



高职高专教育“十一五”规划教材

Java 语言程序设计案例教程

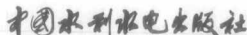
主 编 李明才

副主编 何永太 盛安元 丁 俊



中国水利水电出版社
www.waterpub.com.cn

高职高专教育“十一五”规划教材



内 容 提 要

本书全面、系统地介绍了 Java 语言及其程序设计。全书共 11 章, 主要内容包括: Java 语言概论、Java 语言基本语法、Java 语言的面向对象特性、图形用户界面设计、异常处理、多线程、输入输出及文件处理、Java 与数据库连接、Java 网络编程、初识 JSP、综合案例。

本书内容编排力求做到通俗易懂、重视概念、强化实践, 并采用案例教学, 帮助读者从大量的案例讲解中掌握 Java 语言的基础知识, 达到循序渐进、逐步深入、反复实践、牢固掌握的目的。

本书可供高职高专计算机及相关专业师生使用, 也可作为 Java 语言的培训教材或 Java 语言爱好者的自学用书。

本书配有电子教案和素材文件, 读者可以从中国水利水电出版社网站和万水书苑免费下载, 网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目 (C I P) 数据

Java语言程序设计案例教程 / 李明才主编. — 北京:
中国水利水电出版社, 2010.7
高职高专教育“十一五”规划教材
ISBN 978-7-5084-7645-2

I. ①J… II. ①李… III. ①
JAVA语言—程序设计—高等学校: 技术学校—教材 IV.
①TP312

中国版本图书馆CIP数据核字(2010)第121412号

策划编辑: 雷顺加 责任编辑: 李 炎 加工编辑: 李 刚 封面设计: 李 佳

书 名	高职高专教育“十一五”规划教材 Java 语言程序设计案例教程
作 者	主 编 李明才 副主编 何永太 盛安元 丁 俊
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 14.5 印张 418 千字
版 次	2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷
印 数	0001—4000 册
定 价	25.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

“前—十”言

会委编

Java 语言是由 Sun 公司开发的、近年来在国内外得到广泛应用的一种计算机程序设计语言。它功能丰富，表达简洁，使用方便灵活，应用面广，目标程序效率高，可移植性好，是一种具有良好程序设计风格的应用程序设计语言。现在，在许多高校 Java 语言已不仅成为计算机及其相关专业的必修课，而且在很多非计算机专业也开设了 Java 语言课程，并且 Java 语言也列入了全国计算机等级考试、全国计算机应用技术证书考试（NIT）等的考试范围。

本书力求通俗易懂，重视概念，强化实践，采用案例教学。使读者能从大量的案例讲解中掌握 Java 语言的基础知识，达到循序渐进，逐步深入，反复实践，牢固掌握的目的。

本书共分 11 章，第 1 章介绍 Java 语言概论，第 2 章介绍 Java 语言基本语法，第 3 章介绍 Java 语言的面向对象特性，第 4 章介绍图形用户界面设计，第 5 章介绍异常处理，第 6 章介绍多线程，第 7 章介绍输入输出及文件处理，第 8 章介绍 Java 与数据库连接，第 9 章介绍 Java 网络编程，第 10 章介绍 JSP 基础知识，第 11 章是一个综合案例，通过该案例对全书内容进行总结。

本书由李明才任主编，何永太、盛安元、丁俊任副主编，何永太编写第 1 章和第 7 章，徐伟编写第 2 章，李明才编写第 3 章和第 5 章，盛安元编写第 4 章和第 6 章，陈建锋编写第 8 章和第 9 章，丁俊编写第 10 章和第 11 章，全书由李明才统稿。

由于作者水平有限，书中一定会有不少缺点和错误，敬请读者批评指正。

作者

2010 年 5 月

目 录

录

前言	1.1.2
第1章 Java 语言概论	1
1.1 Java 语言的发展过程	1
1.2 Java 语言的工作原理及特点	1
1.2.1 Java 程序的处理过程	2
1.2.2 Java 字节码	2
1.2.3 Java 虚拟机 (JVM)	2
1.2.4 垃圾回收	3
1.3 Java SDK 的安装与配置	3
1.3.1 下载并安装 Java SDK 开发工具	3
1.3.2 JDK 的配置与测试	5
1.4 Eclipse 的安装与使用	6
1.4.1 Eclipse 的安装与启动	6
1.4.2 Eclipse 的使用	7
1.5 Java 程序举例	11
1.5.1 Java Application 举例	11
1.5.2 Java Applet 举例	12
1.5.3 Java 程序结构	12
本章小结	13
实训一	13
第2章 Java 语言基本语法	14
2.1 Java 语言的数据类型	14
2.1.1 Java 语言基本字汇	14
2.1.2 简单数据类型	15
2.1.3 常量 (字面量)	16
2.1.4 变量	18
2.1.5 类型转换	20
2.2 Java 语言的运算符和表达式	21
2.2.1 算术运算符	21
2.2.2 位运算符	24
2.2.3 关系运算符	27
2.2.4 逻辑运算符	28
2.2.5 赋值运算符	29
2.2.6 条件运算符 (?:)	29

