

中国电子教育学会高教分会推荐
普通高等教育电子信息类“十三五”课改规划教材

Java 实用教程

主 编 靳宗信 郑良仁
副主编 李志民 朱 强 樊红娟



西安电子科技大学出版社
<http://www.xduph.com>

内 容 简 介

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

Java 实用教程

主 编 靳宗信 郑良仁

副主编 李志民 朱 强 樊红娟

本书共24讲，主要介绍了Java的基本语法、方法和数组、面向对象基础、访问控制、主程序类、继承和多态、成员的初始化、抽象类和接口、内部类、异常处理和断言、多线程、输入/输出(I/O)、Java Swing、网络编程、泛型、数据库操作等内容。基本涵盖Java基础中的所有知识点。

第1讲介绍了Java的历史、作用和重要性，并对开发环境的安装和配置进行了详细的讲解。第2讲和第3讲是Java的基本语法介绍，包括标识符、关键字、各种运算符、数据类型等知识点。第4讲首先讲述了数组的创建和使用，包括一维数组、二维数组、字符串数组、方法的调用、参数传递和对象的引用。第5讲介绍了Java的类，包括Scanner类、String类、StringBuffer类等。第6讲是面向对象的三大特性：封装、继承和多态。第7讲介绍了关键字和final关键字。第8讲介绍了类中成员变量的作用域和性质。第9讲介绍了接口和静态内部类等。第10讲和第11讲主要讲述了多线程编程，包括线程的创建、启动、终止、同步、死锁、线程局部变量、线程池等。第12讲介绍了线程的序列化。第13讲和第14讲主要讲述了Java Swing图形用户界面，包括窗口的布局管理、各种基本的组件及组件的事件处理、每个示例程序都给出了完整的源代码。第15讲介绍的是有关网络编程的知识，包括InetAddress类、URL类、URLConnection类、Socket通信和无连接的数据报等内容。第16讲、17讲、18讲主要讲述的是泛型和集合，包括自定义泛型类、泛型接口、泛型方法、集合框架和Map等。第19讲介绍了枚举、线性表和队列等的性能。第20讲介绍的是Java的数据库操作，包

西安电子科技大学出版社

内 容 简 介

本书系统介绍了 Java 语言的基本概念、基本语法和编程方法,较详尽地阐述了 Java 语言的各知识点。本书主要包括方法和数组、访问控制、系统常用类、继承和多态、成员的初始化、抽象类和接口、内部类、异常处理和断言、多线程、输入/输出、Java Swing、网络编程、泛型、集合、数据库操作等内容。

书中每个知识点都配有案例程序,这些程序都经过 Eclipse 编译且正确运行。

本书可作为高等院校计算机类相关专业、电子信息类专业、通信类专业及其他专业的 Java 程序设计课程的配套教材,也可作为大中专院校、成人高校、计算机类培训机构的配套教材,还可作为自学者和软件开发人员的参考用书。

图书在版编目(CIP)数据

Java 实用教程/靳宗信,郑良仁主编. —西安:

西安电子科技大学出版社,2016.8

普通高等教育电子信息类“十三五”课改规划教材

ISBN 978 - 7 - 5606 - 4171 - 3

I. ① J… II. ① 靳… ② 郑 III. ① JAVA 语言—程序设计
—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2016)第 176454 号

策 划 刘小莉

责任编辑 祝婷婷 马武装

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 16.5

字 数 389 千字

印 数 1~3000 册

定 价 35.00 元

ISBN 978 - 7 - 5606 - 4171 - 3/TP

XDUP 4463001 - 1

*** 如有印装问题可调换 ***

前言

Java 语言是非常优秀的面向对象程序设计语言，不仅可以用于开发桌面应用程序、Web 应用程序，还可以用于开发 Android APP，现在已经成为全球最受程序员欢迎的程序设计语言。本书作者有着十几年的 Java 开发经验，书中例题都是经过作者深思熟虑、精心设计的，既能帮助读者理解知识点，同时又能启发读者，达到举一反三的效果。

本书共 24 讲，主要介绍了 Java 的基本语法、方法和数组、面向对象基础、访问控制、系统常用类、继承和多态、成员的初始化、抽象类和接口、内部类、异常处理和断言、多线程、输入/输出(I/O)、Java Swing、网络编程、泛型、集合、数据库操作等内容，基本涵盖了 Java 基础中的所有知识点。

