软件技术系列丛书

Java 程序设计基础

主编　单光庆

西南交通大学出版社
· 成　都 ·

内容简介

本书从 Java 程序设计初学者的角度，详细讲解了 Java 开发中用到的多种技术。全书共 12 章，在讲解 Java 开发环境的搭建及其运行机制、基本语法、数组、面向对象编程的思想时，采用通俗易懂的语言阐述抽象的概念，选用典型翔实的例子演示知识点的运用。在讲解多线程、常用 API、集合、IO、GUI、网络编程的章节中，通过剖析案例、分析代码含义、解决常见问题等方式进行阐述，并增添了许多辅助案例，帮助初学者培养良好的编程习惯。

本书附带案例源代码、习题、教学 PPT、教学实施案例、教学设计大纲等。

本书既可作为高等院校本、专科计算机相关专业的程序设计课程教材，又可作为 Java 技术基础的培训教材，同时也是一本适合广大计算机编程初学者学习的入门级参考教程。

前　言

一、关于本书

21 世纪挑战与机遇并存，没有足够的知识储备必将被时代抛弃。中国 IT 教育产业竞争日趋激烈，用户需求凸显个性，行业发展更需要理性。未来五年 IT 行业将以每年 18%的速度连续增长，将引发 IT 产业新的发展高潮。要实现信息产业大国的目标，应该依赖教育，要圆信息产业强国的梦想，依然要寄托于教育。IT 教育事业任重道远，其产业也正面临着机遇与挑战。我国的计算机教学长久以来一直重原理、轻应用。高等院校的计算机教学机制和教材对计算机本身的认识都存在误区。要改革高校计算机教学，教材改革是重要方面，用计算机教材的改革促进基础教育的改革势在必行。教材是教学过程的重要载体，一本好书，是人生前进的阶梯；一套好教材，是教学成功的基础。加强教材建设是推进人才培养模式改革的重要条件，也是推动中高职协调发展的基础性工程，对促进现代职业教育体系建设，切实提高职业教育人才培养质量具有十分重要的作用。

为什么学习 Java？找一份好工作？个人兴趣爱好？还是公司培训需要？不管大家的最终目的是什么，我们都有一个共同的基础目标：学好 Java。因为 Java 是一种纯面向对象程序设计语言。Java 具有支持网络应用编程、可跨平台使用、安全性好、支持多线程等特点，使它成为非常适合网络应用程序开发的一种程序设计语言。

在编写本书之前，作者已在高校从事了多年的“ Java 语言程序设计 ”“ 面向对象程序设计 ”“ C/C++程序设计 ”“ 数据库技术 ”等课程的教学及科研工作，对于该语言的概念、功能及应用有着较深入的理解和丰富的实践经验。由于 Java 语言的概念、功能及应用相对抽象，在教学过程中，我们发现已有的大部分教材注重文字理论描述，学生难以理解，不能很好地适应教学需求，故组织编写了这本教材。对抽象且难于理解的知识点，用代码实例进行验证，让学生先看到结果，再返回去理解原理。本教材在内容安排、教学深度、习题、实验及课程设计等方面均满足“ Java 程序设计 ”课程的教学要求。

本书以现代教育理念为指导，在讲授方式上注意结合应用开发实例，注重培养学生理解面向对象程序设计思想，以提高分析和解决实际问题的能力。

书中的所有程序源代码都经上机调试通过。

二、本书结构

全书共 12 章，可分为五个部分。具体结构如下：

第一部分为 Java 开发环境搭建，由第 1 章组成。

第 1 章：Java 开发入门。主要介绍 JDK 安装、Java 开发常用工具、Java 开发环境搭建。

第二部分为 Java 开发语言基础和数组，由第 2～3 章组成。

第 2 章：Java 编程基础。主要介绍 Java 开发中的常量、变量、数据类型、类型转换、流程控制语句结构等。

第 3 章：数组。主要介绍数组的定义、初始化等操作，分别针对一维数组、二维数组和多维数组进行案例演示。

第三部分为面向对象，由第 4~7 章组成

第 4 章：面向对象。主要介绍封装、方法、构造方法和方法的重载等。

第 5 章：类的继承、接口。主要介绍类的继承、接口、多态和异常等。

第 6 章：JAVA API。主要介绍 String 类和 StringBuffer 类、System 类和 Runtime 类、Math 类与 Random 类等。

第 7 章：框架与集合类。主要介绍 Collection 接口、List 接口、Set 接口、Map 接口等。

第四部分为 IO 输入输出与图形用户界面，由第 8~9 章组成。

第 8 章：IO 输入输出。主要介绍字节流、字符流和 File 类的常用操作。

第 9 章：GUI 图形用户界面。主要介绍 AWT 布局管理器、AWT 事件处理、AWT 绘图中的常用组件和常用操作。

第五部分为网络编程部分，由第 10~12 章组成。

第 10 章：JDBC。主要介绍对数据库的访问和连接。

第 11 章：多线程。主要介绍多线程的操作工作原理等。

第 12 章：网络编程。主要介绍网络通信与协议、网络编程中的应用。

三、本书特点

本书由易到难，层次结构清晰，实用性较强，强调理论与实践的结合，让读者动脑的同时动手，动手的同时动脑，从而得到真正质的飞跃。本书的主要特点如下：

（1）案例贯穿。以“问题启发”引入知识点，用针对知识点实际案例调试验证解决证明问题，以“知识点案例”式为主的上机调试教学。

（2）图文并茂。本书配备大量的图片，可读性强，能激发学生学习的兴趣，可供高职学生使用。

（3）方便教与学。每章都有案例源代码，并经上机调试通过，方便教师教授和学生学习。

（4）任务驱动。为了完成课程案例，设计了很多任务。通过任务驱动的方法，学生亲历真实任务的解决过程，在解决实际技术问题的过程中掌握相应的知识点，做到“做中学”。

（5）案例贴近生活。在案例的选取上力争贴近学生的生活，让学生有亲切感。

（6）完整的课程资源。提供教学课件、理论及上机源代码、教学例题及答案。

为了方便读者自学，编者尽可能详细地讲解 Java 环境搭建和各主要部分的内容，并附有大量的屏幕图例供读者学习参考，使读者有身临其境的感觉。

本书由重庆城市管理职业学院教师和行业、企业相关人员参与编写。具体分工如下：第 1~7 章、第 9 章由重庆城市管理职业学院单光庆编写，第 8 章和第 11 章由重庆城市管理职业学院朱儒明编写，第 10 章由重庆城市管理职业学院朱广福编写，第 12 章由重庆城市管理职业学院唐世毅编写。全书由单光庆策划和统稿。

