页，开发大规模企业应用程序，增强万维网服务的功能，向消费者的设备提供应用程序。

1998 年是 Java 迅猛发展的一年。在 1998 年 12 月 4 日, Sun 发布了 Java 历史上最重要的一个版本: JDK 1.2。这个版本的发布标志着 Java 已经进入 Java 2 时代。这个时期也是 Java 飞速发展的时期。在这一年中 Sun 发布了 JSP/Servlet、EJB 规范, 以及将 Java 分成了 J2EE、J2SE 和 J2ME。

2000 年 5 月 8 日, Sun 对 JDK 1.2 进行了重大升级, 推出了 JDK 1.3。2002 年 2 月 13 日, Sun 公司发布了 JDK 历史上最为成熟的 JDK 1.4 版本。2004 年 10 月, Sun 发布了期待已久的 JDK 1.5 版本, 同时将 JDK 1.5 改名为 J2SE 5.0。最新版本 JDK 8.0 加入了很多新特性, 是一款革命性开发平台, 全面升级了现有的 Java 流程模式。

Java 问世以来, 其技术发展非常快, 在计算机、移动电话、家用电器等领域都得到了广泛的应用。

1.1.2 Java 语言的特点

Java 语言的前身是在 C++ 的基础上开发的, 它继承了 C、C++ 语言的优点, 增加了一些实用的功能, 使 Java 语言更加精炼。Java 摒弃了 C、C++ 语言的缺点, 去掉了 C、C++ 语言中的指针运算、结构体定义、手工释放内存等容易引起错误的功能和特征, 增强了安全性, 也使其更容易被接受和学习。Java 是独立于平台, 面向 Internet 的分布式编程语言。

Java 是一种简单的、面向对象的、分布式的、解释执行的、健壮的、安全的、结构中立的、可移植的、高效率的、多线程的和动态的语言。

1. 简单

Java 是一种纯面向对象的语言, 它通过提供最基本的方法来完成指定的任务, 只需理解一些基本的概念, 就可以用它编写出适合于各种情况的应用程序。Java 略去了指针、运算符重载、多重继承等内容, 并且通过实现无用信息自动回收, 大大简化了程序设计者的内存管理工作。同时, Java 很小, 基本的解释器及类支持大约仅为 40KB。

2. 面向对象

Java 是一种纯面向对象的语言, Java 的核心是面向对象编程。事实上, 所有的 Java 程序都是面向对象的, 这一点与 C++ 语言不同, 因为在那里可以选择是否面向对象编程。Java 程序面向对象的设计思路不同于 C 语言基于过程的程序设计思路。面向对象程序设计, 具备更好地模拟现实世界环境的能力和可重用性。它将待解决的现实问题转换成一组分离的程序对象, 这些对象彼此之间可以进行交互。一个对象包含了对应实体应有的信息, 以及访问和改变这些信息的方法, 重点放在数据 (即对象) 和对象的接口上。通过这种设计方式, 所设计出来的程序更易于改进、扩展、维护和重用。Java 语言只支持类之间的单继承, 但支持接口之间的多继承, 并支持类与接口间的实现机制。

3. 分布式

Java 是一种分布式的语言。传统的基于 C/S (客户端/服务器) 架构的程序, 客户端向服务器提出服务请求, 服务器将程序执行结果返回, 所以, 服务器负荷较重。Java 采用 Java 虚拟机架构, 可将许多工作直接交由终端处理, 因此, 数据可以被分布式处理。此外, Java 类库的运用, 大大减轻了网络传输的负荷。Java 类库包含了支持 HTTP、FTP 等基于 TCP/IP 的子库。Java 应用程序可凭借 URL 打开并访问网络上的对象, 其访问方式与访问本地文件系统几乎完全相同。Java 的网络功能强大且易于使用, 特别是远程方法调用使得分布式对象之间可以互相通信。