2.2.7 运算符优先级	30
2.2.8 使用圆括号	31
2.3 Java 语言的流程控制语句	31
2.3.1 Java 的选择语句	31
2.3.2 循环语句	35
2.3.3 跳转语句	40
2.4 数组	42
2.4.1 一维数组	42
2.4.2 多维数组	44
2.4.3 另一种数组声明语法	45
2.5 字符串	45
2.5.1 String 类	45
2.5.2 字符串连接	46
2.5.3 字符串与其他类型数据的连接	46
本章小结	47
实训二	47
第3章 Java 语言的面向对象特性	48
3.1 类与对象的概念	48
3.1.1 面向对象的特性	48
3.1.2 类的定义	48
3.1.3 创建对象与定义构造方法	49
3.1.4 修饰符	52
3.2 类的继承	57
3.2.1 继承的概念	58
3.2.2 继承的实现	58
3.3 类的多态	61
3.3.1 多态的概念	61
3.3.2 方法覆盖实现多态	61
3.3.3 方法重载实现多态	62
3.3.4 构造方法的继承与重载	62
3.4 包与接口	63
3.4.1 包	63
3.4.2 接口	65

本章小结	66
实训三	67
第4章 图形用户界面设计	68
4.1 图形用户界面概述	68
4.2 Applet 类	70
4.2.1 Applet 类	70
4.2.2 建立 Applet 小应用程序	70
4.2.3 Applet 的生命周期	72
4.2.4 Applet 类常用方法	74
4.3 HTML 文件参数传递	75
4.4 Java 绘图、字体及颜色处理	77
4.4.1 Java 绘图	77
4.4.2 Java 字体类	83
4.4.3 画笔颜色	84
4.5 框架与面板	85
4.5.1 框架 (Frame)	85
4.5.2 面板 (Panel)	86
4.6 按钮、标签、文本框与文本区域	87
4.6.1 按钮 (Button)	87
4.6.2 标签 (Label)	88
4.6.3 文本框 (TextField) 与文本区域 (TextArea)	89
4.7 事件处理机制	91
4.7.1 Java 中的事件体系结构	91
4.7.2 Component 类	93
4.7.3 代理事件模型	95
4.8 布局设计	96
4.8.1 FlowLayout 布局管理器	97
4.8.2 BorderLayout 布局管理器	98
4.8.3 GridLayout 布局管理器	100
4.8.4 CardLayout 布局管理器	101
4.8.5 GridBagLayout 布局管理器	103
4.9 菜单设计	106
4.10 对话框设计	109
4.11 常用 Swing 组件简介	111
4.11.1 Swing 组件	111
4.11.2 Swing 组件体系结构	112
4.11.3 可插接的外观和感觉	114
4.11.4 常用 Swing 组件	114

4.11.5 对话框 JOptionPane	116
本章小结	116
实训四	117
第5章 异常处理	118
5.1 异常类	118
5.1.1 异常的基本概念	118
5.1.2 异常类	118
5.1.3 系统定义的运行异常	119
5.1.4 用户自定义的异常	120
5.2 异常抛出	120
5.2.1 系统自动抛出的异常	120
5.2.2 语句抛出异常	121
5.3 异常处理	121
5.3.1 try-catch 语句	121
5.3.2 多异常处理	122
5.3.3 finally 子句	123
本章小结	124
实训五	124
第6章 多线程	125
6.1 多线程的基本概念	125
6.2 多线程的创建	127
6.2.1 创建线程	128
6.2.2 线程的启动	128
6.3 多线程的实现	130
6.3.1 线程的状态	130
6.3.2 线程的调度	131
6.3.3 多线程的互斥与同步	135
6.3.4 生产者消费者模型	139
6.3.5 死锁	142
本章小结	144
实训六	144
第7章 输入输出及文件处理	146
7.1 输入/输出流概念	146
7.2 输入/输出流类	146
7.2.1 字节流 InputStream 和 OutputStream 类	147
7.2.2 字符流 Reader 和 Writer 类	148
7.3 标准输入/输出	149
7.3.1 标准输入流	150