四、适用对象

本书既可作为高职高专计算机专业和非计算机专业的数据库基础教材，又可供广大计算机爱好者自学使用。

由于编者水平有限，加之时间仓促，书中难免存在疏漏之处，希望广大读者多提宝贵意见。

编　者

2020 年 3 月

目　录

第 1 章　Java 开发入门

随着网络的发展和技术的改进，各种编程语言随之产生，Java 语言就是其中之一。Java 产生的时间并不长，其发展史要追溯到 1991 年，源于 James Gosling 领导的绿色计划。Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 面向对象程序设计语言和 Java 平台的总称。由 James Gosling 和同事们共同研发，并在 1995 年正式推出，2009 年 4 月被 oracle 公司收购。Java 可运行于多个平台，如 Windows, Mac OS，及其他多种 UNIX 版本的系统。Java 语言的诞生解决了网络程序的安全、健壮、平台无关、可移植等很多难题。本教程通过简单的实例将让大家更好地了解 JAVA 编程语言。

1.1　初识 Java 

在正式学习 Java 之前，先来了解几个关键性问题。那就是什么是 Java，为什么要学习 Java，Java 有哪些特点，Java 有哪些机制，如何来学习 Java 等几个问题。通过这几个问题让大家了解 Java 的一些内容，从而展开 Java 的学习。

1.1.1　Java 简介

1. Java 是什么

首先 Java 是一门计算机编程语言。Java 语言作为一种编程语言，它的语法规则与 C++很相似，但又避免了 C++中存在的弊端，因此有其自身的优点，如简单、面向对象、分布式、解释性、可靠、安全、可移植性、高性能、多线程、动态性等。所以说 Java 是一种解释性、跨平台、通用的编程语言。

Java 也是一种网络程序设计语言。Applet 程序编译器编译成的字节码文件，将被放在 WWW 网页中，并在 HTML 做出标记，只要是用户的主机安装了 Java 就可以直接运行 Applet。Java 比较适合网络环境，因此，成为 Internet 中最流行的编程语言之一。

如果有人认为 Java 只是一门语言的话，那就错了，Java 还是一种计算机语言开发平台。Sun 公司开发了 Java 语言之后，它已经从一门语言演化为一个计算机平台。Java 以其独特的优势，将给未来的网络世界带来巨大的变革。Java 具有“编写一次，到处运行”的特点，完全实现了不同系统之间的相互操作。Java 平台包括 Java 虚拟机和 Java 应用程序界面，其中虚拟机所写的是 JVM，Java 应用程序界面所写的是 Java API。Java 所有的开发都是基于 JVM 和 API 开发的，也就是基于 Java 平台。

2. 为什么要学习 Java

网络使得 Java 成为最流行的编程语言，反过来说 Java 也促进了网络的发展。Java 不但占据网络，而且涉及很多方面，包括桌面级的开发、网络开发和嵌入式开发等。在动态网站和

企业级开发中，Java 作为一种主流编程语言占到了很大份额。在嵌入式方面的发展更是迅速，现在流行的手机游戏，几乎都是应用 Java 语言开发的。可以说 Java 和人们的生活息息相关。目前 IT 行业 Java 技术人员短缺，而且 Java 涉及 IT 行业的各个方面及各个环节，所以说学习 Java 这门技术是从事 IT 职业很不错的选择。

3. Java 的特点

任何一种流行的东西都是有原因的。同样，Java 作为一门流行语言，也是有一定原因的。下面就来介绍一下 Java 有哪些特点，为什么它优于其他语言。

（1）Java 语言是简单的。

很多学习编程技术的人遇到的真正困难往往是编程语言的基础，例如 C 指针，甚至有些技术人员工作几年后还不能完全搞懂 C 指针是怎么回事。对于这个问题，Java 语言从设计之初就注意到了。Java 实际上是一个 C++去掉了复杂性之后的简化版。Java 丢弃了 C++中很少使用的、很难理解的、令人迷惑的那些特性，如操作符重载、多继承、自动的强制类型转换。特别地，Java 语言不使用指针，而是引用。如果读者没有编程经验，会发现 Java 并不难掌握，而如果读者有 C 语言或是 C++语言基础，则会觉得 Java 更简单，因为 Java 继承了 C 和 C++的大部分特性。Java 语言是一门非常容易入门的语言，但是需要注意的是，入门容易不代表真正精通容易。对 Java 语言的学习中还要多理解、多实践才能完全掌握。

（2）Java 语言是面向对象的。

虽然现在很多语言都号称是面向对象语言，但 Java 才是一门纯粹的面向对象语言，从设计之初就是按照面向对象语言设计的。面向对象是一个非常抽象的思想，在后面会有单独一篇来进行介绍。这里只需要知道 Java 面向对象的思想有三大特征：继承、多态和封装。Java 语言提供类、接口和继承等面向对象的特性。为了简单起见，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。总之，Java 语言是一个纯的面向对象程序设计语言。

（3）Java 语言是健壮性和自动内存管理的。

学过 C 或者 C++的人都知道，对内存操作时，都必须手动分配并且手动释放内存。如果将技术分为 10 个等级的话，8 个等级的人都是会犯没有释放内存的错误。没有释放内存在短期内是不容易被发现的，而且也不影响程序运行，但是长时间后就会造成内存的大量浪费，甚至造成系统崩溃。一门语言的健壮性就体现在它对常见错误的预防能力。Java 语言用的是自动内存管理机制，通过自动内存管理机制就可以自动地完成内存分配和释放的工作。Java 的强类型机制、异常处理、垃圾的自动收集等是 Java 程序健壮性的重要保证，对指针的丢弃是 Java 的明智选择。Java 的安全检查机制使得 Java 更具健壮性。