4. 高效解释执行

Java 是高效解释执行的语言。高级语言程序必须转换为机器语言程序才能在计算机上执行。Java 程序在编译时并不直接编译成特定的机器语言程序, 而是编译成与系统无关的“字节码

(bytecode)”，由 Java 虚拟机 (Java Virtual Machine, JVM) 来执行。JVM 使得 Java 程序可以“一次编译，随处运行”。任何系统只有安装了 Java 虚拟机后，才可以执行 Java 程序。JVM 能直接在任何机器上执行，为字节码提供运行环境。

5. 健壮性

Java 是健壮的语言。Java 不需要指针就可以构造其他语言中需要指针构造的数据结构，即不会存取“坏的”指针而造成内存分配、内存泄露等错误。在传统的编程环境下，内存管理是一项困难、乏味的工作。例如，在 C 或 C++ 语言中，必须手工分配、释放所有的动态内存。如果忘记释放原来分配的内存，或是释放了其他程序正在使用的内存时，系统就会出错。同时，在传统的编程环境下，对异常情况必须用既繁琐又难理解的一大堆指令来进行处理。Java 通过自行管理内存分配和释放的方法，从根本上消除了有关内存的问题。Java 提供垃圾收集器，可自动收集闲置对象占用的内存。Java 提供面向对象的异常处理机制来解决异常处理的问题。

6. 安全

Java 是安全的网络编程语言，使用 Java 可以构建防病毒、防篡改的系统。Java 提供了一系列的安全机制以防恶意代码攻击，确保系统安全，如禁止运行时堆栈溢出、禁止在自己处理空间外破坏内存。Java 的安全机制分为多级，包括 Java 语言本身的安全性设计，以及严格的编译检查、运行检查和网络接口级的安全检查。

7. 结构中立

Java 是结构中立的语言。Java 的设计目标是要支持网络应用。Java 编译器会产生一种具备结构中立性的对象文件格式，即 Java 字节码文件。精心设计的 Java 字节码不仅很容易在任何机器上解释执行，还可以快速翻译成本机的代码。

8. 可移植性

结构中立是确保程序可移植的必要条件，此外，还需要很多其他条件的配合。Java 在可移植性方面做了许多工作。Java 通过定义独立于平台的基本数据类型及其运算，使数据得以在任何硬件平台上保持一致。例如，Java 中的 int 类型永远是 32 位的整数，这样就消除了代码移植时的主要问题。

9. 高效率

Java 是高效率的语言。每一次的版本更新，Java 在性能上均做出了改进。在历经数个版本变更后，Java 已经拥有与 C/C++ 同样甚至更好的运行性能。当 JVM 解释执行 Java 程序时，Java 实时编译器 (Just-In-Time, JIT) 会将字节码译成目标平台对应的机器语言的指令代码，并将结果进行缓存，因为 JVM 能够直接使用 JIT 编译技术将经过精心设计的字节码转换成高性能的本机代码，所示 Java 可以在非常低档的 CPU 上顺畅运行。

10. 多线程

Java 是支持多线程的语言。多线程可以带来更好的交换响应和实时行为。多线程是一种应用程序设计方法。线程是从大进程里分出来的、小的、独立的进程，使得在一个程序里可同时执行多项小任务。多线程带来的好处是具有更好的交互性能和实时控制性能，但采用传统的程序设计语言 (如 C/C++) 实现多线程非常困难。Java 实现了多线程技术，提供了一些简便地实现多线程的方法，并拥有一套高复杂性的同步机制。

11. 动态

Java 语言具有动态特性，能够适应不断发展的环境。Java 的动态特性是其面向对象设计方法的扩展，库中可以自由地添加新方法和实例变量，允许程序动态地装入运行过程中所需的类，Java 将

符号引用信息在字节码中保存后传递给解释器，再由解释器在完成动态连接类后，将符号引用信息转换为数据偏移量。存储器生成的对象不在编译过程中确定，而是延迟到运行时由解释器确定。