第 1 讲介绍了 Java 的历史、作用和重要性，并对开发环境的安装和配置进行了详细的讲解。第 2 讲和第 3 讲是 Java 的基本语法介绍，包括标识符、关键字、各种运算符、流程控制等知识点。第 4 讲首先讲述了数组的创建和使用，包括一维数组和二维数组，其次介绍了方法的调用、参数的传递和方法的重载。第 5 讲主要介绍了面向对象程序设计、类的创建和对象的引用。第 6 讲介绍的是包和访问权限修饰符。第 7 讲和第 8 讲讲述了系统常用类，包括 Scanner 类、字符串、数据包装类、日期和时间类、格式化输出类、Math 类、Class 类等。第 9 讲是面向对象程序设计中的重点和难点，讲述了组合和继承的基本语法，super 关键字和 final 关键字的用法，以及向上转型和动态绑定等比较难理解的概念。第 10 讲主要对类中成员变量的初始化方法和顺序进行了讨论。第 11 讲讲述了抽象类、抽象方法以及接口的性质。第 12 讲介绍的是有关内部类的相关知识，包括成员类、局部内部类、匿名内部类和静态内部类等内容。第 13 讲主要是对 Java 中的异常处理机制以及断言的讲解。第 14 讲和第 15 讲主要讲述了 Java 中的多线程机制，包括多线程的实现、创建线程池、线程的调度、访问冲突、线程间的协作以及死锁等内容。第 16 讲和第 17 讲介绍了 Java 的输入/输出，包括 File 类的使用、流的概念、字节流和字符流、文件的随机存取、标准输入/输出、对象的序列化和反序列化等内容。第 18 讲到第 20 讲是 Java Swing 部分，包括各种容器及容器的布局管理，各种基本的组件及组件的事件处理，每个示例程序都是经过作者精心挑选的。第 21 讲介绍的是有关网络编程的知识，包括 InetAddress 类、URL 类、URLConnection 类、Socket 通信和无连接的数据报等内容。第 22 讲、23 讲主要讲述的是泛型和集合，包括自定义泛型类、泛型接口、泛型方法、受限的泛型、通配泛型、集合框架和 Map 等内容，并比较了规则集、线性表和队列等的性能。第 24 讲介绍的是 Java 的数据库操作，包

括对 JDBC 和数据库操作步骤的讲解,并以一个完整的例子说明了如何使用 Java 对数据库中的数据进行增、删、改、查操作。

本书由黄河科技学院的靳宗信老师和郑良仁老师担任主编，中原工学院信息商务学院的李志民老师、朱强老师和河南省劳动干部学校的樊红娟老师担任副主编。书中全部案例都经过 Eclipse 编译并正确运行。有兴趣的读者可以向编者索取源代码和课件，编者邮箱：jinzongxin@163.com。

由于编者水平有限，书中难免有不足之处，欢迎广大读者批评指正。

编者

2016 年 4 月

目 录

第 1 讲 集成 Java 入门	1
1.1 Java 的历史	1
1.2 Java 的用处	2
1.3 Java 开发环境的安装和配置	3
1.3.1 JDK + EditPlus	3
1.3.2 配置环境变量	6
1.3.3 集成开发环境 (IDE)	7
1.4 第一个 Java 程序	7
1.5 Java 程序的开发过程	9
1.6 怎么学好 Java	9
1.7 本讲小结	10
课后练习	10
第 2 讲 Java 基本语法(一)	11
2.1 标识符和关键字	11
2.2 基本数据类型	12
2.3 变量和常量	13
2.3.1 变量	13
2.3.2 定名常量	13
2.3.3 直接常量	14
2.3.4 指数计数法	14
2.4 赋值	14
2.5 运算符	14
2.5.1 算术运算符	14
2.5.2 关系运算符	16
2.5.3 逻辑运算符	17
2.5.4 按位运算符	18
2.5.5 移位运算符	18
2.5.6 其他运算符	19
2.6 本讲小结	20
课后练习	20
第 3 讲 Java 基本语法(二)	21
3.1 分支	21
3.2 循环	22

3.3 跳转	24
3.4 开关	24
3.5 本讲小结	25
课后练习	25
第 4 讲 方法和数组	26
4.1 数组	26
4.1.1 一维数组	26
4.1.2 二维数组	31
4.2 方法	33
4.2.1 方法的调用	33
4.2.2 变量的作用域	34
4.2.3 参数的传递	35
4.2.4 方法重载	36
4.3 本讲小结	36
课后练习	36
第 5 讲 初识面向对象	37
5.1 面向过程程序设计 vs 面向对象程序设计	37
5.2 创建新的数据类型	37
5.3 类的成员	39
5.3.1 成员变量	39
5.3.2 成员方法	40
5.4 构造方法	41
5.5 通过引用访问对象	41
5.5.1 引用类型和引用类型变量	42
5.5.2 引用类型变量和基本类型变量的区别	42
5.5.3 点语法	42
5.5.4 再论参数传递	43
5.6 关键字 this	44
5.7 对象数组	45
5.8 数据的存储	46
5.9 本讲小结	46
课后练习	47

