



国家示范性高等职业教育电子信息大类“十三五”规划教材

Java

程序设计基础

主编◎肖英



华中科技大学出版社
<http://www.hustp.com>



国家示范性高等职业教育电子信息大类“十三五”规划教材

Java

程序设计基础

主 编 肖 英

副主编 刘 嵩 赵丙秀 马 力 彭 军 马金霞

参 编 刘 洁 郭 俐



华中科技大学出版社
<http://www.hustp.com>

内 容 提 要

本书知识点系统连贯、逻辑性强;重视实际应用,案例丰富;学习模块划分合理,重难点突出,利于组织教学。

本书包括 10 个项目:项目 1 认识 Java 语言,项目 2 开发第一个 Java 程序,项目 3 Java 语言基础,项目 4 程序的流程控制,项目 5 方法与数组,项目 6 类与对象,项目 7 面向对象特性,项目 8 常用 API,项目 9 异常处理,项目 10 JDBC 数据库编程。

为了方便教学,本书还配有电子课件等教学资源包,任课教师和学生可以登录“我们爱读书”网(www.ibook4us.com)免费注册并浏览,或者发邮件至 hustpeit@163.com 免费索取。

本书适用于高职高专、应用型本科、中职等层次学校的相关专业使用,可作为 Java 初学者的入门教材和 Java 相关课程设计教材,也可以作为 Java 开发工程师的培训教材,还可以作为企业岗前培训教材。

图书在版编目(CIP)数据

Java 程序设计基础/肖英主编. —武汉:华中科技大学出版社,2017.1

国家示范性高等职业教育电子信息大类“十三五”规划教材

ISBN 978-7-5680-2372-6

I. ①J… II. ①肖… III. ①JAVA 语言-程序设计-高等职业教育-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 278273 号

Java 程序设计基础

Java Chengxu Sheji Jichu

肖 英 主 编

策划编辑:康 序

责任编辑:史永霞

封面设计:抱 子

责任监印:朱 玢

出版发行:华中科技大学出版社(中国·武汉)

电话:(027)81321913

武汉市东湖新技术开发区华工科技园

邮编:430223

录 排:武汉楚海文化传播有限公司

印 刷:武汉科源印刷设计有限公司

开 本:787mm×1092mm 1/16

印 张:14.5

字 数:368 千字

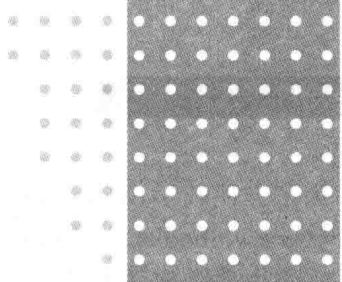
版 次:2017 年 1 月第 1 版第 1 次印刷

定 价:35.00 元



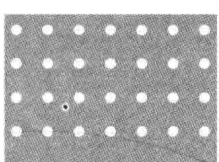
华中科技大学出版社

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究



FOREWORD

前言



Java 语言具有简单、面向对象、分布式、健壮性、安全性、可移植性、多线程、高性能等诸多优点,是近十年来计算机编程语言排行榜上的佼佼者,可以用于开发各种领域的应用软件。熟练掌握 Java 语言是软件从业人员的必备技能。

本书的目标是帮助广大高职高专学生学习和掌握 Java 程序设计语言的核心基础知识和技术。本书知识点系统连贯、逻辑性强;重视实际应用,案例丰富;学习模块划分合理,重难点突出,利于组织教学。作为一种编程语言和技术的入门教程,最难也最重要的是令一些非常复杂又难以理解的编程思想和问题简单化,令初学者能尽量轻松理解并快速掌握。本书对每个知识点都进行了较详尽的讲解,精心设计了相关的例程,尽量用生动形象的案例来讲解抽象的编程思想和模拟这些知识点在实际工作中的可能运用,力求做到知识由浅入深、由易到难,循序渐进地引导学生和读者逐步掌握 Java 程序设计的核心基础知识和技术。