（4）Java 语言是安全的。

网络的发展给人们的生活带来了很多便捷之处，但也为一些不法分子提供了新的犯罪方式。目前网络中的黑客和病毒还没有从根本上得到根治，这就是由于开发的程序中存在漏洞，使用的编程语言安全性不高。Java 作为一种新出现的语言，对安全性上的考虑和设计，首先表现在 Java 是一门强类型语言，其中定义的每一个数据都有一个严格固定的数据类型；并且当数据间进行传递时，要进行数据类型匹配，任何不能匹配的结果都是会报错的。指针一直是黑客侵犯内存的重要手段，在 Java 中，对指针进行了屏蔽，从而不能直接对内存进行操作，

进而大大提高了内存的安全性。由于 Java 通常被用在网络环境中，为此，Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外，Java 对通过网络下载的类具有一个安全防范机制（类 Class Loader），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 Security Manager）让 Java 应用设置安全哨兵。

（5）Java 语言是跨平台的。

随着硬件和操作系统越来越多样化，编程语言的跨平台性越来越重要。一门语言的跨平台性的优劣体现在该语言程序跨平台运行时修改代码的工作量。Java 是一门完全的跨平台语言，它的程序跨平台运行时，对程序本身不需要进行任何修改，真正做到“一次编写，到处运行”。

（6）Java 语言是体系结构中立的。

Java 程序（后缀为 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（后缀为 class 的文件），然后可以在实现这个 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 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，然后可以在实现这个 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，它必须由 Thread 类或其子（孙）类来创建。通常有两种方法来创建线程：其一，使用型构为 Thread(Runnable)的构造子将一个实现了 Runnable 接口的对象包装成一个线程；其二，从 Thread 类派生出子类并重写 run 方法，使用该子类创建的对象即为线程。值得注意的是 Thread 类已经实现了 Runnable 接口，因此，任何一个线程均有它的 run 方法，而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。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（关键字为 synchronized）。

（11）Java 语言是动态的。

Java 语言的设计目标之一是适应于动态变化的环境。Java 程序需要的类能够动态地被载入运行环境，也可以通过网络来载入所需要的类。这也有利于软件的升级。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。

（12）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（Javanet），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。

1.1.2 Java 发展历史

1990 年 Sun 公司启动 James Gosling 领导的绿色计划, 1992 年创建 Oak 语言 —— Java, 1994 年 Gosling 参加硅谷大会演示 Java 功能，震惊世界。1995 年 Sun 正式发布 Java 第一个版本, 目前最新是 jdk13.0。

- 1995 年 5 月 23 日，Java 语言诞生。
- 1996 年 1 月，第一个 JDK-JDK1.0 诞生。
- 1996 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。
- 1996 年 9 月，约 8.3 万个网页应用了 Java 技术来制作。
- 1997 年 2 月 18 日，JDK1.1 发布。
- 1997 年 4 月 2 日，JavaOne 会议召开，参与者逾一万人，创当时全球同类会议规模之纪录。
- 1997 年 9 月，JavaDeveloperConnection 社区成员超过十万。
- 1998 年 2 月，JDK1.1 被下载超过 2 000 000 次。
- 1998 年 12 月 8 日，Java2 企业平台 J2EE 发布。
- 1999 年 6 月，Sun 公司发布 Java 的三个版本：标准版（JavaSE，以前是 J2SE）、企业版（JavaEE 以前是 J2EE）和微型版（JavaME，以前是 J2ME）。
- 2000 年 5 月 8 日，JDK1.3 发布。
- 2000 年 5 月 29 日，JDK1.4 发布。
- 2001 年 6 月 5 日，Nokia 宣布，到 2003 年将出售 1 亿部支持 Java 的手机。
- 2001 年 9 月 24 日，J2EE1.3 发布。
- 2002 年 2 月 26 日，J2SE1.4 发布，自此 Java 的计算能力有了大幅提升。
- 2004 年 9 月 30 日，J2SE1.5 发布，成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性，J2SE1.5 更名为 JavaSE 5.0。
- 2005 年 6 月，JavaOne 大会召开，Sun 公司公开 JavaSE 6。此时，Java 的各种版本已经更名, 以取消其中的数字“ 2 ”:J2EE 更名为 JavaEE, J2SE 更名为 JavaSE, J2ME 更名为 JavaME。
- 2006 年 12 月，SUN 公司发布 JRE6.0。
- 2009 年 04 月 20 日，甲骨文 74 亿美元收购 Sun，取得 Java 的版权。
- 2010 年 11 月，由于甲骨文对于 Java 社区的不友善，因此 Apache 扬言将退出 JCP。
- 2011 年 7 月 28 日，甲骨文发布 Java7.0 的正式版。
- 2014 年 3 月 18 日，Oracle 公司发表 JavaSE 8。
- 2016 年 9 月，JavaSE 9 Oracle 宣布发布。
- 2018 年 3 月，Java10 发布。
- 2018 年 9 月，Java11 发布。
- 2019 年 3 月，Java12 发布。

1.1.3 Java 平台简介

1. Java 平台

Java 编程可以分成三个方向：

（1）JavaSE（J2SE）（Java2 Platform Standard Edition，Java 平台标准版）桌面开发：Java 基础中的基础。

（2）JavaEE（J2EE）（Java2 Platform Enterprise Edition，Java 平台企业版）：用于 Web 开发。

（3）JavaME（J2ME）（Java2 Platform Micro Edition，Java 平台微型版）：用于小型电子设备上的软件开发。

2005 年 6 月，JavaOne 大会召开，Sun 公司公开 JavaSE 6。此时，Java 的各种版本已经更名，以取消其中的数字“ 2 ”：J2EE 更名为 JavaEE，J2SE 更名为 JavaSE，J2ME 更名为 JavaME。

