



普通高等教育“十二五”规划教材

Java程序设计

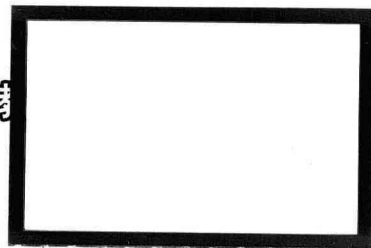
郑珂 刘克成 主 编



中国电力出版社
CHINA ELECTRIC POWER PRESS



普通高等教育“十二五”规划教材



Java程序设计

主 编 郑 珂 刘 克 成
主 审 王 水



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书为普通高等教育“十二五”规划教材。本书从面向对象程序设计的基本概念入手,介绍 Java 程序设计,旨在帮助读者建立面向对象的思想、掌握面向对象程序设计的基本方法、原则和技能。全书立足基本概念和方法,注重实践与应用。对基本概念、理论和技术的介绍力求做到简洁、清晰、严谨、易懂;对例题代码力求精简、规范。全书提供了大量课后练习题以夯实基础,并提供了四个综合应用实例以加强读者对 Java 技术的认识和理解。

本书可作为高等本科院校 Java 程序设计课程的教材,也可作为高职高专相关课程的教材,还可供软件开发和电信工程技术人员学习参考。

图书在版编目(CIP)数据

Java 程序设计 / 郑珂, 刘克成主编. —北京: 中国电力出版社, 2012.6

普通高等教育“十二五”规划教材

ISBN 978-7-5123-3151-8

I. ①J… II. ①郑… ②刘… 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 (2012) 第 123495 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

航远印刷有限公司印刷

各地新华书店经售

*

2012 年 7 月第一版 2012 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 22.5 印张 546 千字

定价 40.00 元

敬告读者

本书封面贴有防伪标签,刮开涂层可查询真伪

本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究



前 言

Java 是一种完全面向对象的程序设计语言, 适合于进行面向对象的软件系统的开发。目前基于 Java 语言产生的 Java 体系是世界上很多开发人员用于开发企业应用的首选。Java 提供三方面的应用, 用于编写工作站的 J2SE、可扩展的企业级应用的 Java 2 平台 J2EE, 以及用于开发嵌入式 Java 2 电子消费平台的 J2ME。Java 已经不仅仅是一种编程语言, 更是一种技术, 一个开发平台。

全书共有 13 章, 第 1 章介绍 Java 语言的历史和特点, 介绍了 Java 虚拟机和 Java 平台, Java 的开发工具和开发步骤。第 2 章介绍了编写 Java 程序的基本要素, 包括 Java 语言的标识符、关键字、数据类型、表达式、运算符、数据类型的转换、变量和变量的作用域。第 3、4 章主要讲解面向对象程序设计的编程概念和方法, 如何在 Java 语言中实现这些概念和方法, 以及面向对象程序设计的基本原则。第 5 章介绍了 Java 的异常处理, 包括异常的产生、异常的捕获、异常的声明抛出和自定义异常等。第 6 章主要讲解 Java 图形用户界面的设计, 包括使用 Swing 组件, 容器的布局管理和事件处理。第 7 章主要讲解 Java 的多线程处理, 包括线程的创建、调度、同步和互斥关系。第 8 章介绍了 Java 的输入和输出, 包括字节流、字符流、文件和目录, 以及过滤流的使用。第 9 章讲解编写 Java 程序时用到的主要工具类, 包括集合、Math 等。第 10 章讲解 Java 数据库的连接。第 11 章介绍了 Java 网络编程, 包括 TCP 方式和 UDP 方式。第 12 章讲解 Java 的服务器端编程, 包括 Servlet 和表示技术 JSP。第 13 章详细讲解基于 Java 技术的四个综合应用实例。

全书立足基本概念和方法, 注重实践与应用。从应用的角度介绍基本概念、通过大量实例说明编程的方法和过程。对基本概念、理论和技术的介绍力求做到简洁、清晰、严谨、易懂。

全书每章都有小结和习题, 小结便于读者复习与总结, 习题基本上取自历年的国家计算机等级考试 Java 习题和 SCJP 考试题, 有助于读者检查学习效果, 巩固已学知识, 进一步提高应用能力。

本书由南阳理工学院的郑珂、刘克成担任主编, 张静、杨艳燕、张松娟担任副主编。郑珂编写了第 1、5、6、10 章, 刘克成编写了 4、13 章, 张静编写了第 3、11 章, 杨艳燕编写了第 9、12 章, 张松娟编写了第 2、7、8 章。

感谢海南大学三亚学院的王水教授审阅了全书, 并提出了许多宝贵意见, 同时感谢中国电力出版社的老师为本书的出版所付出的辛勤工作。另外本书在编写的过程中参阅了大量的文献内容, 在此一并表示感谢。