1.1.3 Java 平台

Java 推出了 3 个领域的应用平台：标准版 Java 2 Platform Standard Edition (Java SE)、企业版 Java 2 Platform Enterprise Edition (Java EE) 和微型版 Java 2 Platform Micro Edition (Java ME)。

Java SE 是各应用平台的基础。Java SE 可以分为 4 个主要部分：JVM、JRE、JDK 和 Java 语言。为了能运行 Java 程序，平台上必须安装有 Java 虚拟机 (Java Virtual Machine, JVM)。JVM 包含在 Java 运行环境 (Java SE Runtime Environment, JRE) 和 Java 开发包 (Java Development Kit, JDK) 等 Java 软件中。JDK 包括了 JRE 及开发过程中所需要的一些工具程序，如 Javac、Java、Appletviewer 等。安装 JRE 软件则可以运行 Java 程序；安装 JDK 软件则不但能运行 Java 程序，还可以编译开发 Java 程序。Java 语言只是 Java SE 的一部分。此外，Java 提供了庞大且功能强大的 API (Application Programming Interface) 类库，可以使用这些 API 作为基础进行程序开发，而无需重复开发功能相同的组件。

Java EE 以 Java SE 为基础，定义了一系列的服务、API、协议等，适用于开发分布式、多层次 (Multi-tiered)、以组件为基础、以 Web 为基础的应用程序。整个 Java EE 的体系是相当庞大的，比较常用的技术有 JSP、Servlet、Enterprise JavaBeans (EJB)、Java Remote Method Invocation (RMI) 等。

Java ME 是 Java 平台版本中最小的一个，是作为小型数字设备上开发及部署应用程序的平台，如消费型电子产品、嵌入式系统等。

1.2 Java 语言开发环境

为了能运行 Java 程序，平台上必须安装有 Java 虚拟机 (Java Virtual Machine, JVM)。JVM 包含在 Java 运行环境 (Java SE Runtime Environment, JRE) 和 Java 开发包 (Java Development Kit, JDK) 等 Java 软件中。JDK 包括了 JRE 及开发过程中所需要的一些工具程序，安装 JDK 软件则可以运行及编译开发 Java 程序。下面将介绍如何下载安装 JDK，并且介绍安装后的配置和编译方法。本小节还会介绍集成开发工具 MyEclipse 及 Tomcat 服务器的安装配置。

1.2.1 Jdk 下载及安装

JDK 是 Java 开发工具包。JDK 中包括 Java 编译器 (javac)、打包工具 (jar)、文档生成器 (javadoc)、查错工具 (jdb)，以及完整的 JRE (Java Runtime Environment, Java 运行环境)。

JDK 一般有 3 种版本，如表 1.1 所示。

表 1.1 JDK 版本

名 称	说 明
SE (Java SE)	Standard Edition, 标准版, 主要用于开发 Java 桌面应用程序
EE (Java EE)	Enterprise Edition, 企业版, 使用这种 JDK 开发 Java EE 应用程序, 用于 Web 方面
ME (Java ME)	Micro Edition, 微型版, 主要用于移动设备、嵌入式设备上的 Java 应用程序

在开发 Java 程序之前,先要在本机上安装 Java 程序开发工具包 JDK。安装好后的 JDK 具有 bin、demo、jre、lib、src 等子目录。在 Sun 公司的网站上有免费的 JDK 可供下载。下载及安装步骤如下所述。

(1) 打开浏览器,在地址栏输入 Sun 公司的网址: <http://java.sun.com/javase/downloads/index.jsp>, 进入下载页面,在该页面中选择合适的 Java 版本,如图 1.1 所示。