Java 程序需要在虚拟机上才可以运行，换言之，只要有虚拟机的系统都可以运行 Java 程序。不同系统上要安装对应的虚拟机才可以运行 Java 程序。

2. 开发步骤

（1）编写源文件（源代码）（.java）。

（2）编译器编译源文件为类文件（.class），可用 J2SE 或 J2EE 编译。

（3）在虚拟机（JVM）上运行。

运行 Java 程序之前要先安装和配置 JDK。

3. JDK 是什么？

（1）JDK 全称为 Java Development Kit；中文：Java 开发工具包。

（2）JDK 是 Sun 公司开发的。

（3）JDK 包括 JRE（Java Runtime Environment）Java 运行环境、Java 工具和 Java 基础的类库（类共 3600 个左右，常用类在 150 个左右），JDK 可以在 Oracle 的官方网站（http ://www.oracle.com/technetwork/java/index.html）中下载。

4. JDK、JRE、JVM 的作用及关系

（1）作用。

- JVM：保证 Java 语言跨平台；
- JRE：Java 程序的运行环境；
- JDK：Java 程序的开发环境。

（2）关系。

- JDK：JRE+工具；
- JRE：JVM+类库。

1.2 Java 开发环境配置

1.2.1 Window 操作系统下安装 Java 步骤

1. 下载 JDK

首先，需要下载 Java 开发工具包 JDK，下载地址 :http ://www.oracle.com/technetwork/java/javase/downloads/index.html，单击图 1.1 所示的下载按钮。

在下载页面中选择 “ 接受许可 ”，并根据自己的系统选择对应的版本。本文以 Window 64

位系统为例，如图 1.2 所示。

下载后根据提示进行 JDK 的安装。在安装 JDK 时也会安装 JRE，一并安装即可。安装过程中可以自定义安装目录等信息，例如选择安装目录为 C:\Program Files（x86）\Java\jdk1.8.0_91。

图 1.1　Java 下载

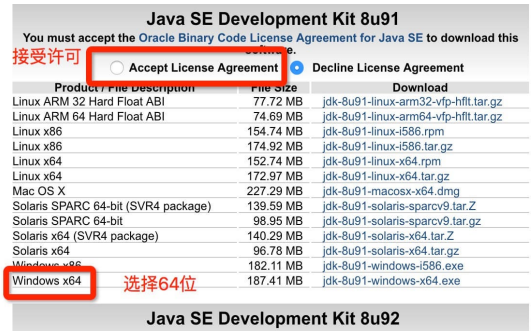

图 1.2　匹配 Window 64 位的 JDK

2. 安装 JDK

下载 JDK 后，双击下载的 EXE 文件，即可开始安装 JDK。首先是弹出许可证协议窗口，其中给出了 Sun 公司的一些开发协议，单击其中的“ 接受 ”按钮，就会弹出如图 1.3 所示的“ 自定义安装 ” 窗口。

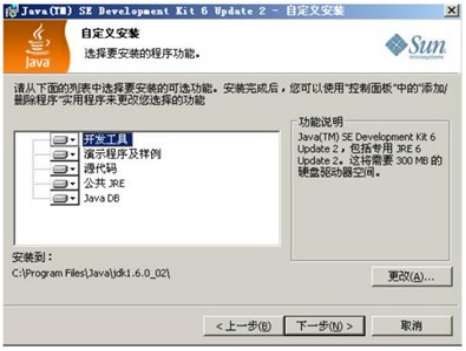

图 1.3　“ 自定义安装 ” 窗口

在窗口中可以选择要安装的 Java 组件和 JDK 文件的安装路径。这里可以采用默认安装 Java 的所有组件并在 C 盘安装。也可以自定义安装目录等信息，例如我们选择安装目录为 C:\Program Files (x86)\Java\jdk1.8.0_91。在后面的配置中，也将按照默认安装进行配置。单击“下一步”按钮后就开始安装 JDK，稍后，单击窗口中的“完成”按钮，就正式完成了 JDK 的安装。

3. 配置环境变量

- path 环境变量：存放可执行文件的存放路径，路径之间用逗号隔开。
- classpath 环境变量：类的运行路径，JVM 在运行时通过 classpath 加载需要的类。

方法：

（1）安装完成后，右击“我的电脑”，单击“属性”，选择“高级系统设置”，如图 1.4 所示。

图 1.4 “高级系统设置”窗口

（2）选择“高级”选项卡，单击“环境变量”，如图 1.5 所示，就会出现如图 1.6 所示的画面。

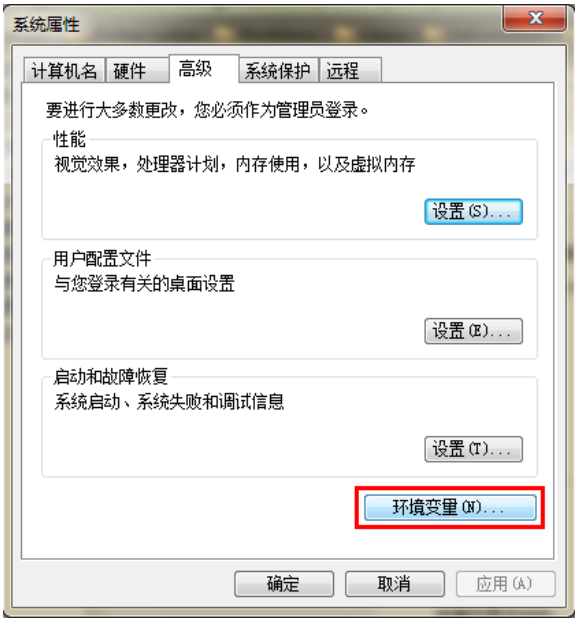

图 1.5 “环境变量”按钮

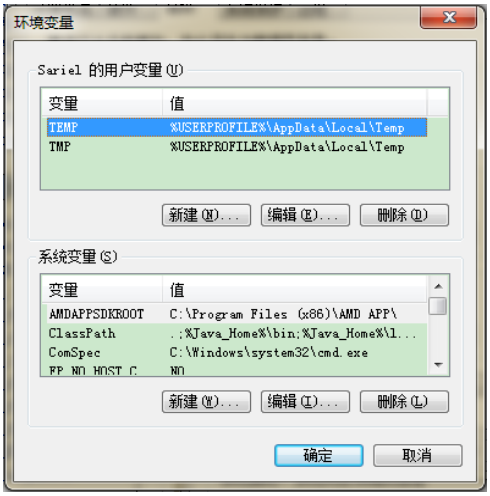

图 1.6 “ 环境变量 ” 窗口

（3）在“ 系统变量 ”中设置 3 项属性：JAVA_HOME，PATH，CLASSPATH（大小写都可）。若已存在则单击“ 编辑 ”；若不存在则单击“ 新建 ”。