由于作者学识水平有限, 书中难免存有错误和不妥之处, 敬请广大读者批评指正, 不胜感激。如果有好的建议和要求, 请与作者联系, 电子邮件地址为 zanghshun@163.com。

编 者

2012 年 4 月

目 录

前言

第 1 章 Java 概述	1
1.1 Java 简介	1
1.2 Java 程序的开发	4
1.3 Java 的集成开发环境 Eclipse	8
1.4 小结	14
习题	14
第 2 章 Java 语言基础	15
2.1 简单的 Java 程序	15
2.2 标识符和关键字	17
2.3 数据类型	18
2.4 常量和变量	23
2.5 表达式和运算符	24
2.6 流程控制	28
2.7 小结	35
习题	35
第 3 章 面向对象程序设计	38
3.1 概述	38
3.2 基本概念	40
3.3 使用现有类创建对象	42
3.4 用户自定义类	47
3.5 封装	68
3.6 继承	69
3.7 多态	76
3.8 反射	79
3.9 接口	84
3.10 包	88
3.11 小结	90
习题	91
第 4 章 面向对象的几个基本原则	94
4.1 UML 类图	94
4.2 类间关系	95
4.3 面向对象的原则	99
4.4 小结	102

习题	102
第 5 章 异常处理	103
5.1 异常概述	103
5.2 异常处理	105
5.3 抛出异常	110
5.4 自定义异常	111
5.5 断言	114
5.6 小结	117
习题	117
第 6 章 图形用户界面	120
6.1 GUI 简介	120
6.2 事件处理	123
6.3 Swing 概述	128
6.4 容器和布局	133
6.5 几种常用的组件	143
6.6 小结	152
习题	152
第 7 章 多线程机制	155
7.1 线程与进程	155
7.2 线程的创建	156
7.3 线程生命周期	159
7.4 线程的优先级和调度	160
7.5 线程的常用方法	161
7.6 线程同步	165
7.7 计时器线程 Timer	173
7.8 线程组	174
7.9 小结	177
习题	177
第 8 章 输入/输出与文件	179
8.1 Java 输入/输出类库	179
8.2 字符的输入/输出	187
8.3 文件与文件输入/输出	190
8.4 数据流	198
8.5 随机流	201
8.6 对象的序列化	203
8.7 小结	206
习题	206
第 9 章 集合框架和实用工具	208
9.1 集合框架	208
9.2 迭代器	221

9.3 泛型	224
9.4 枚举	225
9.5 算法类	226
9.6 实用工具类	228
9.7 小结	230
习题	231
第 10 章 Java 数据库连接	234
10.1 结构化查询语言	234
10.2 JDBC 概述	237
10.3 JDBC 编程	239
10.4 JDBC 高级应用	250
10.5 数据库连接池	257
10.6 小结	261
习题	261
第 11 章 网络编程	264
11.1 网络基础知识	264
11.2 网络编程概述	267
11.3 网络层应用——地址的表示	267
11.4 URL	272
11.5 基于 Socket 的网络编程	278
11.6 UDP 网络程序设计	283
11.7 RMI 远程调用	287
11.8 小结	294
习题	294
第 12 章 服务器端编程	296
12.1 Web 应用概述	296
12.2 服务器端程序 Servlet	301
12.3 表示技术 JSP	305
12.4 JSP 基本语法	306
12.5 JavaBean 组件	316
12.6 小结	319
习题	319
第 13 章 综合实例	321
13.1 计算器	321
13.2 模拟交通信号灯	327
13.3 局域网聊天室	331
13.4 Web 留言板	341
参考文献	349

第 1 章 Java 概述

1.1 Java 简介

Java 是 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言（以下简称 Java 语言）和 Java 平台的总称。Java 的跨平台和动态的 Web/Internet 计算等特性迅速推动了 Web 的发展，并且 Java 语言是一种简单的、面向对象的、强安全性的语言，在网络编程语言中占据相当大的优势，Java 已经成为实现电子商务系统的首选语言。

1.1.1 Java 发展历程

Java 最早诞生于 1991 年，起初被称为 OAK 语言，是 Sun 公司为一些消费性电子产品设计的一个通用环境，其目的只是为了开发一种独立于平台的软件技术。

在网络出现之前，OAK 是默默无闻的，甚至差点“夭折”，但是网络的出现改变了 OAK 的命运。在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋 Web 浏览的人们来说简直不可容忍，他们迫切希望能在 Web 中看到一些交互式的内容，开发人员也极希望能够在 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，同时这些程序还要有极大的安全保障，对于用户的这些要求，传统的编程语言显得很无能为力。Sun 的工程师敏锐地察觉到了这一点，从 1994 年起，他们开始将 OAK 技术应用于 Web 上并且开发出了 Hot Java 浏览器，该浏览器显示了 Java 的跨平台和动态计算等魅力，从此 Java 被广泛接受并推动了 Web 的迅速发展。