第 6 讲 访问控制	48	9.2 继承	71
6.1 包	48	9.2.1 继承的语法	71
6.1.1 系统的包	48	9.2.2 父类的初始化	72
6.1.2 自己创建的包	48	9.2.3 再论方法重载	74
6.1.3 打包	49	9.2.4 变量的隐藏和方法的重写	75
6.2 访问权限修饰符	51	9.2.5 super 关键字	75
6.2.1 成员的访问权限	51	9.3 final 关键字	77
6.2.2 类的访问权限	51	9.3.1 final 修饰的变量	77
6.3 本讲小结	52	9.3.2 空白 final	77
课后练习	52	9.3.3 final 修饰的方法	78
第 7 讲 系统常用类(一)	53	9.3.4 final 修饰的参数	79
7.1 Scanner 类	53	9.3.5 final 修饰的类	79
7.2 字符串	53	9.4 多态	79
7.2.1 不可变长字符串	53	9.4.1 向上转型	80
7.2.2 可变长字符串	54	9.4.2 动态绑定	81
7.2.3 字符串的比较	56	9.4.3 多态的好处	82
7.3 数据包装类	57	9.4.4 多态的缺陷	83
7.4 本讲小结	58	9.5 本讲小结	84
课后练习	58	课后练习	84
第 8 讲 系统常用类(二)	59	第 10 讲 成员的初始化	85
8.1 日期和时间类	59	10.1 定义初始化	86
8.1.1 Date 类	59	10.2 构造方法初始化	86
8.1.2 DateFormat 类	59	10.3 实例语句块	87
8.1.3 SimpleDateFormat 类	60	10.4 静态数据的初始化	87
8.1.4 Calendar 类	61	10.5 静态语句块	88
8.1.5 GregorianCalendar 类	62	10.6 类的加载及初始化顺序	88
8.2 格式化输出类	62	10.7 本讲小结	91
8.2.1 printf() 和 format()	63	课后练习	91
8.2.2 String.format()	63	第 11 讲 抽象类和接口	92
8.2.3 Formatter 类	63	11.1 抽象类和抽象方法	92
8.3 Arrays 类	64	11.2 接口	93
8.4 Math 类	65	11.2.1 接口中的域和方法	93
8.5 System 类	66	11.2.2 接口的实现	93
8.6 Random 类	67	11.2.3 扩展接口	94
8.7 Class 类	67	11.2.4 嵌套接口	96
8.8 本讲小结	68	11.2.5 接口的好处	97
课后练习	68	11.3 本讲小结	99
第 9 讲 继承和多态	70	课后练习	99
9.1 组合	70	第 12 讲 内部类	100
		12.1 成员类	100

12.2 局部内部类	102	第 15 讲 Java 多线程(二)	135
12.3 匿名内部类	103	15.1 访问共享资源	135
12.4 静态内部类	104	15.1.1 访问冲突	135
12.5 内部类的继承	105	15.1.2 解决冲突	136
12.6 内部类的好处	105	15.1.3 静态方法同步	137
12.7 本讲小结	106	15.2 线程间协作	137
课后练习	106	15.3 死锁	140
第 13 讲 异常处理和断言	107	15.4 本讲小结	142
13.1 Java 的异常	107	课后练习	142
13.2 异常处理机制	108	第 16 讲 输入/输出(一)	143
13.2.1 捕获异常	108	16.1 File 类	143
13.2.2 finally	110	16.2 文件过滤器	144
13.2.3 声明异常	112	16.3 流	145
13.2.4 抛出异常	113	16.4 字节流和缓冲字节流	145
13.3 捕获所有异常	113	16.5 字符流和缓冲字符流	147
13.3.1 异常轨迹	113	16.6 本讲小结	148
13.3.2 重新抛出异常	115	课后练习	148
13.3.3 异常链	118	第 17 讲 输入/输出(二)	149
13.3.4 异常的丢失	119	17.1 文件随机存取	149
13.4 自定义异常	120	17.2 标准输入/输出	150
13.5 异常的限制	121	17.3 对象的序列化和反序列化	152
13.6 断言	122	17.4 本讲小结	153
13.7 本讲小结	123	课后练习	154
课后练习	123	第 18 讲 Java Swing	155
第 14 讲 Java 多线程(一)	125	18.1 Swing 入门	156
14.1 Java 中的线程	125	18.1.1 一组例子	156
14.2 Java 多线程的实现	126	18.1.2 显示框架	159
14.3 线程池	128	18.2 容器	160
14.3.1 固定尺寸线程池	128	18.2.1 顶层容器	160
14.3.2 可变尺寸线程池	129	18.2.2 中间层容器	161
14.3.3 单任务线程池	129	18.3 布局管理	169
14.4 线程的调度	130	18.3.1 BorderLayout	169
14.4.1 线程休眠	130	18.3.2 FlowLayout	170
14.4.2 线程优先级	131	18.3.3 GridLayout	171
14.4.3 线程让步	132	18.3.4 CardLayout	172
14.4.4 线程合并	132	18.3.5 BoxLayout	174
14.5 前台线程和后台线程	133	18.3.6 绝对布局	175
14.6 本讲小结	134	18.4 基本组件	177
课后练习	134	18.4.1 AbstractButton	177