本书共包含 10 个项目。项目 1 认识 Java 语言,介绍计算机程序和 Java 程序设计语言的基本情况;项目 2 开发第一个 Java 程序,主要介绍 JDK 的安装使用、Java 的集成开发环境的安装使用,并动手实现第一个 Java 应用程序;项目 3 Java 语言基础、项目 4 程序的流程控制 and 项目 5 方法与数组,详细介绍 Java 语言的核心语法、流程控制语句、方法的定义和调用、数组的使用等,这些都是使用 Java 语言编写应用程序代码的最重要的基本知识,读者一定要认真学习每一个知识点;项目 6 类与对象和项目 7 面向对象特性,详细介绍了 Java 面向对象编程的主要知识和技术,对于初学者来说面向对象的编程思想需要多思考、多实践、花较大精力来理解和掌握;项目 8 常用 API,介绍了 Java 系统类库提供给程序员使用的一些常用且实用的类,尤其是集合类在实际开发中很常用;项目 9 异常处理,介绍了 Java 的异常处理机制和异常处理语句,可以令读者编写的程序更健壮和更友好;项目 10 JDBC 数据库编程,详细介绍了 Java 连接数据库和对数据库进行操作的步骤和代码的编写,JDBC 数据库编程技术在实际开发中运用十分广泛,读者应认真学习和掌握 JDBC 数据库编程技术,为后续运用 Java 语言开发各种类型的应用程序打下坚实的基础。

Java 程序设计课程是实践性较强的课程,读者需要通过大量的实践以掌握 Java 语言的使用、编程技巧,养成良好的规范编码的习惯,因此本书每个项目均设计了许多例程,使读者能通过这些例程加强对知识点的理解,也能通过动手实现这些例程而潜移默化地锻炼动手编码能力。每章结束时都给出了典型的实训任务,通过对这些实训任务的实践练习,可以进一步锻炼和提高编码能力。

为了方便教学,本书还配有电子课件等教学资源包,任课教师和学生可以登录“我们爱读书”网(www.ibook4us.com)免费注册并浏览,或者发邮件至 hustpeiit@163.com 免费索取。

本书由武汉软件工程职业学院肖英任主编;由武汉软件工程职业学院刘嵩、赵丙秀、马力,江西应用科技学院彭军,青岛理工大学琴岛学院马金霞任副主编;由武汉软件工程职业学院刘洁、郭俐任参编,最后由肖英统稿。

由于时间仓促,加之编者水平有限,书中不妥或错误之处在所难免,殷切希望广大读者批评指正。同时,恳请读者一旦发现错误,及时与编者联系,以便尽快更正,编者将不胜感激。

编者

2016年12月

CONTENTS

目录

项目 1 认识 Java 语言	1
1.1 计算机程序	1
1.2 Java 程序设计语言	2
1.3 Java 平台介绍	4
项目 2 开发第一个 Java 程序	6
2.1 下载、安装和配置 JDK	6
2.2 JDK+记事本开发 Java 程序	8
2.3 集成开发环境 Eclipse	10
2.4 使用 Eclipse 开发 Java 程序	12
2.5 Java 程序开发的一般流程	14
项目 3 Java 语言基础	15
3.1 Java 语言的基本语法	15
3.2 常量与变量	17
3.3 运算符与表达式	25
3.4 简单的输入输出	33
项目 4 程序的流程控制	36
4.1 流程控制语句	36
4.2 选择结构语句	37
4.3 循环结构语句	46
4.4 跳转控制语句	57
项目 5 方法与数组	61
5.1 方法	61
5.2 数组	69
项目 6 类与对象	85
6.1 类与对象的基本概念	85

6.2 创建类	86
6.3 对象的使用	90
项目 7 面向对象特性	97
7.1 对象的创建与销毁	97
7.2 引用赋值	100
7.3 方法	105
7.4 类的静态成员	115
7.5 继承	118
7.6 抽象类与接口	125
7.7 类的转型	132
7.8 内部类	135
7.9 包与访问控制修饰符	139
项目 8 常用 API	147
8.1 API 的概念	147
8.2 字符串处理	149
8.3 Math 类	158
8.4 日期时间类	159
8.5 集合类	162
项目 9 异常处理	177
9.1 异常概述	177
9.2 处理异常	180
9.3 throw 和 throws 关键字	184
9.4 自定义异常	186
9.5 Java 的内置异常	187
项目 10 JDBC 数据库编程	190
10.1 JDBC 数据库编程概述	190
10.2 JDBC 数据库编程基本操作	195
10.3 JDBC 编程进阶	213
参考文献	224