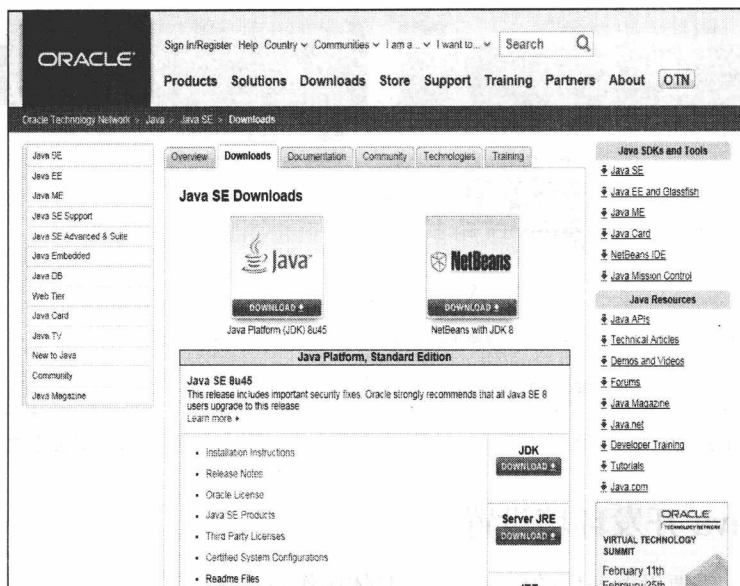


图 1.1 下载页面

(2) 在图 1.2 所示的页面中,显示的是不同平台下的 JDK 安装包,如 Windows、Linux、Solaris。找到适合于自己计算机平台的 JDK 版本,这里下载 `jdk-8u40-windows-i586.exe` 可执行程序。需要注意的是,在下载 JDK 工具包之前,要选择接受协议。