18.4.2 菜单	179	第 22 讲 泛型	230
18.4.3 标签和文本编辑组件	180	22.1 泛型	230
18.4.4 组合框和列表框	182	22.2 自定义泛型类和接口	231
18.4.5 滑块和进度条	183	22.3 自定义泛型方法	232
18.4.6 选择框	185	22.4 受限的泛型	233
18.4.7 表格和树	186	22.5 原始类型和向后兼容	234
18.5 本讲小结	189	22.6 通配泛型	234
课后练习	189	22.7 本讲小结	237
第 19 讲 事件处理(一)	190	课后练习	237
19.1 Java 的事件处理机制	190	第 23 讲 集合	238
19.2 动作事件	193	23.1 集合框架	238
19.3 调整事件和改变事件	196	23.2 Collection	239
19.4 选择事件	197	23.2.1 Set(规则集)	239
19.5 文本事件	200	23.2.2 Comparator(比较器接口)	241
19.6 本讲小结	201	23.2.3 List(线性表)	242
课后练习	201	23.2.4 Queue(队列)	243
第 20 讲 事件处理(二)	202	23.2.5 规则集和线性表的性能比较	246
20.1 焦点事件	202	23.3 Map	247
20.2 窗口事件	204	23.4 本讲小结	248
20.3 鼠标事件	206	课后练习	248
20.4 键盘事件	209	第 24 讲 数据库操作	249
20.5 适配器类	210	24.1 JDBC	249
20.6 本讲小结	211	24.2 结果集及常见方法	250
课后练习	211	24.3 操作数据库步骤	252
第 21 讲 Java 网络编程	212	24.4 一个例子	253
21.1 InetAddress 类	212	24.5 本讲小结	255
21.2 URL 类	213	课后练习	255
21.3 URLConnection 类	216	参考文献	256
21.4 Socket 通信	217		
21.5 无连接的数据报	222		
21.6 本讲小结	229		
课后练习	229		

第1讲 集成Java入门

Java 是一种优秀的编程语言，要学好 Java，首先要了解它的历史，配置好开发环境，了解它的优缺点，找到适合自己的学习方法。

1.1 Java 的历史

伟大的程序员 Larry Wall 曾说过优秀的程序员应该具备三个美德：懒惰、急躁和傲慢。Java 语言就是在一群懒惰、急躁而傲慢的程序天才手中诞生的。

1990 年 12 月，Sun 的工程师 Patrick Naughton 快被当时糟糕的 Sun C++ 工具折磨疯了。他大声抱怨，并威胁要离开 Sun 转投当时在 Steve Jobs 领导之下的 NeXT 公司。领导层为了留住他，给他一个机会，启动了一个叫做 Stealth(秘密行动)的项目。随着 James Gosling 等人的加入，这个项目更名为 Green。该项目旨在为家用电器提供支持，使这些电器智能化并且能够彼此交互。Bill Joy、James Gosling、Mike Sheridan 和 Patrick Naughton 是该项目的核心成员(见图 1.1)。



图 1.1 Green 项目组的核心成员

正如人们事后分析的那样，这些天才的程序员太懒惰，所以没有把 C++ 语言学好，开发中碰了一头包；太急躁，所以都不愿意停下来读读 Scott Meyers 的新书《Effective C++》；太傲慢，所以轻易地决定开发一种新的编程语言。他们把这种语言命名为 C+++-，意思是 C++“加上一些好东西，减去一些坏东西”。显然这个糟糕的名字不可能长命百岁，很快这种颇受同伴喜爱的小语言被命名为 Oak(橡树，1991 年 2 月)，因为他们办公室窗外有一棵橡树。

1992 年 3 月，由于 Oak 已被用作另一种已存在的编程语言名称，因此必须选一个新的



名字——它就是 Java，其灵感来源于咖啡。

1995 年 5 月，在 SunWorld 大会上，Sun 公司正式介绍了 Java 和 HotJava。

1996 年 1 月，JDK 1.0 版本发布，1.0 版本的 Java 只能做一些简单的事情。

1997 年 2 月，JDK 1.1 版本发布。其主要特点是增加了 JDBC (Java Data Base Connectivity, Java 数据库连接)、RMI (Remote Method Invoke, 远程方法调用) 和内部类。

1998 年 12 月，JDK 1.2 版本发布，被命名为 Playground。该版本通常被称为 Java 2 版本，是见证重大转变的最流行版本。其主要特点是集合框架、JIT 编译器、策略工具、Java 基础类、Java 二维类库和对 JDBC 的改进。

2002 年 2 月，J2SE 1.4 版本发布，被命名为 Merlin。其主要特点是对 XML (Extensible Markup Language, 可扩展标记语言) 的处理、Java 打印、支持日志、JDBC 3.0、断言和对正则表达式的处理。