7.3.2 标准输出流	150	10.4.1 page 指令	191
7.3.3 标准错误输出流	150	10.4.2 include 指令	193
7.4 常用的文件处理	150	10.5 JSP 常见内置对象	194
7.4.1 文件的顺序读写	150	10.5.1 request	194
7.4.2 文件的随机读写	156	10.5.2 response	194
7.4.3 目录和文件管理	158	10.5.3 out	195
本章小结	160	10.5.4 cookie	195
实训七	160	10.5.5 session	196
第 8 章 Java 与数据库连接	161	10.5.6 application	196
8.1 JDBC 简介	161	本章小结	197
8.1.1 JDBC 概述	161	实训十	197
8.1.2 JDBC 与 ODBC 的比较	161	第 11 章 综合案例	206
8.1.3 两层模型与三层模型	162	11.1 系统模块设计	206
8.1.4 JDBC 驱动程序种类	162	11.2 数据库结构设计	207
8.2 JDBC 访问数据库	163	11.2.1 学生表结构	207
8.2.1 JDBC 的常用接口和类	163	11.2.2 课程表结构	207
8.2.2 T-SQL 常用基本语法	164	11.2.3 成绩表结构	208
8.3 JDBC 应用	165	11.2.4 操作员表结构	208
8.3.1 JDBC 应用步骤	165	11.3 详细设计	208
8.3.2 JDBC 应用实例	166	11.3.1 主窗口设计	208
本章小结	169	11.3.2 登录界面设计	209
实训八	169	11.3.3 系统管理模块设计	209
第 9 章 Java 网络编程	171	11.3.4 其他三个管理模块	211
9.1 网络基础知识	171	11.3.5 数据库操作模块	211
9.2 网络通信中 Java 的网络类和接口	171	11.4 具体实现	212
9.2.1 URL 和 URLConnection	172	11.4.1 主窗口	212
9.2.2 Socket (套接字)	173	11.4.2 数据库操作模块	212
本章小结	176	11.4.3 操作员登录模块	217
实训九	177	11.4.4 操作员添加模块	218
第 10 章 初识 JSP	181	11.4.5 操作员浏览模块	218
10.1 JSP 简介	181	11.4.6 操作员查询模块	220
10.1.1 JSP 概述	181	11.4.7 操作员删除模块	222
10.1.2 JSP 开发环境	182	11.4.8 其他模块的实现	223
10.2 编写一个 JSP 页面	188	本章小结	223
10.3 JSP 的执行过程	191	参考文献	224
10.4 JSP 指令简介	191		

第1章 Java 语言概论



本章导读

本章简单介绍 Java 语言的发展过程,讲授 Java 程序的组成结构和运行机理、Java SDK 开发平台和使用 Eclipse 开发工具编写 Java 程序的方法。



主要知识点

- Java 语言的发展历史
- Java 的工作原理及特点
- 安装并设置 Java 开发平台
- Java 程序的组成结构
- 使用 Eclipse 集成开发环境编写运行 Java 程序的方法

1.1 Java 语言的发展过程

Java 语言最早诞生于 1991 年,刚开始它只是 Sun 公司为一些消费性电子产品所设计的通用环境。因为当时 Java 的应用对象只限于 PDA、电子游戏机、电视机顶盒之类的消费性电子产品,所以并未被众多的编程技术人员所接受。

在 Java 出现以前,Internet 上的信息内容都是一些静态的 HTML 文档。正是在 Web 中看不到交互式的内容,所以人们很不满意当时的 Web 浏览器,他们迫切希望能够在 Web 上创建一类无须考虑软硬件平台就可以执行的应用程序,并且这些程序还要有极大的安全保障。正是由于这种需求给 Java 带来了前所未有的施展舞台。

Sun 公司的工程师从 1994 年起把 Java 技术应用于 Web 上,并且开发出了 HotJava 的第一个版本。于是,Java 的名字逐渐变得广为人知。

Java 在 Sun World 95 中被正式发布。Java 的“Write Once, Run Anywhere (一次编写,到处运行)”口号使得 Java 一出现就引起了广泛的注意,用 Java 技术开发的软件可以不用修改或重新编译就可直接应用于任何计算机上。

从此以后,Java 随着网络的快速发展,而成为一个应用最广泛的程序语言。“网络即是计算机”是 Sun 公司的格言。一时间,使用 Java 技术进行软件开发成为广大技术人员的一种时尚。

到 2010 年 4 月为止,Java 已经发布了一系列的版本,目前最新版本为 JDK 7.0 (1.7.0) 预览版。

1.2 Java 语言的工作原理及特点

Java 是一种高级的、通用的、面向对象的、适用于网络环境的程序设计语言,同时它又是一

种操作平台,为程序的运行提供一个统一通用的环境,屏蔽底层的操作系统及硬件环境的差异性。

1.2.1 Java 程序的处理过程

一个 Java 程序的执行必须经过编辑、编译、运行三个步骤。

编辑是指在某个 Java 语言开发环境中进行程序代码的输入与编写,最终形成后缀名为.java 的 Java 源文件。

编译是指使用 Java 编译器 (javac 命令) 把源文件翻译成二进制代码的过程,这期间也进行语法级别错误和引用错误的排查,编译后将生成后缀名为.class 的字节码文件。该字节码文件并不是一个可以直接运行的文件。

运行是指使用 Java 解释器将字节码文件翻译成机器代码,执行并得到运行结果。这一过程如图 1.1 所示。



图 1.1 Java 程序运行流程

1.2.2 Java 字节码

字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码,它是一种二进制文件,是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件,它必须由专用的 Java 解释器来解释执行,因此 Java 是一种在编译基础上进行解释运行的语言。

1.2.3 Java 虚拟机 (JVM)

Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此 Java 程序不能直接运行在现有的操作系统平台上,它必须运行在被称为 Java 虚拟机的软件平台之上。

Java 虚拟机 (JVM) 是驻留于计算机内存的虚拟计算机或逻辑计算机,实际上是一段负责解释并执行 Java 字节码的程序。Java 解释器只是 Java 虚拟机的一部分。