1995 年 Sun 公司正式以 Java 这个名字将其推出，几乎所有的 Web 开发人员都为之欢呼，于是 Java 成了一颗耀眼的明星。1996 年 1 月，第一个 JDK-JDK1.0 诞生；同年 4 月，10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术；9 月，约 8.3 万个网页应用了 Java 技术来制作，Java 得到了广泛的应用。

1999 年 6 月，Sun 公司发布 Java 的三个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME），三个版本在应用场合和 API 的覆盖范围方面各不相同。J2SE（Java2 Standard Edition）是 Java2 的标准版，主要用于桌面应用程序的编程；J2ME（Java2 Micro Edition）主要应用于嵌入式系统开发，如手机和 PDA 的编程；J2EE（Java2 Enterprise Edition）是 Java2 的企业版，主要用于分布式网络程序的开发，如电子商务网站和 ERP 系统。在范围上，J2SE 包含 Java 语言核心的类，如数据库连接、输入/输出、网络编程等技术；J2EE 除了核心类之外还包含用于开发企业级应用的类，如 EJB、Servlet、JSP、XML、事务控制等技术；J2ME 把标准版进行压缩并添加一些可以运行在资源有限的设备上的特性。

2004 年 9 月 Sun 公司发布 JDK1.5，在该版本中加入了诸如泛型、自动装包/拆包、枚举、可变参数、静态导入等新特性来简化开发。2005 年 Sun 公司发布 JavaSE6.0，为了表示该版本的重要性，又把 J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

本教材以 Java 的标准版为主，Java SE6.0 为基础。

1.1.2 Java 语言

Java 语言是一种高级语言，它简单、面向对象、不依赖机器的结构，不受 CPU 和环境的限制，它具有可移植性、鲁棒性、安全性，并提供了并发机制，具有很高的性能；同时 Java 语言提供了丰富的类库，使程序设计人员可以方便地建立自己的系统。

1. Java 语言是简单的

Java 语言的语法与 C 语言和 C++ 语言很接近，因此大多数程序员很容易学习和使用 Java。同时，Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的特性，如操作符重载、多继承、自动的强制类型转换等。特别地，Java 语言不使用指针，并提供了自动垃圾回收功能，使得程序员不必为内存管理而担忧。

2. Java 语言是面向对象的

Java 语言是一个纯的面向对象的程序设计语言，提供类、接口和继承等原语。为了简单起见 Java 只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为 implements）。Java 语言全面支持动态绑定，而 C++ 语言只对虚函数使用动态绑定。

3. Java 语言是分布式的

Java 语言支持 Internet 应用的开发，在基本的 Java 应用编程接口中有一个网络应用编程接口（java.net），它提供了用于网络应用编程的类库，包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI（远程方法激活）机制也是开发分布式应用的重要手段。

4. Java 语言是健壮的

Java 的强类型机制、异常处理、垃圾自动回收等特点是 Java 程序健壮性的重要保证。对指针的丢弃和提供的安全检查机制使得 Java 更具健壮性。

5. Java 语言是安全的

Java 通常被用在网络环境中，为防止恶意代码的攻击，对通过网络下载类具有一个安全防范机制（类 ClassLoader），如分配不同的名字空间以防替代本地的同名类、字节代码检查，并提供安全管理机制（类 SecurityManager）让 Java 应用设置安全哨兵。

6. Java 语言是跨平台的

Java 程序（扩展名为 .java 的文件）在 Java 平台上被编译为体系结构中立的字节码格式（扩展名为 .class 的文件），然后可以在支持 Java 平台的任何系统中运行。这种途径适合于异构的网络环境和软件的分发。

7. Java 语言是可移植的

Java 语言的可移植性来源于体系结构中立性，另外 Java 还严格规定了各个基本数据类型的数据长度。Java 系统本身也具有很强的可移植性，Java 编译器是用 Java 实现的，Java 的运行环境是用 ANSIC 实现的。

8. Java 语言是解释型的

如前所述，Java 程序在 Java 平台上被编译为字节码格式，并可以在实现这个 Java 平台的任何系统中运行。在运行时，Java 平台中的 Java 解释器对这些字节码进行解释执行，执行过程中需要的类在连接阶段被载入到运行环境中。

9. Java 是高性能的

与那些解释型的高级脚本语言相比，Java 的确是高性能的。事实上，Java 的运行速度随

着 JIT (Just-In-Time) 编译器技术的发展越来越接近于 C++。

10. Java 语言是多线程的

在 Java 语言中, 线程是一种特殊的对象。Java 语言支持多个线程的同时执行, 并提供多线程之间的同步机制。

11. Java 语言是动态的

Java 语言的设计目标之一是适应动态变化的环境。Java 程序需要的类能够动态地被载入到运行环境, 也可以通过网络来载入所需要的类, 这有利于软件的升级。另外, Java 中的类有一个运行时刻的表示, 能进行运行时刻的类型检查。