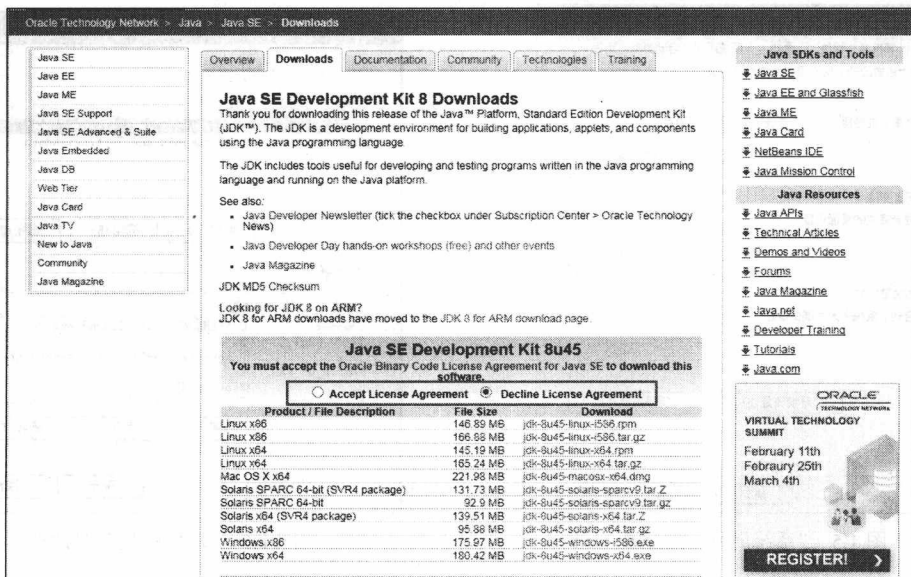


图 1.2 选择安装包

(3) 下载好后双击图标,弹出安装向导对话框,在这里设置 JDK 的安装路径,可以自行更改路径,也可以使用默认路径,如图 1.3 所示。

(4) 单击“下一步”按钮,开始安装 JDK,如图 1.4 所示。

(5) 在安装过程中会出现安装 Java 运行环境的对话框,选择默认设置不做改变,单击“下一步”按钮,继续安装,安装成功后会出现相应的提示信息。

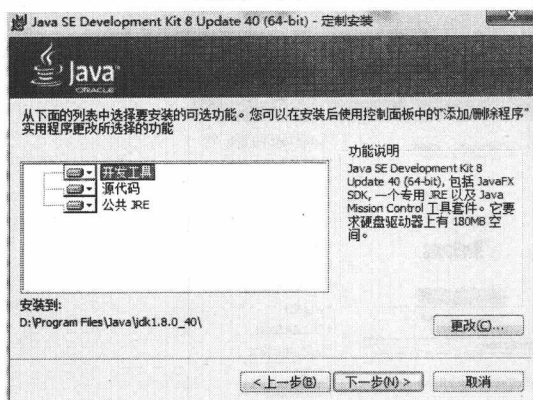


图 1.3 选择安装路径

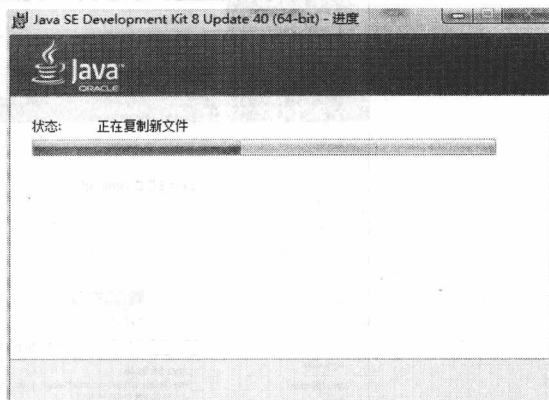


图 1.4 安装进度

1.2.2 Java 开发环境设置

JDK 安装后,需要设置开发的环境变量,具体操作与步骤如下。

(1) 在“我的电脑”上单击鼠标右键,在弹出的快捷菜单中选择“属性”命令,弹出“系统属性”对话框,选择“高级”选项卡,如图 1.5 所示。

(2) 单击“环境变量”按钮,在打开的“环境变量”对话框中,单击“新建”按钮,如图 1.6 所示。

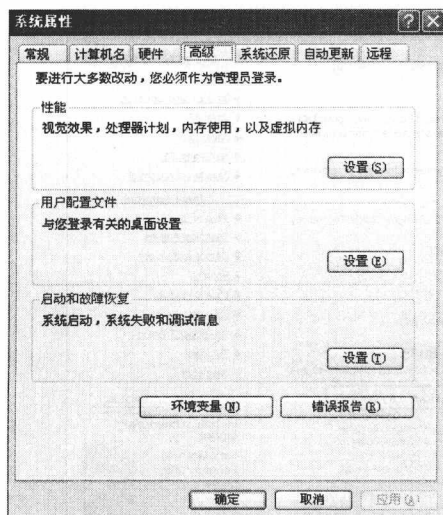


图 1.5 “系统属性”对话框

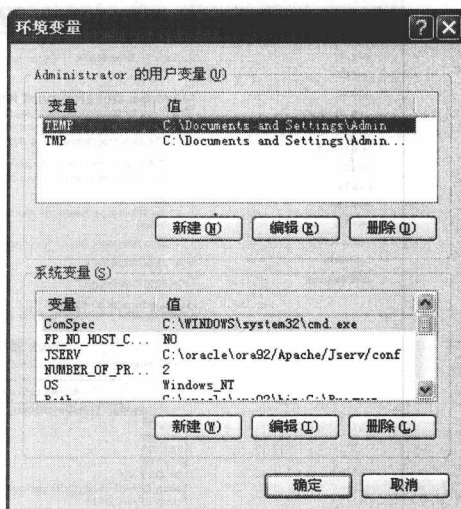


图 1.6 “环境变量”对话框

(3) 在弹出的“新建用户变量”对话框中创建一个新的系统变量名“JAVA_HOME”，其值

为 JDK 的安装路径。单击“确定”按钮保存,如图 1.7 所示。

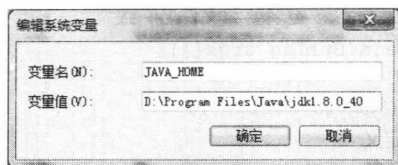


图 1.7 新建用户变量

(4) 在图 1.6 中选中“Path”变量名,单击“确定”按钮,在弹出的“编辑系统变量”对话框中设置变量值为“%JAVA_HOME%\bin”,单击“确定”按钮保存,如图 1.8 所示。