变量设置参数如下：

① 变量名：JAVA_HOME。

变量值：C：\Program Files（x86）\Java\jdk1.8.0_91　　// 根据自己的实际路径配置，如图 1.7、1.8 所示。

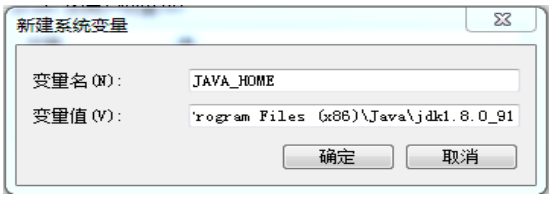

图 1.7　新建变量名 JAVA_HOME 与对应的变量值设置

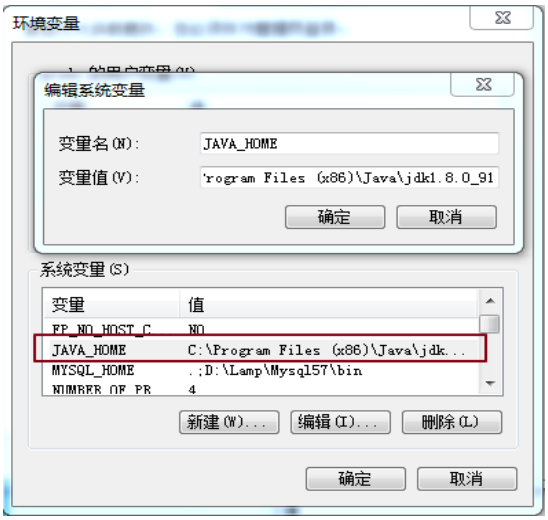

图 1.8　编辑变量名 JAVA_HOME 与对应的变量值

② 变量名：CLASSPATH。

变量值：.；%JAVA_HOME%\lib\dt.jar；%JAVA_HOME%\lib\tools.jar；　　//注意前面有个“ . ”，如图 1.9 所示。