Java 语言的优良特性使得 Java 应用具有无比的健壮性和可靠性, 同时减少了应用系统的维护费用。Java 对对象技术的全面支持和 Java 平台内嵌的 API 能缩短应用系统的开发时间并降低成本。Java 的编译一次就可到处运行的特性, 使得它能够提供一个随处可用的开放结构和在多平台之间传递信息的低成本方式, 特别是 Java 企业应用编程接口 (Java Enterprise APIs) 为企业计算及电子商务应用系统提供了有关技术和丰富的类库。

1.1.3 Java 平台

Java 语言环境由编译时系统和运行时系统两大部分组成。编译时系统把编写好的 Java 程序编译成字节码指令, 运行时系统则执行这些字节码指令实现应用。运行时系统又称为 Java 平台。

平台是一个程序运行的软硬件环境, 人们所熟知的 Windows 2000、Linux 等是操作系统和硬件组成的基于硬件的平台, Java 平台不同于这些平台, 它是一种全新的软件平台, 是建立在以上提到的各种平台之上的, 能在计算机上运行高度交互的、动态的、安全可靠的小应用程序和应用程序。Java 平台包括两个主要组成部分: Java 虚拟机 (the Java Virtual Machine) 和 Java 应用编程接口, 即 Java API (the Java Application Programming Interface)。

Java 虚拟机 (JVM) 是一个由软件模拟实现的虚拟计算机, 是 Java 平台的核心和 Java 跨平台的基础, 它解释运行 Java 程序。执行过程如图 1-1 所示, Java 的源文件 (.java) 经编译生成字节码文件 (.class), 然后 JVM 把字节码文件解释为机器码并执行。字节码指令, 非机器码指令, 这种字节码与特定的硬件无关, 是 Java 虚拟机能解释的指令。从这个过程可以看出为了运行 Java 程序, 计算机上必须安装 JVM 和 Java 解释器。在任何操作系统下, 只要安装了 Java 平台, 就可运行 Java 程序。

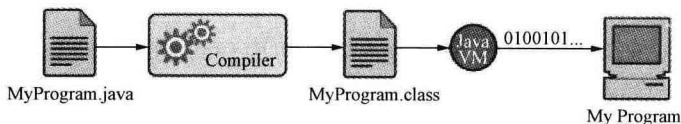


图 1-1 Java 应用程序的编译运行过程

对于每个 Java 平台下面的底层平台 (操作系统), Java 虚拟机的具体实现是不同的, 但只有一套 Java 虚拟机规范, 因此 Java 平台为任何硬件平台提供了一个标准的、相同的编程接口。Java 应用编程接口——Java API 是 Java 应用的标准接口, 是应用开发的基本框架, 是开放的和可扩展的。它为各个关键领域规定了一系列的基本接口, 提供给程序员用来进行 Java

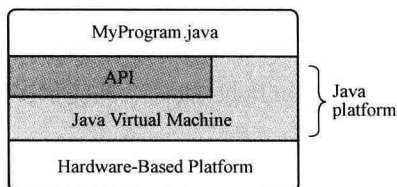


图 1-2 Java 平台与 Java 程序

应用开发。图 1-2 所示描述了一个 Java 程序和 Java 平台的关系。

从图 1-2 中可知 Java 程序是运行在 Java 平台上的而不是直接在基于硬件的平台上，要开发 Java 程序就需安装所必须的 JavaAPI 和 JVM。因此 Java 平台也可以称为 Java 开发工具集 Java Developer Kit，简称 JDK，截止本书出版其最新的版本是 JDK 6。

1.2 Java 程序的开发

Java 可以开发三种程序：Java 应用程序（Application）、Java 小应用程序（Applet）和 Java 服务器端程序（Servlet）。Java Application 是可以独立运行的程序；Java Applet 是必须要嵌入到网页中，通过浏览器运行的程序；而 Servlet 是运行在 Web 服务器端的程序。无论是哪种 Java 程序，在开发时都需要使用以下两种工具。

（1）Java 的开发包 JDK。

（2）文本编辑器。任何一个文本编辑器都可以使用，最简单的是 Windows 中自带的记事本。

1.2.1 Java 开发环境

Java 为不同硬件平台提供了不同的 Java 虚拟机，所以 Java 可以在不同的平台（Windows、Linux）上运行，本书以 Windows 平台为基础，讲解 Java 应用程序的开发。