在运行 Java 程序时,首先会启动 JVM,然后由它来负责解释执行 Java 的字节码,并且 Java 字节码只能运行于 JVM 之上。这样利用 JVM 就可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来,只要在不同的计算机上安装了针对于特定具体平台的 JVM,Java 程序就可以运行,而不用考虑当前具体的硬件平台及操作系统环境,也不用考虑字节码文件是在何种平台上生成的。JVM 把这种不同软硬件平台的具体差别隐藏起来,从而实现了真正的二进制代码级的跨平台移植。JVM 是 Java 平台无关的基础,Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。Java 的这种运行机制可以通过图 1.2 说明。

Java 语言这种 “Write Once, Run Anywhere” 的方式,有效地解决了目前大多数高级程序设计语言需要针对不同系统来编译产生不同机器代码的问题,即硬件环境和操作平台的异构问题,大大降低了程序开发、维护 and 管理的开销。

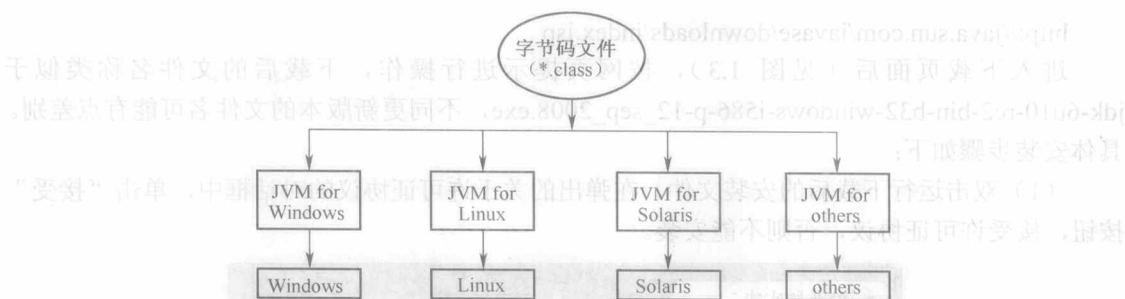


图 1.2 JVM 工作方式

需要注意的是, Java 程序通过 JVM 可以达到跨平台特性, 但 JVM 是不跨平台的。也就是说, 不同操作系统之上的 JVM 是不同的, Windows 平台之上的 JVM 不能用在 Linux 上面, 反之亦然。

1.2.4 垃圾回收

Java 虚拟机使用两个独立的堆内存, 分别用于静态内存分配和动态内存分配。其中一个是非垃圾回收堆内存, 用于存储所有类的定义、常量池和方法表。另一个堆内存再分为两个可以根据要求往不同方向扩展的小块。用于垃圾回收的算法适用于存放在动态堆内存中的对象。垃圾回收器将在回收对象实例之前调用 `finalize()` 方法。即使显示调用垃圾回收方法 (`System.gc()`), 也不能保证立即运行, 这是因为垃圾回收线程的运行优先级很低, 可能经常会被中断。

1.3 Java SDK 的安装与配置

若要编写 Java 程序, 就需要开发工具。现在可用于开发 Java 程序的工具很多, Java SDK 是 Sun 公司 (现被 Oracle 公司收购) 提供的免费开发工具集。

Java SDK 的意思是 Java Software Development Kit, 即 Java 软件开发工具包, 也称为 JDK。截止 2010 年 4 月, 提供下载的 Java SDK 标准版软件最新版本为 1.6 Update 20 (JDK 1.7 预览版也已发布), 有不同操作系统的不同版本。下面介绍在 Windows XP 操作系统下安装和配置 Java SDK 的过程。

1.3.1 下载并安装 Java SDK 开发工具

Java SDK 目前有以下几个版本:

Java SE (Java Platform, Standard Edition): 即通常所说的 Java 平台标准版, 提供基础 Java 开发工具、执行环境与 API (Application Program Interface)。

Java ME (Java Platform, Micro Edition): Java 平台微型版, 适用于消费性电子产品, 提供嵌入式系统所使用的 Java 开发工具、执行环境与 API。

Java EE (Java Platform, Enterprise Edition): Java 平台企业版, 它是由 Sun 公司所提出的一组技术规格, 规划企业用户以 Java 技术开发、分发、管理多层次应用结构。

作为学习 Java 语言和一般的应用开发, 使用 Java SE 版本足够了, 本书采用的是 Java SDK 1.6 版本。可以从 Sun 公司网站上下载, 下载网址是:

<http://java.sun.com/javase/downloads/index.jsp>

进入下载页面后（见图 1.3），按网页提示进行操作，下载后的文件名称类似于 jdk-6u10-rc2-bin-b32-windows-i586-p-12_sep_2008.exe，不同更新版本的文件名可能有点差别。具体安装步骤如下：