2004 年 9 月，J2SE 5.0 发布，被命名为 Tiger。其主要特点是支持泛型、自动装箱、注释处理和 Instrumentation。

2006 年 12 月，Java SE 6 版本发布，被命名为 Mustang。其主要特点是支持脚本语言、JDBC 4.0、Java 编译 API (Application Programming Interface, 应用程序编程接口) 并整合了 Web 服务。

2010 年 1 月，Oracle (甲骨文) 公司收购了 Sun 公司及其产品。现在 Java 由 Oracle 公司控制。

2011 年 7 月，Java SE 7 版本发布，被命名为 Dolphin。这个版本距上次发布有 5 年之久，并且只有这个版本花费了较长时间。其主要特点是支持动态语言、Java nio 包、多重异常处理、try with resource 功能和诸多小的增强。

2014 年 3 月，经过两年半的努力和屡次的延期，甲骨文的 Java 开发团队终于发布了 Java 8 的正式版本。它的最大改进就是 Lambda 表达式，其目的是使 Java 更易于为多核处理器编写代码；其次，新加入的 Nashorn 引擎也使得 Java 程序可以和 JavaScript 代码互操作；再者，新的日期时间 API、GC (Garbage Collection, 垃圾收集) 改进、并发改进也相当令人期待。

1.2 Java 的用处

Java 语言从技术上可以分为三个版本。

(1) J2SE——Java 2 Standard Edition (Java 标准版)：支持所有 Java 标准规范中所定义的核心类函数库和所有的 Java 基本类别。J2SE 定位在客户端程序的应用上。

(2) J2EE——Java 2 Enterprise Edition (Java 企业版)：在 J2SE 的基础上增加了企业内部扩展类函数库的支持，比如支持 Servlet/JSP 的 javax.servelet.* 和 EJB 的 javax.ejb.* 的类函数库。J2EE 定位在服务器端程序的应用上。

(3) J2ME——Java 2 Micro Edition (Java 的微型版)：只支持 Java 标准规范中所定义的核心类函数库的子集。J2ME 定位于嵌入式系统的应用上。

随着移动互联网时代的到来，手机应用程序的开发已经“热得发烫”。很多程序员都转向了 Java，使用 Java 语言进行 Android APP 的开发。这使得使用 Java 的程序员越来越多，图 1.2 为 2016 年 2 月排名前 20 位的编程语言排行榜。

2016年2月	2015年2月	变化	编程语言	得分率	涨幅
1	2	^	Java	21.145%	+5.80%
2	1	▼	c	15.594%	-0.89%
3	3		c++	6.907%	+0.29%
4	5	^	c#	4.400%	-1.34%
5	8	^	Python	4.180%	+1.30%
6	7	^	PHP	2.770%	-0.40%
7	9	^	Visual Basic.NET	2.454%	+0.43%
8	12	⤴	Perl	2.251%	+0.86%
9	6	▼	JavaScript	2.201%	-1.31%
10	11	^	Delphi/Object Pascal	2.163%	+0.59%
11	20	⤴	Ruby	2.053%	+1.18%
12	10	▼	Visual Basic	1.855%	+0.14%
13	26	⤴	Assembly language	1.828%	+1.08%
14	4	▼	Objectire-C	1.403%	-4.62%
15	30	⤴	D	1.391%	+0.77%
16	27	⤴	Swift	1.375%	+0.65%
17	18	⤴	R	1.192%	+0.23%
18	17	▼	MATLAB	1.091%	+0.06%
19	13	▼	PL/SQL	1.062%	-0.20%
20	33	⤴	Groovy	1.012%	+0.51%

图 1.2 2016 年 2 月编程语言 TOP 20 排行榜

1.3 Java 开发环境的安装和配置

JDK(Java Development Kit)是开发 Java 程序的核心工具包，当然，为了提高开发效率，也可以使用 IDE(集成开发环境)进行 Java 的开发。本书的全部代码都是在集成开发环境Eclipse下完成的。下面将对 Java 开发环境的安装和配置进行简单的介绍。

1.3.1 JDK+EditPlus

1. 下载 JDK

本书使用 Oracle 公司的 JDK 7，其下载网址为 <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>。因为本书使用的操作系统为 64 位的 Windows 7 系统，所以下载版本为 jdk-7u75-windows-x64.exe。如果不开发 Java 程序，只运行 Java 程序，只需下载并安装 JRE(Java Runtime Environment)即可。