1. 下载 JDK

可在 Java 的官方网站(<http://www.oracle.com/technetwork/java/index.html>)上下载 Windows 环境下的开发工具包，本书使用 JDK 6（全称为 Java SE Development Kit 6），并下载 Java API 的中文文档，具体文件如下：

（1）jdk-1_6_0_10-windows-i586-p.exe，JDK 的开发包；

（2）JDK_API_1_6_zh_CN.html，JDK 的中文文档。

2. 安装 JDK

安装 JDK 非常简单，只需运行 jdk-1_6_0_10-windows-i586-p.exe 文件就可以，在安装的过程中可以设置安装的路径，还可以选择安装的组件，默认的安装路径为 C:\Program Files\Java（假设系统安装在 C 盘）。

安装 JDK 后将生成 JRE 和 JDK 两部分内容。JRE（Java Run Time Enviroment）即 Java 程序的运行环境，面向最终用户，用来运行、测试和传输 Java 程序，它包括 Java 虚拟机(JVM)、Java 平台核心类和一些支持文件。JDK 是开发 Java 程序时所用的 Java 开发工具集，面向开发人员，除了包含一个完整的可以运行 Java 程序的 JRE 外，还包含用于开发的各种工具集，如 Java 编译器 javac。

3. 配置 JDK

安装完 JDK 后，并不能立即使用，要能够进行 Java 程序的编译和调试，必须告诉操作系统完成这些操作的应用程序在什么位置，因此必须先配置环境变量。安装目录的 jdk 文件夹中的目录文件及它们的功能如表 1-1 所示。

表 1-1

JDK 的目录结构

文件/目录名	功能描述
bin	包含编译器、解释器等主要的可执行文件
lib	可执行程序使用的归档类包
jre	Java 运行时环境的根目录
src.zip	核心 API 所有类的源文件的归档包
demo	包含源代码的程序示例
include	用于本地计算机的 C 语言头文件
sample	范例

目录 bin 包含了编译和运行 Java 程序的可执行文件，目录 lib 包含了开发 Java 程序所需的基本类库。因此为了能够让系统正确使用 JDK，就必须在系统的环境变量中指定 bin 目录和 lib 目录的位置。下面以 Windows XP 系统为例介绍环境变量的配置。

右击桌面“我的电脑”图标，选择“属性”选项，出现系统设置界面，在“高级”选项卡中单击“环境变量”按钮，进入如图 1-3 所示对话框。

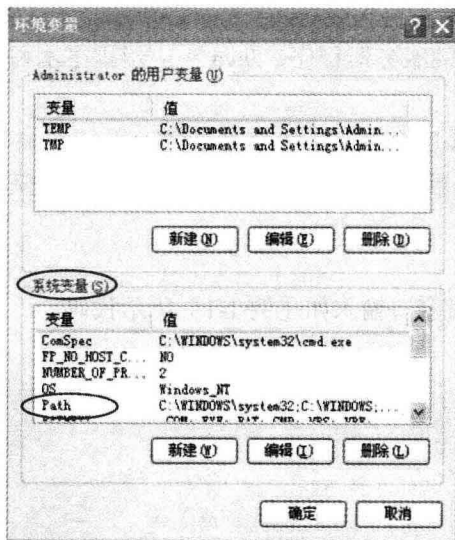


图 1-3 系统的环境变量对话框

在“系统变量”组框内找到变量“Path”，该变量是用来告诉操作系统一些可执行文件的位置。单击“编辑”按钮，将“;C:\Program Files\Java\jdk1.6.0_10\bin;”加入到“变量值”文本框内（C:\Program Files\Java 为 JDK 的安装目录），如图 1-4 所示。

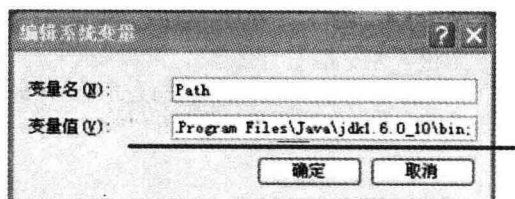


图 1-4 环境变量“Path”的编辑

如果没有找到该变量，可以在图 1-3 中的环境变量对话框中单击“新建”按钮，设置变量名为“Path”，变量值为“C:\Program Files\Java\jdk1.6.0_10\bin;”。

为了能够在 Java 程序中使用 Java 类库，可以用同样的方法设置环境变量“classpath”，该变量的值为“.C:\Program Files\Java\jdk1.6.0_10\lib”（注意目录的前面加上“.”）。

4. 验证

如果设置正确，在 DOS 命令提示符下输入“javac”命令，在屏幕上会出现该命令的参数，如图 1-5 所示。否则，将会显示“系统找不到 javac 文件”。

设置正确后就可以进行 Java 程序的编译和运行了。



图 1-5 javac 命令参数

1.2.2 Java 程序的开发步骤

本书以 Java 应用程序为主，第 12 章讲解 Java 的服务器程序，Java 小应用程序本书不做介绍。

本节假定操作系统为 Windows，使用 Windows 的命令提示符来编译和运行一个 Java 应用程序，该程序将输出字符串“Hello,Word!”。如果使用 Linux 操作系统，这些步骤也是基本相同的。