本章目标

- ◆ 计算机程序
- ◆ Java 语言的特点



1.1 计算机程序

计算机程序告诉计算机应该做什么。计算机执行的任何操作都是由程序控制的。程序设计是将计算机要执行的操作或者计算机要解决的问题转变成程序的过程。程序设计的过程主要包括分析问题、确定算法、用选定的程序设计语言编写源程序、调试和运行程序。

程序设计语言是计算机能够理解的、用于人和计算机之间进行交流的语言。现实世界中，当需要与英国人交流的时候会选择说英语，当需要与法国人交流时会选择说法语，同样，当需要与计算机交流的时候应该选择程序设计语言。

程序员可以使用各种程序设计语言编写计算机程序，计算机程序设计语言种类繁多，大致可以划分为三大类：

- 机器语言；
- 汇编语言；
- 高级语言。

所有计算机都能直接理解自己的机器语言。机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合，是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有直接执行和速度快的特点。机器语言对于计算机而言易于理解，对于程序员而言就极难理解了。程序员用机器语言编写程序需要熟记所用计算机的全部指令代码和代码的含义，需要处理每条指令和每一数据的存储分配和输入输出等，编程工作十分烦琐和耗时，一般只有计算机生产厂家的专业人员会使用机器语言。

汇编语言也称为符号语言，汇编语言使用助记符来代替机器指令的操作码，用地址符号或标号代替指令或操作数的地址。在不同的设备中，汇编语言对应着不同的机器语言指令集，通过汇编过程转换成机器指令。汇编语言相对于机器语言稍易于理解，但仍然是更贴近计算机器底层的。汇编语言通常被应用于计算机底层应用、硬件操作和高要求的程序优化的场合等，如驱动程序、嵌入式操作系统和实时运行程序等。

由于汇编语言助记符量多而难以记忆,且汇编语言依赖于具体的硬件体系,于是又产生了近似于人类自然语言的高级程序设计语言(简称高级语言)。高级语言的语法和结构更类似普通英文和数学语言,且与计算机的硬件结构及指令系统无关,因此它的表达力强,容易学习掌握,编程相对直观简单,可移植性和通用性更好。由于高级语言不能被计算机所理解和执行,因此高级语言编译生成的程序代码一般比用汇编语言设计的程序代码要长些,执行的速度也要慢些。高级语言通常按其基本类型、代系、实现方式、应用范围等分类。微软的 C# 和 Oracle 的 Java 都是目前应用广泛、功能较强大的高级程序设计语言。



1.2 Java 程序设计语言

1991 年, SUN MicroSystem 公司的 James Gosling、Bill Joe 等人,为在电视、烤面包箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak 的软件(即一种橡树的名字, Java 语言的前身),但当时并没有引起人们的注意,直到 1994 年下半年, Internet 的迅猛发展,加快了 Java 语言研制的进程,使得它逐渐成为 Internet 上最受欢迎的编程语言。2009 年,甲骨文股份有限公司(Oracle)收购了 SUN MicroSystem 公司,此后由甲骨文股份有限公司继续推进 Java 语言的发展。

Java 语言发展迅速并被广泛应用,是目前使用最具市场的网络编程语言之一,这与 Java 语言本身的特点是密切相关的。Java 语言具有简单、面向对象、分布式、鲁棒性、安全、体系结构中立、可移植性、解释执行、高性能、多线程、动态等多种特点。

1. 简单

Java 语言是一种面向对象的语言,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 略去了运算符重载、多重继承等模糊的概念,并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外,Java 也适合于在小型机上运行,它的基本解释器及类的支持只有 40 KB 左右,标准类库和线程的支持也只有 215 KB 左右。

2. 面向对象

Java 语言的设计集中于对象及其接口,它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法,实现了模块化和信息隐藏,而类则提供了一类对象的原型,并且通过继承机制,子类可以使用父类所提供的方法,实现了代码的复用。

3. 分布式

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议,用户可以通过 URL 地址在网络上很方便地访问其他对象。

4. 鲁棒性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。它提供自动垃圾收集来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成的面向对象的异常处理机制,在编译时,Java 提示出可能出现但未被处理的异常,帮助程序员正确

地进行选择以防止系统的崩溃。另外,Java 在编译时还可捕获类型声明中的许多常见错误,防止动态运行时不匹配问题的出现。

5. 安全

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员,同时也避免了指针操作中容易产生的错误。

6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行时刻环境,Java 程序就可在任意的平台上运行。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时,Java 的类库中也实现了与不同平台的接口,使这些类库可以移植。另外,Java 编译器是由 Java 语言实现的,Java 运行时刻环境由标准 C 实现,这使得 Java 系统本身也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时信息,使得连接过程更加简单。