(5) 新建“CLASSPATH”变量名,设置变量值为“%JAVA_HOME%\lib\tools.jar”,单击“确定”按钮保存,如图 1.9 所示。

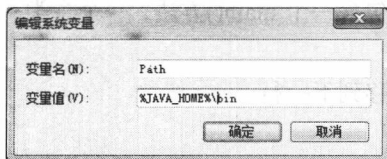


图 1.8 编辑系统变量

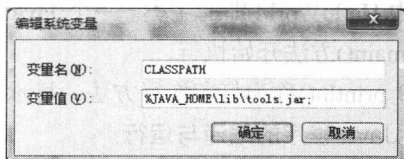


图 1.9 新建用户变量

(6) 验证变量设置的正确性。选择“开始”→“运行”命令,在运行窗口中输入“cmd”命令,打开命令行编辑器窗口,在该窗口中分别输入“javac”(编译器)和“java”(解释器)命令,并按 Enter 键运行这两条命令。如果可以看到如下的帮助信息,则说明 JDK 的安装和设置是正确的,如图 1.10 所示。

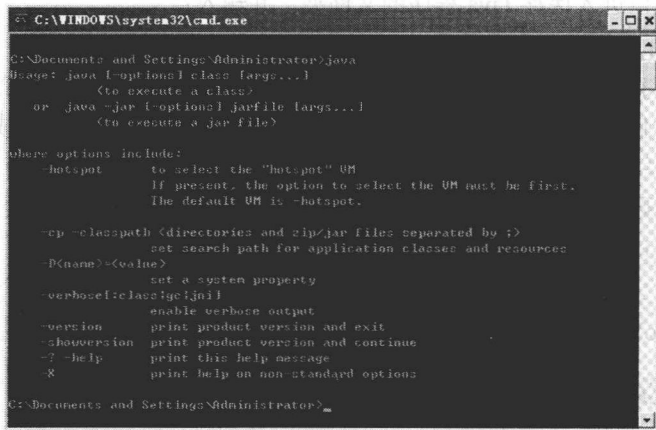


图 1.10 cmd 窗口

1.2.3 Java 程序的基本结构

用 Java 语言可以编写两种程序,一种是应用程序(Application),另一种是小应用程序(Applet)。应用程序是基于桌面型的应用,可以独立运行,也可以用在网络、多媒体等开发上。小应用程序是一种特殊的 Java 程序,不可以独立运行,是嵌入到 Web 网页中,且由带有 Java 插件的浏览器解释运行,主要用在 Internet 上。

下面所示是一段简单的 Java 应用程序 HelloWorld.java。

```
public class HelloWorld { //类名要和文件名一致
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    } //结束 main 方法的定义
} //结束类 HelloWorld 的定义
```

1. HelloWorld 程序中基本语法

(1) 程序中关键词 `class` 定义类，其后大括号之间的语句构成了一个类体。

(2) 关键词 `public` 表示“公有”，若类的对象或变量被定义为 `public`，表示该对象或变量可以被外界访问；若定义为 `private`，则不能被外界访问。

(3) `main()` 方法又称为主方法，必须被说明为 `public static void`。包含 `main()` 方法的类为主类，该程序中 HelloWorld 即为主类。一个 Java 应用程序只能包含一个 `main()` 方法，一个主类，程序从主类的 `main()` 方法开始执行。