1. 编写 Java 源程序

(1) 打开“文本编辑器”→“记事本”，在编辑窗口输入如 [例 1-1] 所示代码。

【例 1-1】 第一个 Java 程序，参考代码为：

```
public class Hello{
    public static void main(String args[]){
        System.out.println("Hello,World!");
    }
}
```

注意程序中应区分大小写，如 String 不能写成 string，因为 Java 是对大小写敏感的。程序中 String 和 System 是系统定义类库，public、static、void、class 等是 Java 的关键字。

Java 是面向对象程序设计，Java 应用开发的过程就是类的设计和编写过程，“class”关键字用来定义一个类，类名是“Hello”，类名后面的一对花括号是该类的类体，一般是方法和变量。

public static void main (String args[]) 用来定义 main 方法，该方法是 Java 应用程序的入口，有了该方法，程序才能够独立运行。关键字“public”表示该方法是公共方法，关键字“static”表示该方法为静态方法或者成为类方法，关键字“void”表示该方法不返回任何数据类型，main 后面的一对圆括号内是方法参数，main 方法的参数是一个字符串数组，可以写为“String args[]”，也可以写为“String[]args”。

main 方法体内只有一条语句，该语句的作用是在默认的输出窗口输出字符串“Hello,World!”。

(2) 保存 Java 源程序。编写好代码后，就可以保存 Java 程序了，如图 1-6 所示，在保存时要注意以下几个方面。

1) Java 源程序的扩展名为.java，文件的名称应和程序中类名称保持完全一致，本例的文件名称即为 Hello.java。进行文件的保存时要注意文件名称的大小写。

2) 由于记事本默认的文件类型为文本文件扩展名，即.txt，所以需要把“保存类型”从文本文件改为“所有文件”，以便保存.java 类型文件。

3) 在保存时注意把文件保存在一个方便的目录下，并记住文件的保存位置。本例是把文件保存在“C:\java”目录下，如图 1-6 所示。

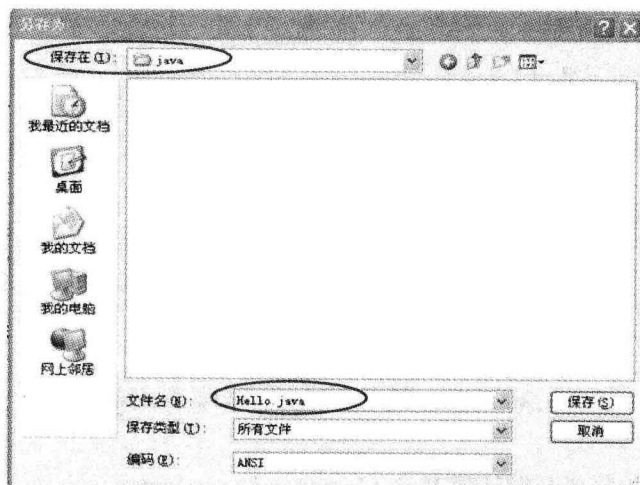


图 1-6 保存 java 源程序

2. 编译源程序

Java 源程序需要经过编译生成字节码文件 (*.class) 后，才能够由 JVM 解释执行，编译 Java 文件的步骤如下。

(1) 打开 Windows 命令提示符。可以单击“开始”按钮，在“运行”窗口中输入命令提示符“cmd”，或选择命令提示符应用程序图标来启动（在 Windows 系统的附件中）。

(2) 使用 cd 命令把目录转换到 Java 源文件所在的目录，命令如下：

```
cd c:\java
```

(3) 输入编译命令“javac”，调用 Java 编译器编译源文件。注意文件名称不能写错，注意大小写，而且要带上扩展名“.java”，命令如下：

```
javac Hello.java
```

(4) 查看是否编译成功。如果编译通过，将会生成字节码文件 Hello.class，在命令提示符窗口中使用“dir”命令察看，如图 1-7 所示。如果程序中有错误出现，则系统会终止编译并给出相应的错误信息；如果系统指出未找到 javac.exe 命令，则说明 path 环境变量设置不正确。

3. 运行字节码文件

经过编译之后,生成的字节码文件,就可以由 JVM 解释执行,在命令提示符中输入运行 Java 程序命令并加上字节码文件名,就可以运行程序了,命令如下:

```
java Hello
```