9. 高性能

和其他解释执行的语言如 BASIC 不同,Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码,从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,我们可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就能很容易地实现网络上的实时交互行为。

11. 动态

Java 语言的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。

Java 语言的特性给网络编程带来了许多方便。根据用途的不同,Java 语言可分为以下三种版本:

Java SE(Java Standard Edition):Java 的标准版,主要运用于桌面级的应用和数据库的开发。

Java EE(Java Enterprise Edition):Java 的企业版,主要用于企业级开发,主要提供了企业级 Web 应用开发的各种技术。

Java ME(Java Micro Edition):Java 的移动版,主要用于嵌入式的和移动式的开发,如手机应用软件开发。



1.3 Java 平台介绍

Java 平台由两大部分组成：Java 虚拟机(Java Virtual Machine,JVM)和 Java 应用编程接口(Java Application Programming Interface,Java API)。

Java 设计的初衷是使要建的能在任何平台上运行的程序不需要再在每个单独的平台上由程序员进行重写或重编译。Java 虚拟机使这个愿望变为可能,因为它能知道每条指令的长度和平台的其他特性。JVM 是通过在实际的计算机上仿真模拟各种计算机功能来实现的。Java 虚拟机有自己完善的硬件架构,如处理器、堆栈、寄存器等,还具有相应的指令系统。JVM 屏蔽了与具体操作系统平台相关的信息,使得 Java 程序只需生成在 Java 虚拟机上运行的目标代码(字节码),就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时,实际上最终还是把字节码解释成具体平台上的机器指令执行的。

Java API 是一些预定义的类库,开发人员需要用这些类来访问 Java 语言的功能。Java API 包括一些重要的语言结构以及基本图形、网络和文件 I/O、数据库操作组件等,是软件组件的集合。

图 1-1 展示了 Java SE 的整个平台系统：

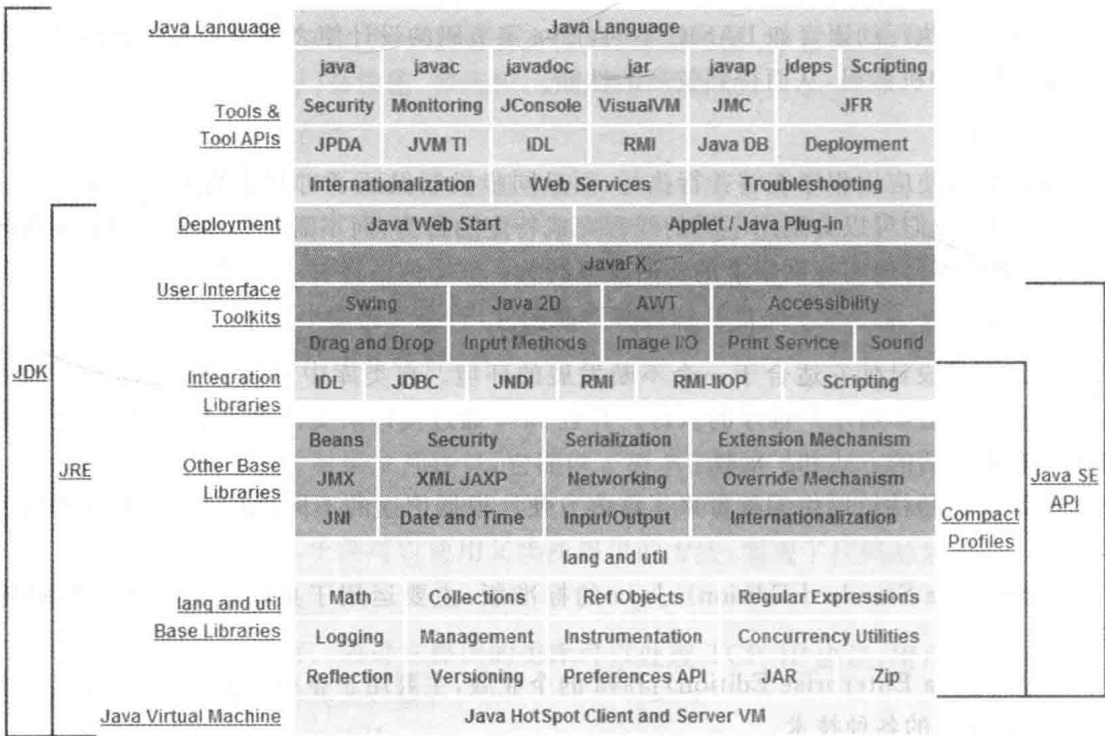


图 1-1 Java SE 平台系统

图 1-1 中,JDK 为 Java Development Kit,即 Java 开发工具包,包括运行环境、编译工具及其他工具、源代码等。而 JRE 为 Java Runtime Environment,即 Java 运行环境,运行 Java 程序