(4) `println()` 称为打印换行方法，显示文字后光标将移到下一行。

2. Java 程序的编译与运行

(1) 编辑源程序。Java 源程序一般用 Java 作为扩展名，是一个文本文件，用 Java 语言写成，可以用任何文本编辑器创建与编辑。

例如，打开记事本编辑器，编辑 HelloWorld 代码，存放在特定文件夹。

(2) 编译源程序。HelloWorld.java 程序不能直接运行，运行前使用“`javac`”编译器，读取 Java 源程序，并翻译成 Java 虚拟机能够明白的指令集合，并以字节码的形式保存在文件中。通常，字节码文件以 `class` 作为扩展名。

例如，在命令窗口进入保存 Java 程序的文件夹，并输入：

```
Javac HelloWorld.java
```

便将 HelloWorld.java 文件编译成 HelloWorld.class 类文件，并存放在当前文件夹。

(3) 解释执行，使用“`java`”解释器，读取字节码，取出指令并且翻译成计算机能执行的代码，完成运行过程。字节码运行的平台是 Java 虚拟机，只要计算机上安装有 Java 虚拟机，不论采用哪种操作系统，硬件配置如何不同，运行的结果都一样。

例如，在命令窗口输入：

```
Java HelloWorld
```

便可以运行 HelloWorld.class 文件，得到程序结果。

1.2.4 MyEclipse 开发工具及使用

MyEclipse 是在 Eclipse 基础上加上插件开发而成的集成开发环境，主要用于 Java、Java EE 及移动应用的开发。单纯的 Eclipse 通常进行 Java Web 的开发时，需要安装 Eclipse 的插件，众多软件厂商和开源组织开发了相应的插件，其中 MyEclipse 最为常用。

1. 下载和安装 MyEclipse

下载 MyEclipse 可以通过官方网站链接 <http://www.myeclipsecn.com/>，也可以通过网络中的资源搜索，由于 MyEclipse 是一款非开源的开发工具，所以只能在官网中下载试用版本。图 1.11 给出了 MyEclipse Professional 2013 版本的下载安装，单击“Next”按钮进行安装。