图 1.9　变量名 CLASSPATH 与对应变量值

③ 变量名：Path。

变量值：%JAVA_HOME%\bin；%JAVA_HOME%\jre\bin；如图 1.10、1.11 所示。

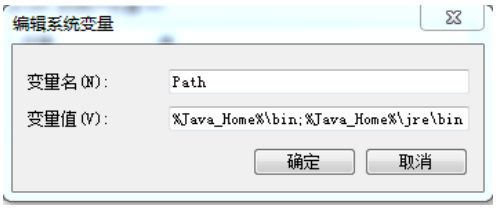

图 1.10　变量名 Path 与对应变量值设置

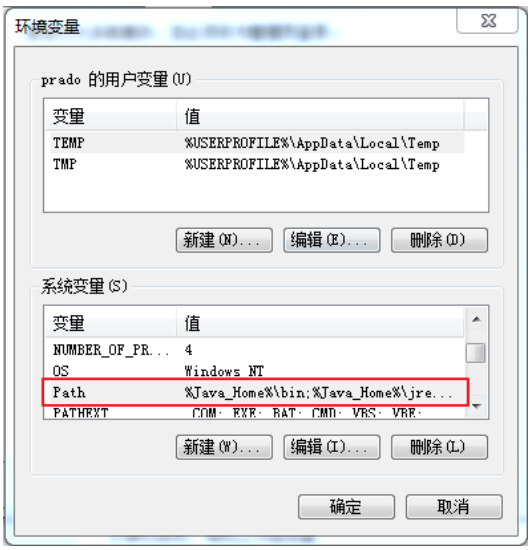

图 1.11　变量名 Path 与对应变量值

这是 Java 的环境配置。配置完成后，可以启动 Eclipse 来编写代码，它会自动完成 Java 环境的配置。

注意：如果使用 1.5 以上版本的 JDK，不用设置 CLASSPATH 环境变量，也可以正常编译和运行 Java 程序。

4. 测试 JDK 是否安装成功

重点掌握两个程序：

• javac.exe：Java 编译器工具，可以将编写好的 Java 文件（.java）编译成 Java 字节码文件（.class）。

• java.exe：Java 运行工具，启动 Java 虚拟机进程，运行编译器生成的字节码文件（.class）。

方法：

（1）单击“ 开始 ”→“ 运行 ”（快捷键：Win+R），键入“ cmd ”，单击“ 确定 ”，如图 1.12 所示。

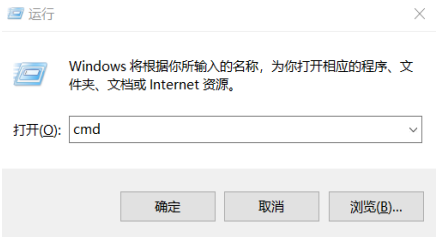

图 1.12　cmd 窗口

（2）查看当前 Java 运行的版本：可以使用 -version 参数来查看当前 Java 的运行版本，命令如下：

```
java–version
```

执行结果如图 1.13 所示。

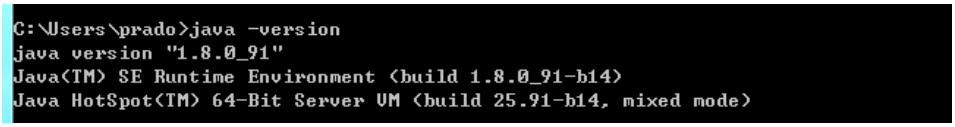

图 1.13　查看 Java 版本号

再键入“ javac ”“ java ”等几个命令，若无出现错误信息，则说明环境变量配置成功，如图 1.14 所示。

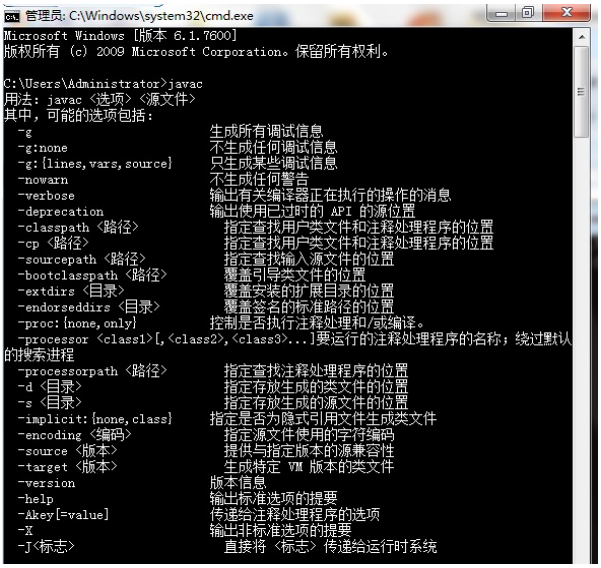

图 1.14　环境变量配置成功

1.2.2 编写 Java 程序

安装好以上的工具，并配置好环境变量后，下面就可以规范 Java 中编写代码的格式。

【例 1.1】第一个 Java 程序，通过一个简单的实例来展示 Java 编程与运行过程，在 F 盘下创建文件 HelloWorld.Java(文件名需与类名一致)，在屏幕上显示 Hello World 代码字样。

1. 在记事本中编写程序 HelloWorld. Java 代码

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

输入代码后，不要忘记保存，且一定要将文件扩展名改为.java，否则不能编译。在有些计算机中，默认是没有显示扩展名的，所以要首先将文件扩展名显示出来。具体操作方法：

双击“我的电脑”图标，选择菜单栏中的“工具”→“文件夹选项”→“查看”命令，如图 1.15 所示。

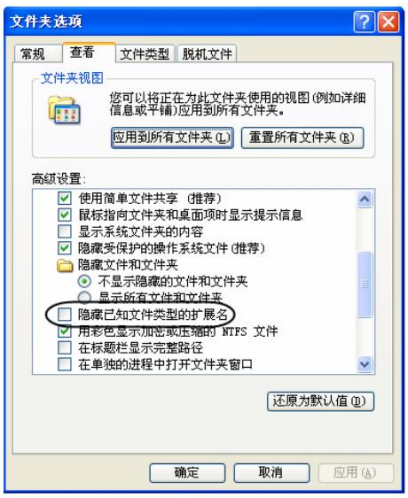

图 1.15　扩展名设置

在图 1.15 中取消“隐藏已知文件类型的扩展名”复选框的勾选，单击“确定”按钮，这样系统中的所有文件就都具有扩展名了。

2. 编译 HelloWorld. Java 程序

编写并保存 Java 程序后，选择“开始”→“运行”命令，在“运行”命令框中输入“cmd”，弹出命令提示符界面。首先输入“f:”命令，这样就切换到 F 盘下，这是因为上一节开发的程序保存在 F 盘中。(如果读者编写的程序不在 F 盘中，这里就输入所编写程序所在的位置。)进入 F 盘后，输入“javac HelloWorld.java”命令，其中 javac 是 JDK 中的编译命令，而 HelloWorld.java 是上一节中编写的 Java 程序的文件名。执行“javac HelloWorld.java”命令后，会在 F 盘下产生一个名称为 HelloWorld.class 的文件，它是执行编译命令所产生的文件。操作方式如图 1.16 所示。

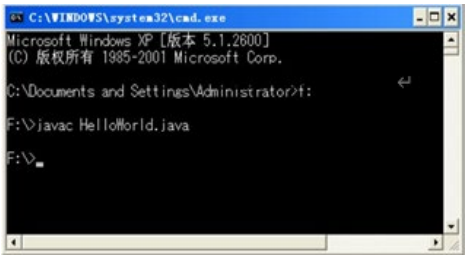

图 1.16　编译 Java 程序

注意：在 javac 命令后输入的文件名中一定要有.java 扩展名，否则会发生错误。

3. 运行 Java 程序

编译 Java 程序后，产生一个以.class 为扩展名的文件，运行 Java 程序就是运行该文件。在图 1.16 所示界面的命令输入下继续输入“ javaHelloWorld ”命令，如图 1.17 所示。

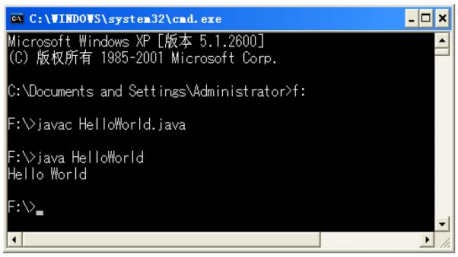

图 1.17　运行 Java 程序

从运行结果中可以看到输出了“ Hello World ”信息，这就是开发的该程序的功能。运行 Java 程序是通过 Java 命令来完成的。

注意：在 Java 命令后输入的文件名没有扩展名，如果有，则会发生错误。

接下来简单地讲解一下前面开发的 HelloWorld 程序，让读者对 Java 程序先有一个初步了解。

HelloWorld 程序中的第一行的内容是“ public class HelloWorld ”，其中“ HelloWorld ”是一个类名，“ class ”是判断“ HelloWorld ”为一个类名的关键字，而“ public ”是用来修饰类的修饰符。每一个基础类都有一个类体，使用大括号包括起来。

程序中的第二行为“ public static void main(String args[]) ”，它是一个特殊方法，主体是“ main ”，其他的都是修饰内容。这条代码语句是一个 Java 类固定的内容，其中 main 定义一个 Java 程序的入口。和类具有类体，方法具有方法体一样，其同样也要使用大括号括起来。

程序的第三行为“ System.out.println("Hello World"); ”，该语句的功能是向输出台输出内容。在该程序中输入的是“ Hello World ”信息，从而才有了图 1.15 的运行结果。

1.3　Eclipse 的安装与启动

在前面讲解了使用记事本来开发 Java 程序，因为要调用命令提示符界面，所以显得有些麻烦。而 Java 的一些集成开发工具解决了这一问题。目前 Java 的集成开发工具有很多，这里采用开发中最常用 Eclipse 来进行讲解。

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。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具（JavaDevelopment Tools，JDT）。安装 Eclipse 前需要安装 JDK，关于 JDK 的安装和配置参见前述内容。从 Eclipse 的官方网站：(http://www.eclipse.org/）下载最新版本的 Eclipse。