所必需的环境的集合,包含 JVM 标准实现及 Java 核心类库。

当我们在一台提供了 Java 运行环境 JRE 的计算机上运行 Java 程序,如应用程序 Java Program,通过 Java API 和 JVM 可以把 Java 程序从硬件依赖中分离出来,如图 1-2 所示。

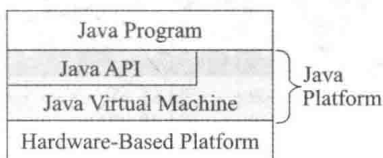


图 1-2 分离 Java 程序

作为一种独立于硬件平台的编程语言,Java 代码比本地代码慢一些,但随着技术的不断发展,Java 语言的表现不牺牲可移植性的前提下也在不断地接近本地代码。

通过编译器,可以把 Java 程序源代码编译成一种中间代码,称为字节码。字节码可以看作是运行在 Java 虚拟机上的机器代码指令,可以运行在任何 JVM 上。Java 字节码使得 Java 程序编译一次,到处运行成为可能。

要点提醒:

- ◇程序设计语言是计算机能够理解的、用于人和计算机之间进行交流的语言。
- ◇程序设计的过程主要包括分析问题、确定算法、用选定的程序设计语言编写源程序、调试和运行程序。
- ◇计算机程序设计语言可以划分为三大类:机器语言、汇编语言、高级语言。
- ◇Java 语言的三种版本:Java SE(Java Standard Edition),Java EE(Java Enterprise Edition)和 Java ME(Java Micro Edition)。
- ◇Java 平台由两大部分组成:Java 虚拟机(Java Virtual Machine)和 Java 应用编程接口(Java Application Programming Interface)。
- ◇JDK 为 Java Development Kit(Java 开发工具包),包括运行环境、编译工具及其他工具、源代码等。而 JRE 为 Java Runtime Environment(Java 运行环境),运行 Java 程序所必需的环境的集合,包含 JVM 标准实现及 Java 核心类库。
- ◇Java 语言程序具有编译一次,到处运行的特点。

实训任务 □□□

[实训 1-1]浏览甲骨文股份有限公司网站,了解 Java 语言的发展动态。

[实训 1-2]通过互联网了解比较流行的开发 Java 程序的工具及其特点。

项目 2 开发第一个Java程序

本章目标

- ◆ JDK 的安装和配置
- ◆ 使用 JDK+记事本开发 Java 程序
- ◆ 使用 Eclipse 集成开发环境开发 Java 程序



2.1 下载、安装和配置 JDK

“工欲善其事，必先利其器”，在开始 Java Application 的编程之旅前，须先准备好开发环境。JDK 即 Java 开发工具包，包括了 Java 编译工具、运行环境、其他工具和 Java 基础类库等，JDK 是其他 Java 集成开发工具的基础。编写 Java 程序首先需要安装和配置好 JDK。

Step1: 下载和安装 JDK

登录 Oracle 公司官网，找到 Java SE 的下载页面 (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)，选择“Java SE Downloads”，如图 2-1 所示。在打开的下载页中，根据自己的操作系统平台选择合适的 JDK 安装文件，如图 2-2 所示，64 位的 Windows 操作系统可选择文件名为“jdk-8u111-windows-x64.exe”的 JDK 下载安装。