图 1.3 为 JDK 的下载界面，在此界面中先点击“Accept License Agreement”，然后选择合适的 JDK 版本即可。

Java SE Development Kit 7u75

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	119.43 MB	jdk-7u75-linux-i586.rpm
Linux x86	136.77 MB	jdk-7u75-linux-i586.tar.gz
Linux x64	120.83 MB	jdk-7u75-linux-x64.rpm
Linux x64	135.66 MB	jdk-7u75-linux-x64.tar.gz
Mac OS X x64	185.86 MB	jdk-7u75-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.55 MB	jdk-7u75-solaris-i586.tar.Z
Solaris x86	95.87 MB	jdk-7u75-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.66 MB	jdk-7u75-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u75-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	138.66 MB	jdk-7u75-solaris-sparc.tar.Z
Solaris SPARC	98.56 MB	jdk-7u75-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u75-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.37 MB	jdk-7u75-solaris-sparcv9.tar.gz
Windows x86	127.8 MB	jdk-7u75-windows-i586.exe
Windows x64	129.52 MB	jdk-7u75-windows-x64.exe

图 1.3 JDK 的下载界面

2. 安装 JDK

双击下载后的 jdk-7u75-windows-x64.exe，会出现如图 1.4 所示的界面。

点击“下一步”按钮，出现如图 1.5 所示的 JDK 安装界面，在此界面中可以更改安装目录，也可以使用默认安装目录，更改完成后点击“下一步”按钮。

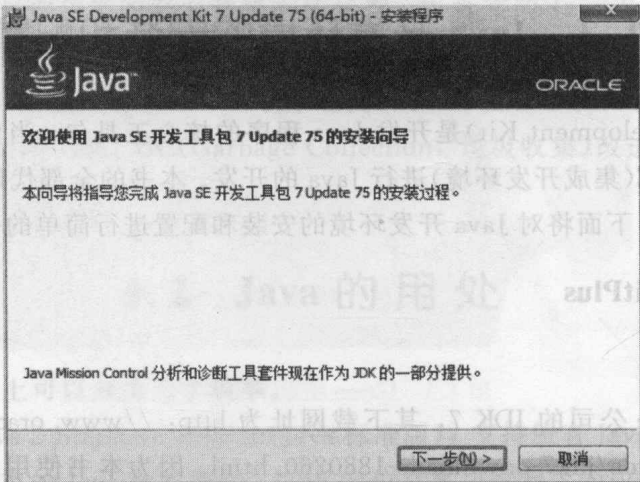


图 1.4 JDK 的安装界面一

安装完成后，出现如图 1.6 所示的界面，此时要求安装 JRE，可以选择点击“取消”按钮不进行安装。如果需要，则单击“下一步”按钮完成安装。