1. Eclipse 的下载

（1）登录其官方网站：www.eclipse.org。图 1.18 所示为 Eclipse 官方网站的首页。

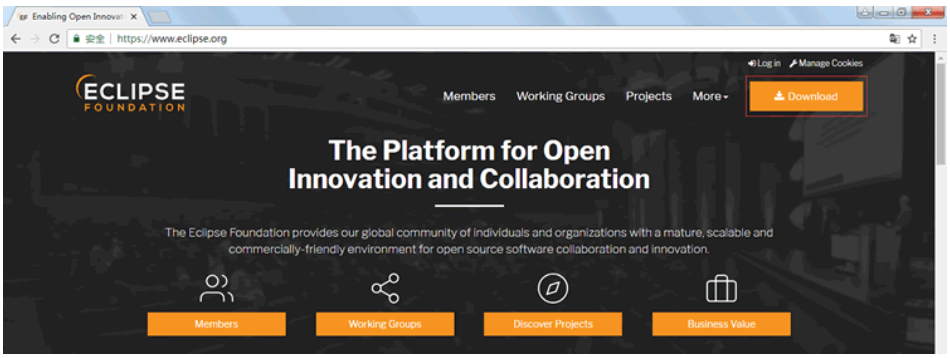

图 1.18　Eclipse 官网首页

（2）从首页中单击“ Download ”按钮，进入图 1.19 所示的页面。

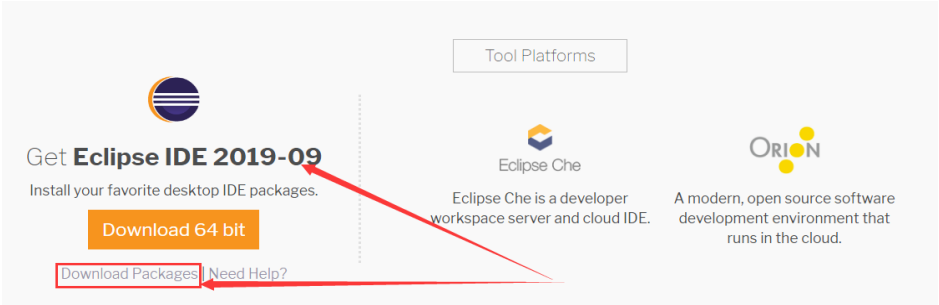

图 1.19　跳转页面

（3）单击“ Download Packages ”，进入 Eclipse 下载页面。从“ Eclipse IDE for Enterprise Java Developers ”后面选择适合当前系统的版本，这里单击 64 bit 按钮，下载 64 位的安装包，如图 1.20 所示。

图 1.20　下载 Eclipse

下载完成后会得到一个名为“ eclipse_jee_2019_09_win32_x86_64.zip ”的压缩文件。虽然 Eclipse 本身是用 Java 语言编写，但下载的压缩包中并不包含 Java 运行环境（即安装 Eclipse，应首先安装 JDK），需要用户自己另行安装 JRE，并且要在操作系统的环境变量中指明 JRE 中 bin 的路径。

（4）Eclipse 的安装非常简单，只需将下载的压缩包进行解压，然后双击“ eclipse.exe ”文件即可。

2. 启动 Eclipse

由于 Eclipse 是一个开源项目，因此，所有社区和开发者都可以为 Eclipse 开发扩展功能。

下载和安装 Eclipse 后，就可以启动 Eclipse。在 Eclipse 文件下有一个 eclipse.exe 文件，双击该文件，就可以启动 Eclipse。第一次启动 Eclipse 时，会要求用户选择一个工作空间（Workspace），会出现如图 1.21 所示的窗口。

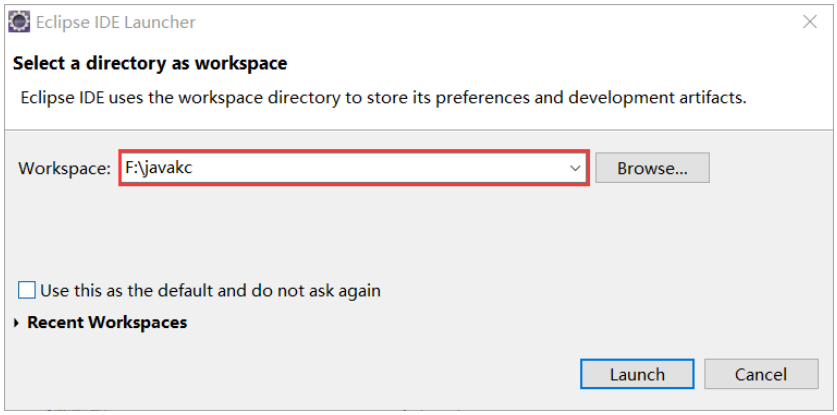

图 1.21　Eclipse 程序路径

窗口中“ 工作空间 ”文本框显示的通常是 C 盘下的位置，读者也可以进行修改来确定通过 Eclipse 开发的项目和程序保存的位置。单击“ 确定 ”按钮后，弹出如图 1.22 所示的 Eclipse 欢迎窗口。