运行结果如图 1-7 所示。

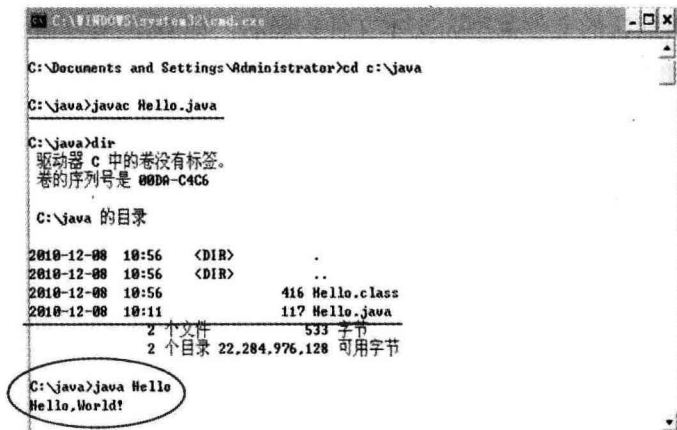


图 1-7 编译 Java 程序

1.3 Java 的集成开发环境 Eclipse

Eclipse 是 Java 的集成开发环境 IDE (Integrated Development Environment), IDE 主要用于程序开发,集成了代码编写、分析、编译、debug 等功能,使程序开发更简单、方便。Java IDE 常用的有 Eclipse、NetBeans、Jbuilder 等。

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。同时 Eclipse 附带了一个标准的插件集,包括 Java 开发工具 (Java Development Tools, JDT)。

Eclipse 最初是 IBM 公司的一个软件产品,2001 年 11 月贡献给开源社区,现在它由非营利软件供应商联盟——Eclipse 基金会 (Eclipse Foundation) 管理。Eclipse 面世之初曾经受到业界的批评,但是 2003 年 3 月 Eclipse 发布了 2.1 版,立刻引起了轰动,人们蜂拥下载该软件,导致 Eclipse 的服务器瘫痪。

Eclipse 是一个通用的工具平台——普遍使用的、开放式的、扩展的 IDE。它本身只是一个框架平台,但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。它提供了功能丰富的开放环境,允许开发者高效地创建一些能够无缝集成到 Eclipse 平台中的插件,它的设计思想是:一切皆为插件。如图 1-8 所示, Eclipse 的核心是非常小的,其所有的功能都以插件的形式存在。如用于图形开发的插件 (SWT/Jface)、Java 的开发环境插件 (JDT)、插件的开发环境 (PDF) 等。Eclipse 对这些插件的管理是动态的载入和调用,即只有当真正使用这些插件时才调入内存,不用时会适当的时候消除。

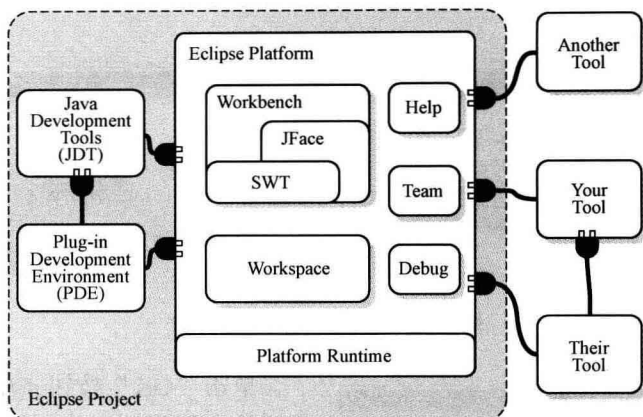


图 1-8 Eclipse 的体系结构

Eclipse 作为开发 Java 的 IDE，从出现到现在已经有几年的时间了，其因具有特性丰富、操作容易和生产力提高等优点受到了广大开发者的青睐，本书采用 Eclipse 作为 Java 程序的开发环境。

1.3.1 Eclipse 安装和运行

Eclipse 是应用 Java 语言编写的软件，所以需要 Java 运行环境（JRE）的支持，因此在安装 Eclipse 之前要先安装 JDK。

Eclipse SDK（Software Development Kit）即 Eclipse 软件开发工具包，Eclipse 组织提供标准免费工具包供用户下载，官方地址为 <http://www.eclipse.org/downloads/index.php>。本书使用 eclipse-SDK-3.6.1-win32，它包含 Eclipse 开发环境、Java 开发环境、Plug-in 开发环境，以及所有源代码和文档。

安装 Eclipse 的步骤非常简单，只需将下载的压缩包 eclipse-SDK-3.6.1-win32.zip 按原路径直接解压即可。在解压缩后的目录中找到 Eclipse.exe 文件，如图 1-9 所示，双击该文件启动 Eclipse，启动画面闪现一个很酷的月蚀（eclipse）图片。