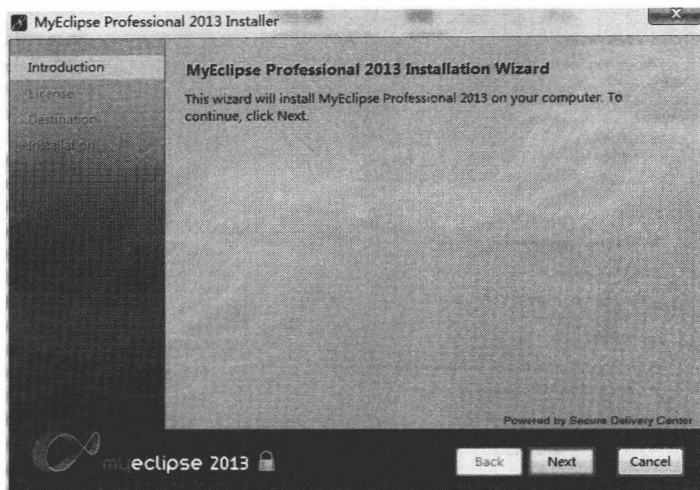


图 1.11 MyEclipse Professional 2013 安装

安装时需要手动选择工作空间目录，完成后启动 MyEclipse，工作界面如图 1.12 所示。

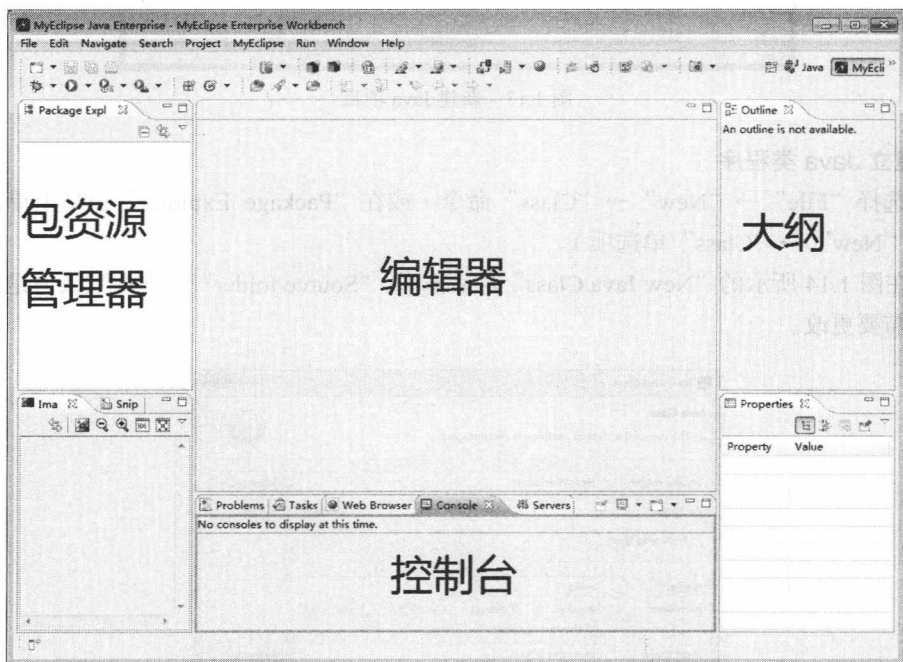


图 1.12 MyEclipse 工作界面

2. 建立 Java 项目

(1) 选择 “File” → “New” → “Project” 命令（或是在 “Package Explorer” 窗口上单击鼠标右键，选择 “New” → “Project” 单选项）。

(2) 在 “New Project” 对话框中，选择 “Java Project” 选项（或展开 “Java” 的数据夹，选择 “Java Project” 选项）。

(3) 在 “New Java Project” 对话框中，输入 Project 的名称及创建 Project 位置（一般选择在工作空间中创建，然后单击 “Finish” 按钮，如图 1.13 所示）。

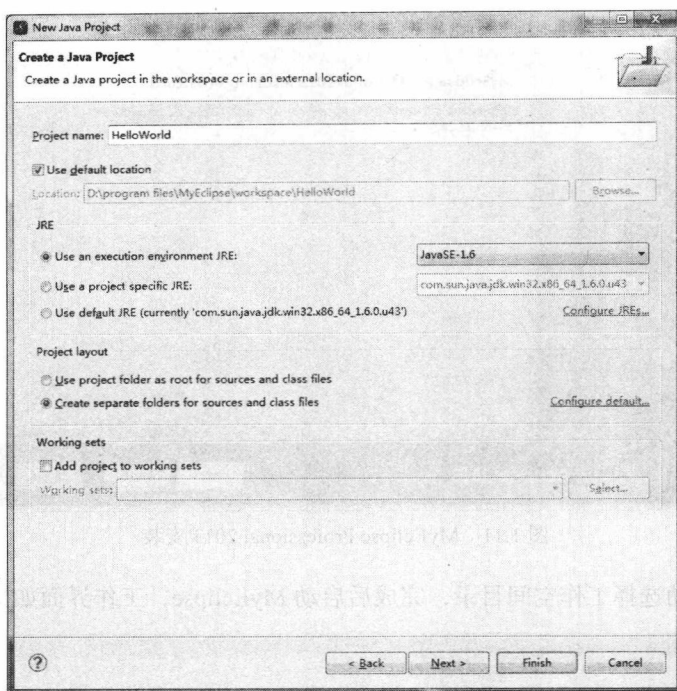


图 1.13 新建 Java 项目

3. 建立 Java 类程序

(1) 选择“File”→“New”→“Class”命令（或在“Package Explorer”窗口上单击鼠标右键，选择“New”→“Class”单选项）。

(2) 在图 1.14 所示的“New Java Class”对话框中，“Source folder”文本框默认值是项目的数据夹，不需要更改。



图 1.14 建立 Java 类