（1）双击运行下载后的安装文件。在弹出的关于许可证协议的对话框中，单击“接受”按钮，接受许可证协议，否则不能安装。



图 1.3 JDK 下载地址

（2）在弹出的“自定义安装”对话框中，选择 JDK 的安装路径。单击“更改”按钮可更改安装路径。为加快安装速度，Java DB、公共 JRE、源代码等选项可不安装，但开发工具选项是必需的，如图 1.4 所示。

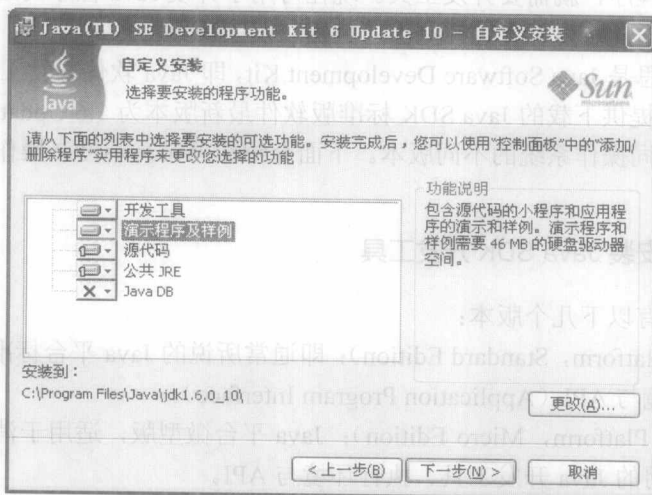


图 1.4 选择 JDK 安装路径

（3）单击图 1.4 中的“下一步”按钮，开始安装。如果在图 1.4 中选择了安装公共 JRE，则在安装的过程中还会弹出另一个“自定义安装”对话框，提示用户选择 Java 运行时环境的安装路径，其操作过程与图 1.4 所示界面类似。

(4) 最后单击“完成”按钮完成 JDK 的安装。

在默认情况下,安装后会在 C 盘“Program Files”文件夹下产生如图 1.5 所示的文件夹结构,其中:

“bin”文件夹下包含一些开发工具,这些开发工具能够帮助开发、执行、调试以及文档化 Java 程序。

“jre”文件夹下包含 Java 虚拟机、类库和其他支持 Java 程序运行的文件。

“lib”文件夹下包含开发工具所需要的附加类库和支持文件。

“demo”文件夹下带有 Java 源文件的例子,这些例子包括使用 Swing、Java 的基础类和 Java 平台调试结构的例子。

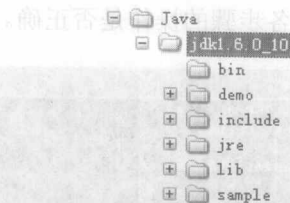


图 1.5 Java SDK 安装文件夹结构

1.3.2 JDK 的配置与测试

JDK 安装完成后,需要设置环境变量及测试 JDK 配置是否成功,具体步骤如下:

(1) 在 Windows 系统桌面上右键单击“我的电脑”图标,选择“属性”菜单项。在弹出的“系统属性”对话框中选择“高级”选项卡,然后单击“环境变量”按钮,将弹出“环境变量”对话框。

(2) 在“环境变量”对话框中,单击“系统变量”区域中的“新建”按钮,将弹出“新建系统变量”对话框。

(3) 在“新建系统变量”对话框中,在“变量名”文本框中输入“JAVA_HOME”,在“变量值”文本框中输入 JDK 的安装路径“C:\Program Files\Java\jdk1.6.0_10”,如图 1.6 所示。最后单击“确定”按钮,完成变量 JAVA_HOME 的创建。

(4) 查看是否存在 Path 变量,若存在,则编辑该变量,在变量值后追加“;%JAVA_HOME%\bin”,如图 1.7 所示;若不存在,则新建该变量,并设置变量值为“%JAVA_HOME%\bin”。

(5) 查看是否存在 Classpath 变量,若存在,则在变量值后加入:“.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar”;若不存在,则创建该变量,并设置同上的变量值。



图 1.6 创建 JAVA_HOME 变量

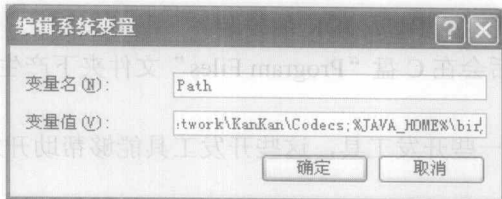


图 1.7 编辑 Path 变量

(6) 测试 JDK 的安装和配置是否成功。依次单击“开始”按钮，单击“运行”菜单项，在弹出的“运行”对话框中输入“cmd”命令，进入命令提示符窗口。进入任意目录下输入“javac”命令，按 Enter 键执行该命令，系统会输出 javac 命令的使用帮助信息，如图 1.8 所示。这说明 JDK 安装配置成功，否则需要检查上面各步骤的操作是否正确。