图 2-1 选择 Java SE 下载

Java SE Development Kit 8u111		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.78 MB	jdk-8u111-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.73 MB	jdk-8u111-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.35 MB	jdk-8u111-linux-i586.rpm
Linux x86	175.04 MB	jdk-8u111-linux-i586.tar.gz
Linux x64	158.35 MB	jdk-8u111-linux-x64.rpm
Linux x64	173.04 MB	jdk-8u111-linux-x64.tar.gz
Mac OS X	227.39 MB	jdk-8u111-macosx-x64.dmg
Solaris SPARC 64-bit	131.92 MB	jdk-8u111-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	93.02 MB	jdk-8u111-solaris-sparcv9.tar.gz
Solaris x64	140.38 MB	jdk-8u111-solaris-x64.tar.Z
Solaris x64	96.82 MB	jdk-8u111-solaris-x64.tar.gz
Windows x86	189.22 MB	jdk-8u111-windows-i586.exe
Windows x64	194.64 MB	jdk-8u111-windows-x64.exe

图 2-2 选择适当的 JDK 安装文件

JDK 下载完成后,双击安装文件,程序会自动解压缩并开始安装过程,按照向导一步一步进行安装即可,安装过程中需记住安装的路径,在配置 JDK 时将会用到这个安装路径。如果安装过程中选择了 JDK 和 JRE 都安装,则安装完成后的目录路径类似 C:\Program Files\Java\jdk1.8.0_111 和 C:\Program Files\Java\jre1.8.0_111。

Step2: 配置 JDK

在 Windows 系统下,JDK 需要进行环境变量的配置。具体过程如下:

(1) 打开【控制面板】,打开【系统】,选择【高级系统设置】,在打开的对话框中选择【高级】标签页,然后单击【环境变量】按钮。

(2) 在【环境变量】的【系统变量】中新建一个名字为“java_home”的变量,配置其值为“C:\Program Files\Java\jdk1.8.0_111”,即 JDK 所在的路径,建了该变量后,别的变量可以使用 %java_home% 来引用它的值。

(3) 在【系统变量】中找到变量 classpath(若无 classpath,则新建之),配置其值为“.;%java_home%\lib\dt.jar;%java_home%\lib\tools.jar”。注意:变量 classpath 中的第一个值是英文句号“.”,表示当前目录。

(4) 在【系统变量】中找到变量 path(若无 path,则新建之),在 path 变量值的最后添加“%java_home%\bin”。注意:变量值的各个值之间要以英文分号“;”进行分隔。

Step3: 测试 JDK 配置成功与否

(1) 找到【Windows 系统】,选择【命令提示符】,打开命令提示符窗口。

(2) 在命令提示符窗口中键入“java”命令并回车,如果出现图 2-3 所示的 java 命令的用法说明就表明 JDK 的 path 配置成功。



图 2-3 JDK 的 path 配置成功

2.2 JDK+记事本开发 Java 程序

JDK 没有提供 Java 程序源代码的编写环境,因此程序的源代码编写需要在其他的文本编辑器中进行,常见的适合 Java 的文本编辑器有很多,如记事本、Editplus、UltraEdit 等。

使用记事本编辑第一个 Java 程序源代码,保存时后缀名为 .java。

例程 2-1 HelloWorld.java。

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World——欢迎进入 Java 世界!");
    }
}
```

注意:

Java 程序源代码语句中所涉及的各种标点符号全部应是英文状态下输入的标点符号。比如 "Hello World——欢迎进入 Java 世界!" 中的引号必须是英文状态下的引号,但是引在双引号内部的文字和符号中英文皆可。

将上述 Java 程序源文件 HelloWorld.java 保存到 D:\目录下。

1. 编译 Java 源程序

在命令提示符窗口,将当前工作目录切换到 D:\目录,键入 "javac HelloWorld.java" 命令并回车,若无报错信息,如图 2-4 所示,则说明:①JDK 的 classpath 配置成功;②HelloWorld.java 编译成功。

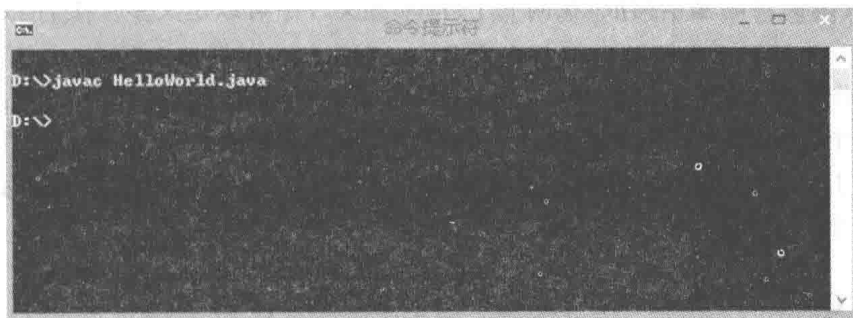


图 2-4 编译 HelloWorld.java

HelloWorld.java 编译成功在 D:\ 目录下会产生字节码文件 HelloWorld.class, .class 字节码文件是可以由 Java 虚拟机执行的文件。

2. 运行 Java 源程序

在命令提示符窗口, 键入“java HelloWorld”命令并回车, 观察运行结果, 如图 2-5 所示。

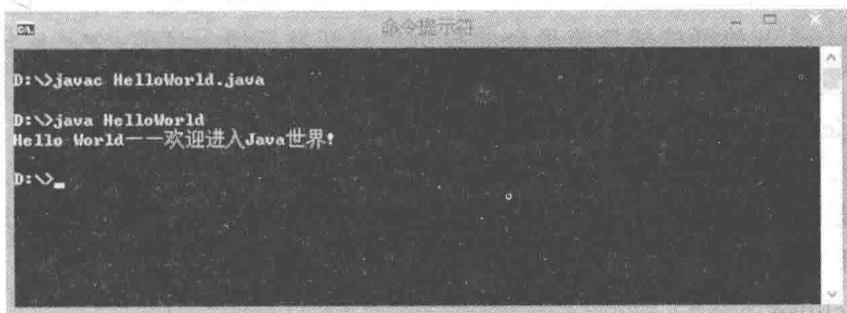


图 2-5 运行 HelloWorld

3. 源代码解析