安装完成后(如果选择安装 JRE)，会在安装目录下出现两个文件夹，如图 1.7 所示。

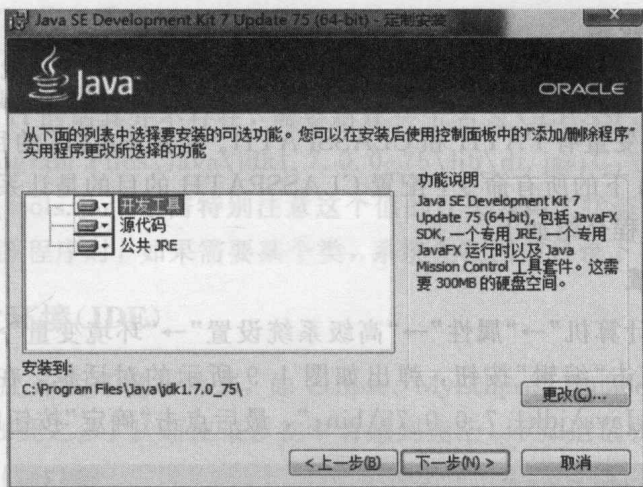


图 1.5 JDK 的安装界面二

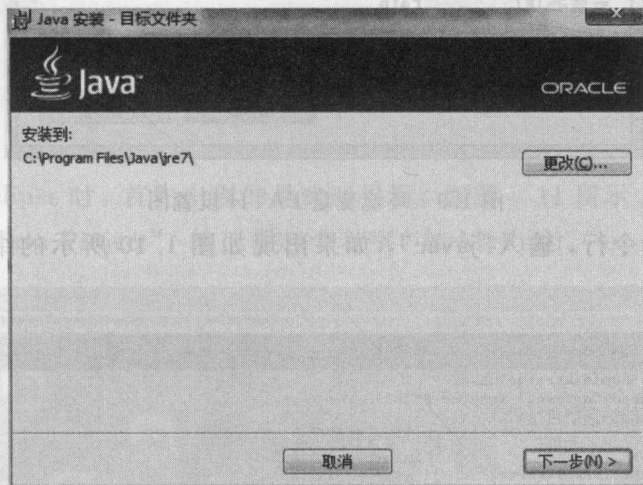


图 1.6 JDK 的安装界面三

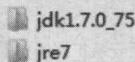


图 1.7 安装目录下的文件夹

3. 测试安装是否成功

安装完成后，要测试安装是否成功，可以在命令行中输入 `java-version` 命令，如图 1.8 所示，如果出现版本信息，即表示安装成功。

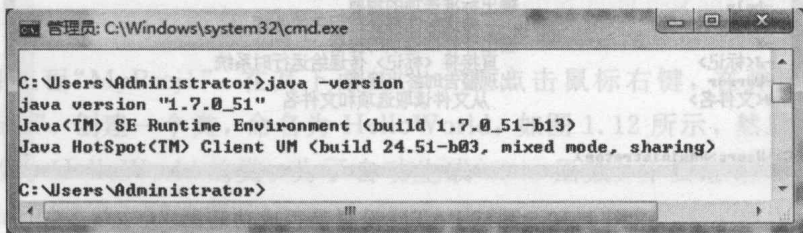


图 1.8 JDK 安装成功测试图



1.3.2 配置环境变量

JDK 安装成功后，为了能在任意的目录下编译、运行 Java 程序，必须为其配置环境变量。需要配置的环境变量有 PATH 和 CLASSPATH。配置 PATH 的目的是让系统能够找到 JDK 安装目录 bin 下的所有命令，配置 CLASSPATH 的目的是让系统能够找到 JDK 所提供的类库。配置过程介绍如下。

1. PATH 的配置

鼠标右键点击“计算机”→“属性”→“高级系统设置”→“环境变量”→“系统变量”。找到 PATH 变量，然后点击“编辑”按钮，弹出如图 1.9 所示的对话框，在原有值的前面加上“C:\Program Files\Java\jdk1.7.0.0_75\bin;”，最后点击“确定”按钮即可。

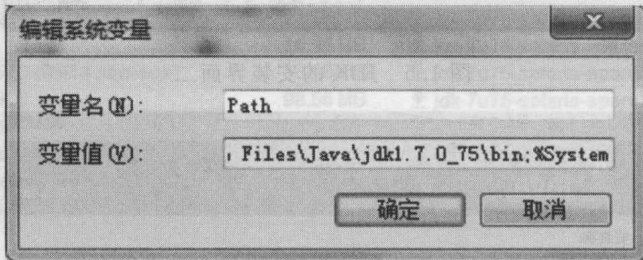


图 1.9 环境变量 PATH 设置图

现在可以打开命令行，输入“javac”，如果出现如图 1.10 所示的结果，就说明设置成功了。

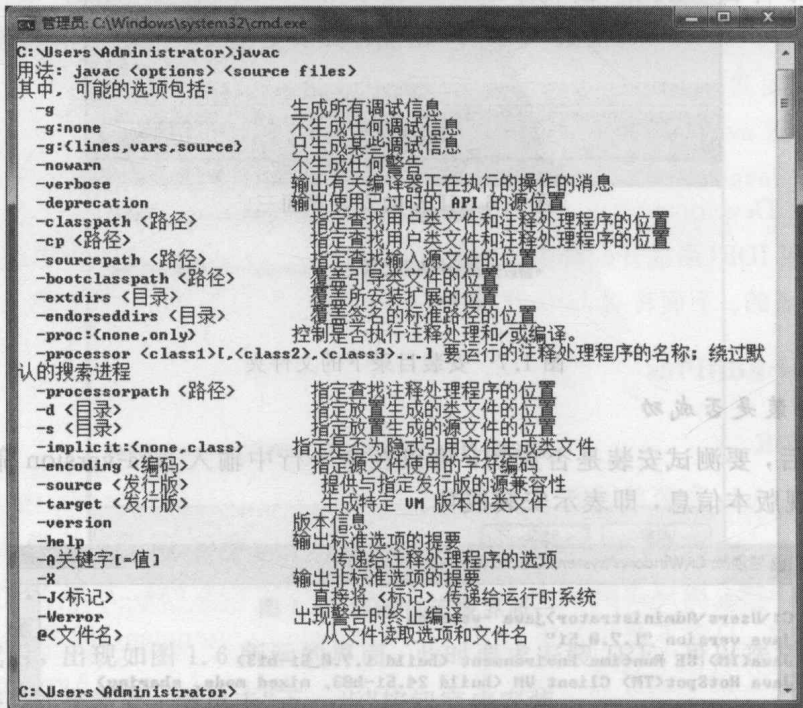


图 1.10 测试 PATH 设置是否成功截图

2. CLASSPATH 的配置

环境变量 CLASSPATH 的设置与 PATH 基本相同,找到“系统变量”后,如果是第一次设置 CLASSPATH 的话其并不存在,需要新建一个名为 CLASSPATH 的环境变量,其值设置为“.;C:\Program Files\Java\jdk1.7.0.0_75\lib\dt.jar;C:\Program Files\Java\jdk1.7.0.0_75\lib\tools.jar;”。需特别注意这个值的开头为英文状态下的“.”。表示当前目录,即当编译 Java 源程序时,如果需要某个类,系统先在当前目录下查找。

1.3.3 集成开发环境(IDE)