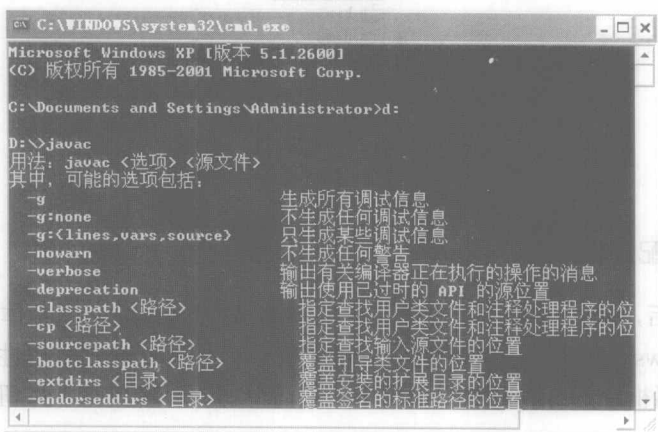


图 1.8 显示 javac 命令的使用帮助

1.4 Eclipse 的安装与使用

Eclipse 是一个基于 Java 的、开放源码的、可扩展的应用开发平台，它为编程人员提供了一流的 Java 集成开发环境（Integrated Development Environment, IDE）。它是一个可以用于构建集成 Web 和应用程序开发工具的平台。

1.4.1 Eclipse 的安装与启动

在 Eclipse 的官方网站 <http://www.eclipse.org> 可下载 Eclipse 的最新版本，下载后的文件一般为一个 zip 格式的压缩文件。

(1) 将下载后的压缩文件解压后，双击 eclipse.exe 文件就可启动 Eclipse。

(2) 解压完成后，启动的 Eclipse 是英文版的，可以通过安装 Eclipse 的多国语言包，实现 Eclipse 的本地化。多国语言包可从 Eclipse 官方网站进行下载。具体的 Eclipse 汉化步骤请参看相关文献。

(3) 每次启动 Eclipse 时，都需要设置工作空间，工作空间用来存放创建的项目，如图 1.9 所示。单击“Browse”按钮可选择一个存在的文件夹。可通过勾选“Use this as the default and do not ask again”选项屏蔽该对话框。

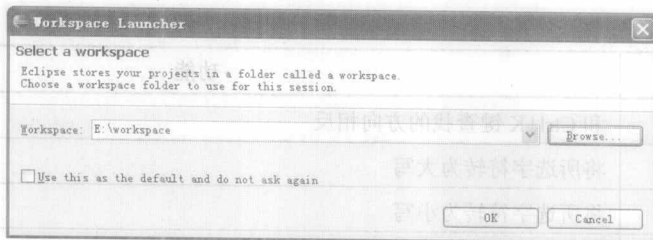


图 1.9 启动 Eclipse 时设置工作空间

(4) 最后单击“OK”按钮，若是初次进入，在步骤(3)中选择工作空间，则出现 Eclipse 欢迎界面，如图 1.10 所示。

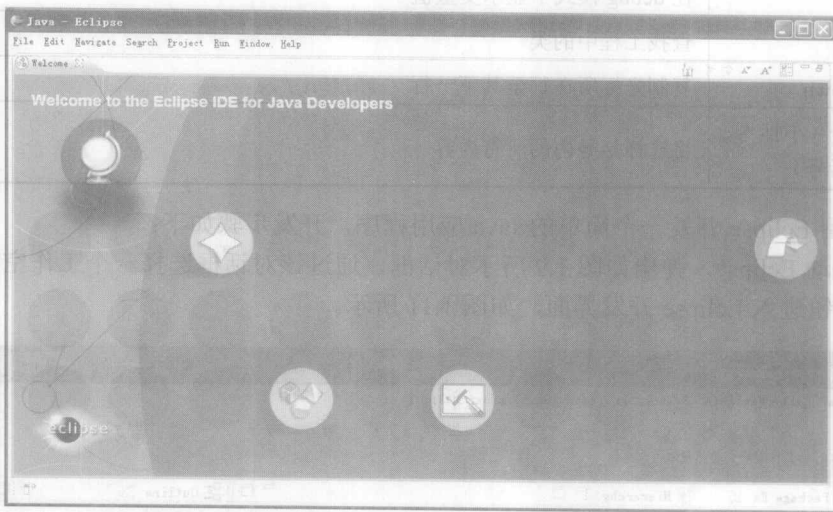


图 1.10 Eclipse 的欢迎界面

1.4.2 Eclipse 的使用

Eclipse 3.2 开发工具的常用快捷键如表 1.1 所示，熟悉这些快捷键可大幅提高工具使用效率。

表 1.1 Eclipse 3.2 开发工具的常用快捷键

快捷键	功能
F3	跳转到类或变量的声明处
Alt+/	代码提示
Alt+上下方向键	将选中的一行或多行向上或向下移动
Alt+左右方向键	跳到前一次或后一次的编辑位置，在代码跟踪时用得比较多
Ctrl+/	注释或取消注释
Ctrl+D	删除光标所在行的代码
Ctrl+K	将光标停留在变量上，按该快捷键可查找下一个同样的变量
Ctrl+Q	回到最后编辑的位置