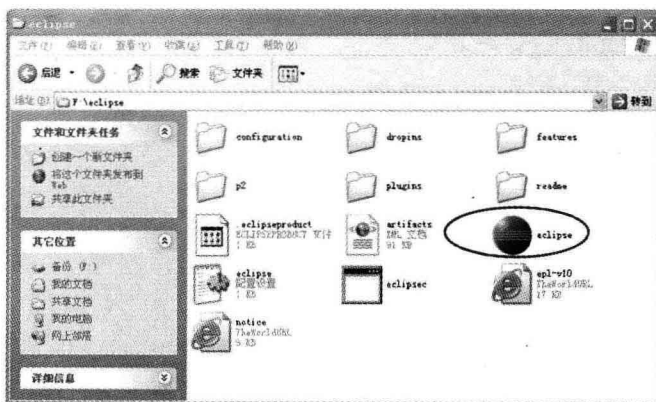


图 1-9 安装后 Eclipse 的启动文件

然后可以设置个人的工作空间，以后用户所创建的 Java 工程和应用程序默认放置在这个目录下，如图 1-10 所示，单击“Browse”按钮选择合适的工作空间，并勾选“Use this as the

default and do not ask again”。

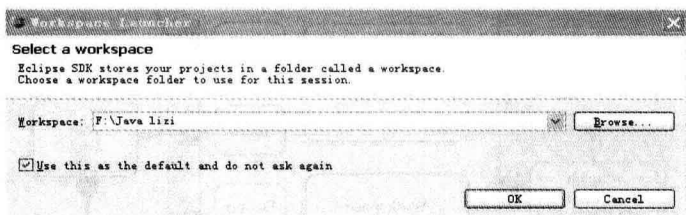


图 1-10 设置个人的工作环境

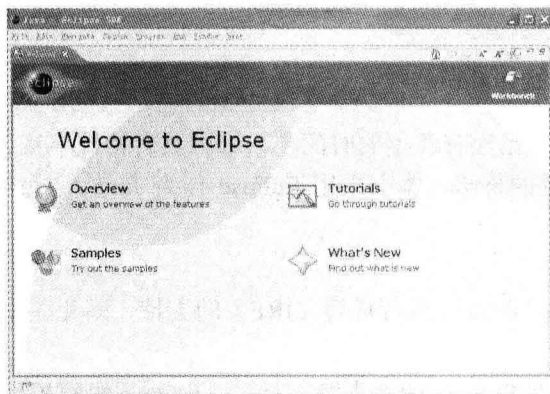


图 1-11 Eclipse 欢迎界面

单击“OK”按钮完成工作空间的设置，系统将会显示 Eclipse 的欢迎界面（如果不能显示欢迎界面，则选择“Help”→“Welcom”命令即可显示欢迎界面），如图 1-11 所示。

1.3.2 Eclipse 中文包的安装

1.3.1 节中 Eclipse SDK 是英文界面，如果用户习惯使用中文界面，则需要安装 Eclipse 的中文插件。下载时需要注意下载和 Eclipse 版本相同的中文包，在本书中 Eclipse 版本是 3.6.1，所以下载的语言类包的版本应是 3.6.1（如 BabelLanguagePack-eclipse-zh_3.6.）。

语言包的运行和安装步骤如下：

- (1) 解压下载的语言包中的.zip 文件；
- (2) 先关闭 Eclipse，然后将语言包解压缩后的 features 和 plugins 目录中的所有文件分别复制到 Eclipse 安装目录中的 features 和 plugins 目录中，完成安装；
- (3) 如果操作系统是中文的 Windows，直接打开 Eclipse 便可以显示中文界面，关闭 Eclipse 的中文欢迎界面，将进入 Eclipse 的工作台，如图 1-12 所示；

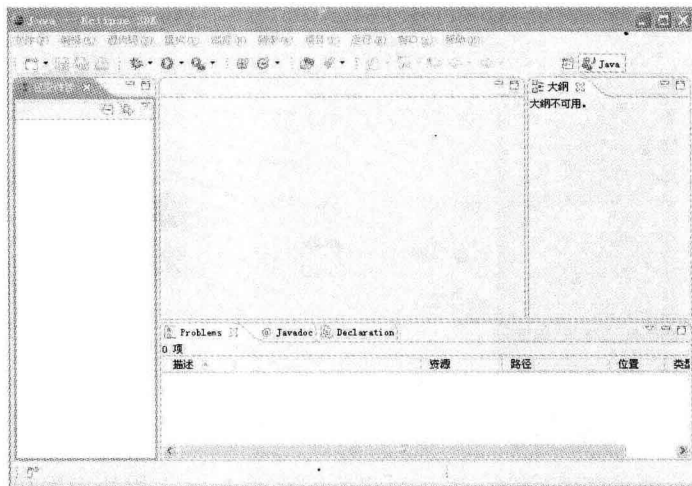


图 1-12 Eclipse 中文工作台