优秀的 Java 集成开发环境非常多,如 Eclipse、MyEclipse、JBuilder、JCreator、NetBeans 等。集成开发环境如此之多,从而使很多初学者感到迷茫,不知道选哪个才好。本书使用的是绿色版 Eclipse4.4.1,解压后即可使用,但使用前必须先安装并配置好 JDK。有些版本的 Eclipse 自带 JDK,但必须安装才能使用。关于以上列举的几种 IDE,读者可查找有关资料来比较它们的优劣。

1.4 第一个 Java 程序

第一次打开 Eclipse 时,首先出现的是欢迎界面,如图 1.11 所示。关掉欢迎界面,点击“File”→“New”→“Java Project”,给工程起个名字,如 MyProj1。



图 1.11 Eclipse 欢迎界面

鼠标点开工程“MyProj1”,在其下面的“src”上点击鼠标右键,在弹出的菜单中选择“New”→“Class”,创建一个类,命名为 HelloWorld,如图 1.12 所示,然后点击“Finish”按钮即可创建名为 HelloWorld 的类。为了自动生成 main 函数,不要忘记勾选“public static void main(String[] args)”选项。

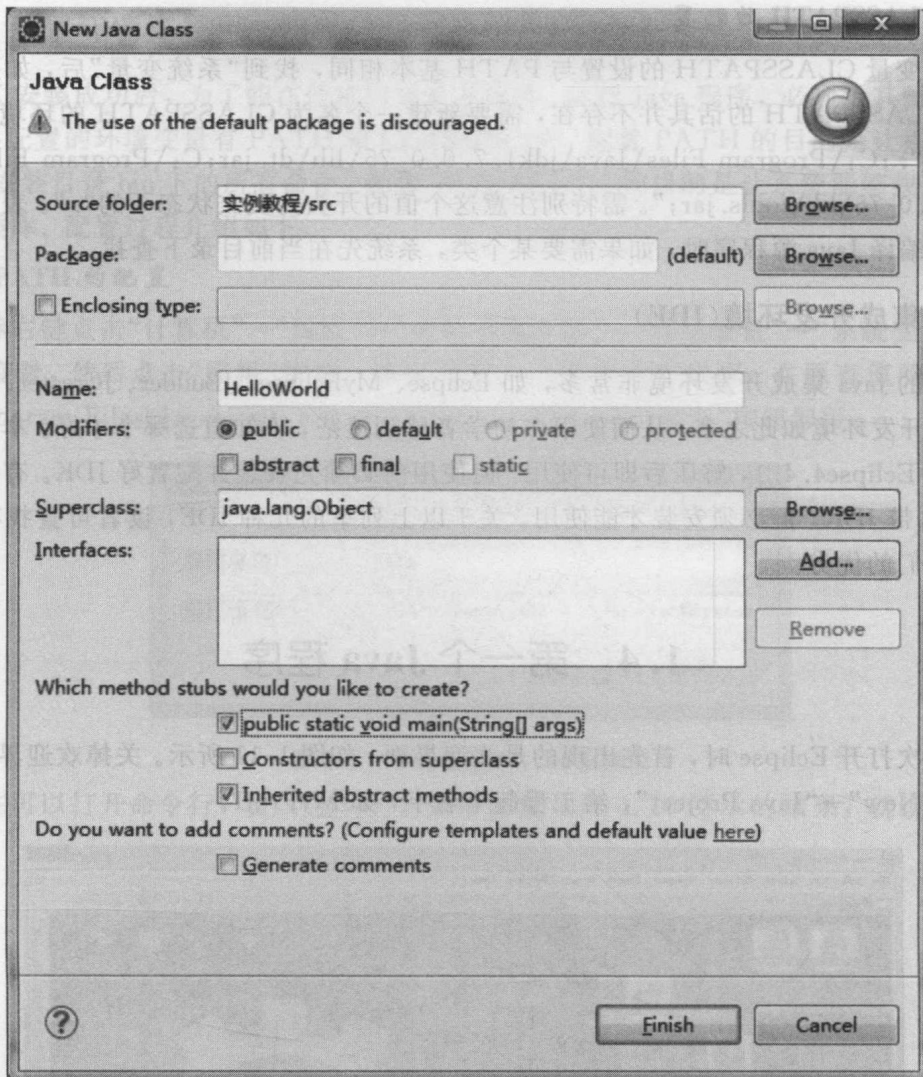


图 1.12 创建类示意图

Eclipse 自动生成代码如下：

```
//HelloWorld.java
public class HelloWorld {
    /*
     * @param args
     */
    public static void main(String[] args) {
        //TODO Auto-generated method stub
    }
}
```

程序中 `/* */` 和 `//` 后的部分为注释，其中 `/* */` 为注释块，`//` 为单行注释。我们可以在 `main` 函数中添加下面的输出语句，运行后可使其输出“HelloWorld!”。

```
System.out.println("HelloWorld!");
```