续表

快捷键	功能
Ctrl+Shift+K	和 Ctrl+K 键查找的方向相反
Ctrl+Shift+X	将所选字符转为大写
Ctrl+Shift+Y	将所选字符转为小写
Ctrl+Shift+/	注释代码块
Ctrl+Shift+\	取消注释代码块
Ctrl+Shift+M	导入未引用的包
Ctrl+Shift+D	在 debug 模式中显示变量值
Ctrl+Shift+T	查找工程中的类
Ctrl+Shift+Down	复制光标所在行至其下一行
鼠标双击括号（小括号、中括号、大括号）	将选择括号内的所有内容

下面应用 Eclipse 开发一个简单的 Java 应用程序，开发步骤如下：

(1) 启动 Eclipse，弹出如图 1.9 所示对话框，通过该对话框选择一个工作空间，然后单击“OK”按钮进入 Eclipse 开发界面，如图 1.11 所示。

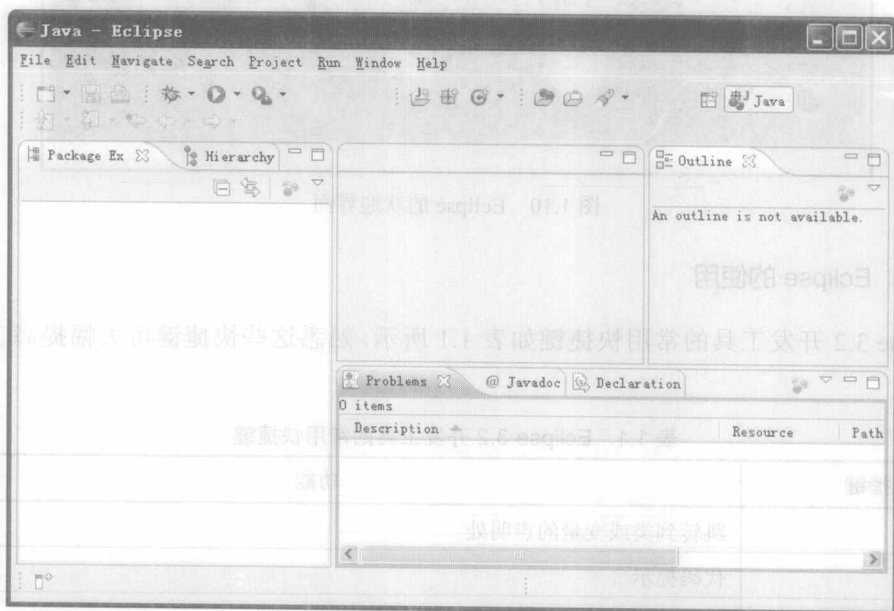


图 1.11 Eclipse 开发界面

(2) 依次单击菜单栏中的“File”/“New”/“Project”菜单项，将弹出“New Project”对话框，如图 1.12 所示。

(3) 在图 1.12 所示对话框中选择“Java Project”选项，并单击“Next”按钮。

(4) 在弹出的“New Java Project”对话框中，在“Project Name”文本框中输入项目名称，本例中项目名称为“MyJava”。其他保留默认配置，如图 1.13 所示。

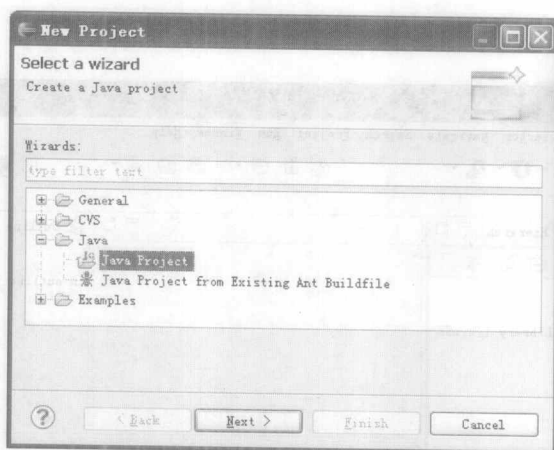


图 1.12 新建项目对话框

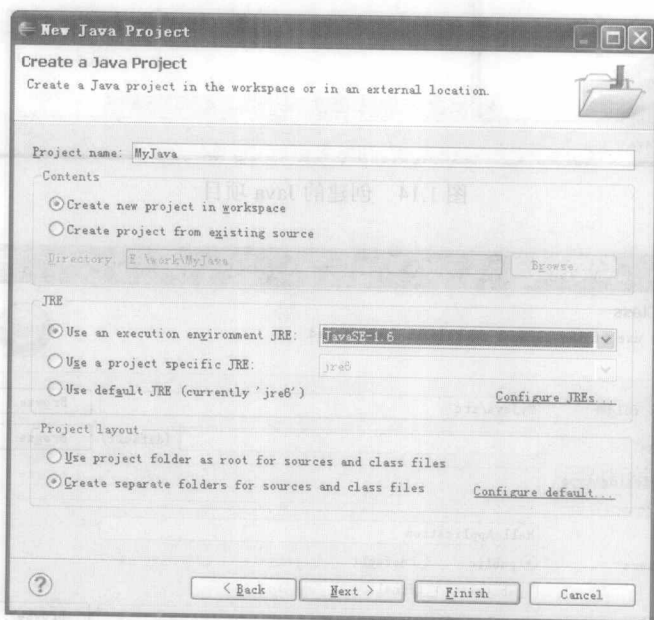


图 1.13 配置项目对话框

(5) 单击“Finish”按钮，完成 Java 项目的创建，创建的项目会在 Eclipse 界面的左侧栏中显示，如图 1.14 所示。

(6) 添加一个 Java Class。在图 1.14 的左侧“Package”选项卡中右键单击项目名“MyJava”，在弹出的快捷菜单中依次选择“New”/“Class”，弹出如图 1.15 所示的“New Java Class”对话框。