```
public class HelloWorld { //1
    public static void main(String[] args) { //2
        System.out.println("Hello World——欢迎进入 Java 世界!"); //3
    } //2
} //1
```

将源代码中标示 1 的两行单独拿出来看:

```
public class HelloWorld {
}
```

这段代码定义了一个类 HelloWorld, 其中 HelloWorld 是类的名字, class 是定义类的关键字, public 指出这个类是公共类, 一对大括号 {} 括起来的是类体。其中只有 public 是可以省略的, 把源代码改成如下代码重新编译、运行得到的结果是一样的。

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World——欢迎进入 Java 世界!");
    }
}
```

类是 .java 源文件的基本构成,在同一个 .java 源文件中可以定义多个类,但是要注意以下几点:

- 在同一个 Java 源文件(即 .java 文件)中类的名字不能相同。
- 在同一个 Java 源文件中最多包含一个用 public 修饰的类。
- 在一个 Java 源文件中如果有 public 修饰的类,则 .java 文件的主文件名必须和 public 修饰的类的名字完全相同,包括大小写。



2.3 集成开发环境 Eclipse

记事本只是普通的文本编辑工具,不能帮助检查 Java 程序源代码中的错误,使用记事本这类的文本编辑工具编写程序源代码,开发效率会比较低。集成开发环境(IDE)是集应用程序源代码编辑、编译、调试、运行等功能为一体的开发环境,很好地提高了开发效率。

Eclipse 是一个开放可扩展且跨平台的自由集成开发环境。目前,它是比较流行的基于 Java 的集成开发环境之一。Eclipse 的功能强大,与其他功能相对固定的 IDE 软件相比更加灵活,这是因为 Eclipse 得到众多插件的支持。Eclipse 主要包括 Eclipse Platform、JDT、CDT 和 PDE 四个部分。其中,Eclipse Platform 提供通用的开发平台,JDT 支持 Java 开发,CDT 支持 C 开发,PDE 支持插件开发。Eclipse 提供了实时代码纠错功能,这样可以更快地找到代码中的错误,提高开发效率。

1. 下载 Eclipse

登录 Eclipse 官网(<http://www.eclipse.org>),可找到 Eclipse IDE for Java Developers 的最新版本的下载页,如 <https://eclipse.org/downloads/packages/eclipse-ide-java-developers/neon1a>,根据自己的操作系统平台选择合适的 Eclipse IDE 下载即可,如图 2-6 所示。



图 2-6 下载 Eclipse

下载的是 Eclipse IDE 的压缩包文件,无须安装,将压缩包解压缩到合适的目录下即可使用。例如解压缩至“C:\Program Files\eclipse”目录的话,双击该目录下的“eclipse.exe”即可启动 Eclipse 集成开发环境。

2. 配置 Eclipse 工作空间

启动 Eclipse 时,出现图 2-7 所示的工作空间设置对话框,可以修改默认的工作空间地址,