(7) 在图 1.15 中，在“Name”栏中输入要添加的类的名称，本例中输入“HelloApplication”。由于本例只包含一个主类，因此在下面勾选“public static void main(String[] args)”选项，让 Eclipse 自动在类中创建 main 方法；若添加的不是主类，则无需勾选此选项。其他保持不变，单击“Finish”按钮完成类文件的添加。Eclipse 会自动打开刚才添加的类文件，并且自动生成类定义和 main 方法定义的代码框架。在 main 方法中输入例 1.1 的 main 方法中的代码。最后

界面如图 1.16 所示。

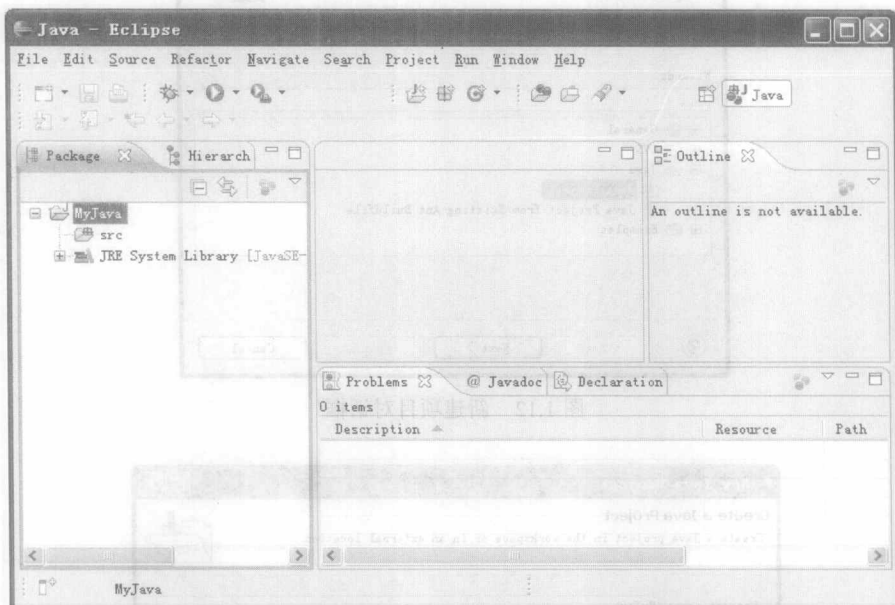


图 1.14 创建的 Java 项目

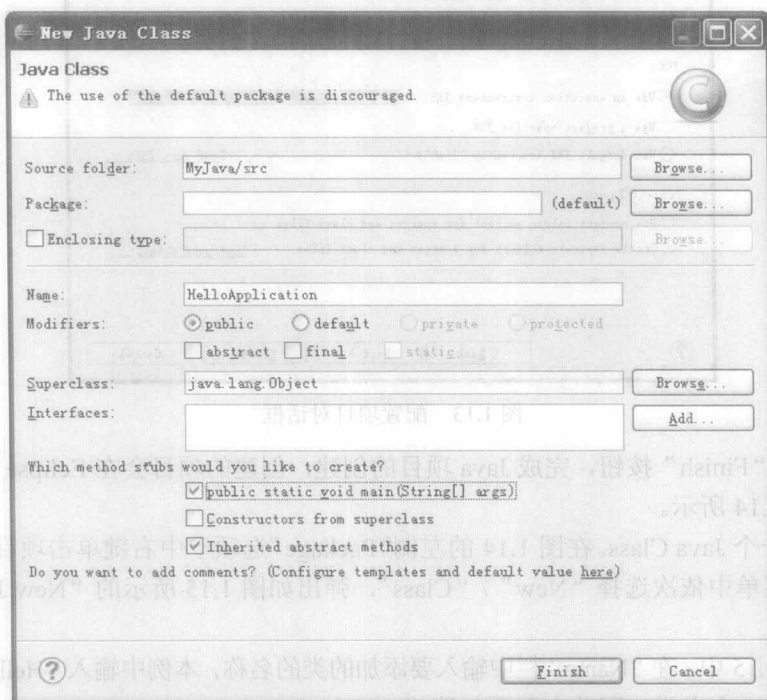


图 1.15 添加类对话框

步骤(8) 运行程序。单击工具栏中的  按钮, Eclipse 会自动完成程序的编译和运行, 本例程序运行后会在 Eclipse 下方的 Console 窗格中显示“Hello World!”的信息。