图 1.22　Eclipse 欢迎窗口

关闭欢迎窗口后，就会弹出真正用于开发的窗口。

在 Eclipse 集成开发工具开发 HelloWorld 程序的步骤分为：新建 Java 项目、新建 Java 类、编写 Java 代码、运行程序。

（1）选择菜单栏中“ 文件 ”→“ 新建 ”→“ 项目 ”命令，弹出如图 1.23 所示的“ 新建项目 ”窗口。

（2）选择“ Java 项目 ”选项，单击“ 下一步 ”按钮，弹出如图 1.24 所示的“ 新建 Java 项目 ”窗口。

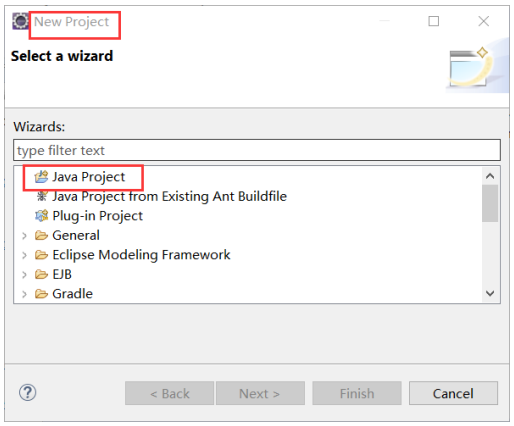

图 1.23　“ 新建项目 ”窗口

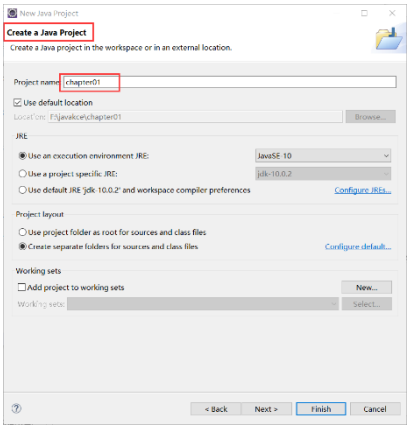

图 1.24　“ 新建 Java 项目 ”窗口

（3）在“ 新建 Java 项目 ”窗口的“ 项目名 ”文本框中输入自己要创建的项目名。由于这里是本书的第一章，就设置创建的项目名为“ chap1 ”，单击“ 完成 ”按钮，这样就创建了一个名称为“ chapter01 ”的 Java 项目，此时就会在 Eclipse 开发窗口的项目结构区显示该项目。

（4）在“ chapter01 ”项目上右击，选择“ 新建 ”→“ 类 ”命令，弹出如图 1.25 所示的“ 新建 Java 类 ”窗口。

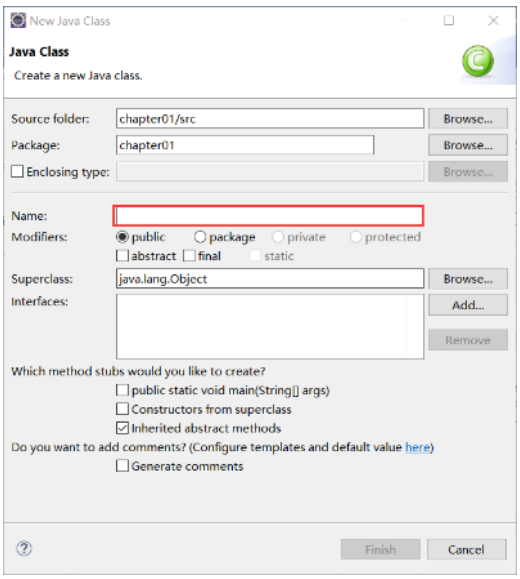

图 1.25　“ 新建 Java 类 ”窗口

在“新建 Java 类”窗口中有很多需要填写的选项。首先是填写包，包的概念会在后面进行讲解，如果这里不填，则采用默认值，也就是不使用包。下面需要填写的就是 Java 类的名称，在名称文本框输入“HelloWorld”。最后在“想要创建哪些方法存根”中勾选第一个复选框，也就是 main 方法，因为它是一个类的入口。设置好这些选项后，单击“完成”按钮，在编码区输入如下代码。

```
//日期：2019 年 10 月 23 日
//功能：在控制台显示" Hello World!"
//作者：光庆
publicclassHelloWorld{
publicstaticvoidmain(String[]args){
System.out.println("Hello World！欢迎来重庆城市管理职业学院！");
}
}
```

注意：由于 Eclipse 自动生成一些注释和空格，为了少占篇幅和方便学习，在后面的代码中会将这些东西去掉，然后加上一些更易懂的注释。如果发现自己开发的代码和书中的不太一样，也不要奇怪。

(5) 完成 Java 程序的编写后，就可以编译和运行该 Java 程序。在 Eclipse 集成开发工具中，编译和运行是一体的，不需要分别执行。选择菜单栏中的“运行”→“运行方式”→“Java 运行程序”命令，选择要运行的 HelloWorld.java 程序，单击“确定”按钮，即可运行 Java 程序。

说明：并不是每一次运行 Java 程序都是这么复杂的。第一次这样运行后，如果后面代码被修改需要再次运行，可以直接选择“运行”→“运行上次启动”命令来运行该程序。也可以通过单击 Eclipse 工具栏中的  按钮运行程序。

如在 Eclipse 中，执行结果如图 1.26 所示。

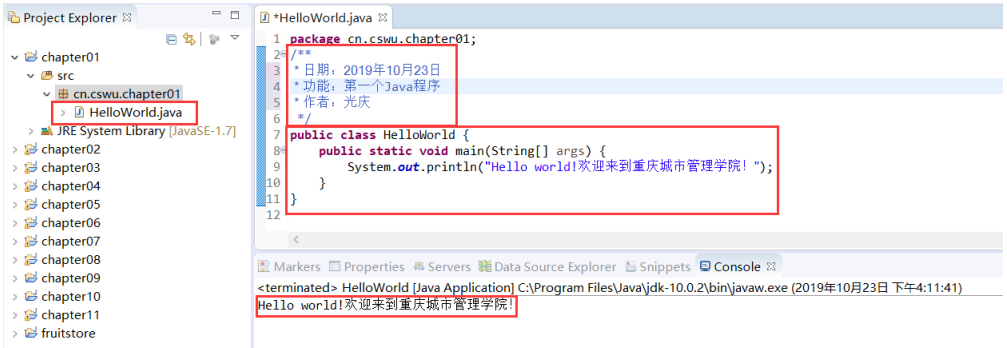

图 1.26　在 Eclipse 工具中输出语句加上单位名

从运行结果中可以看到输出了“Hello World!欢迎来重庆城市管理职业学院！”信息，从而完成了该程序的开发。如果没有出现如图 1.24 所示的运行结果，读者就需要认真查一下什么地方出了问题，或者重新编写开发。

注：

public static void main（String args []）{

}都是 Java 一切程序运行的入口。

1.4 本章小结

本章初步介绍了 Java 程序开发的相关知识和过程。首先简单地讲解了读者最关心的几个问题，并没有过多地讲解 Java 发展和起源等内容。接着讲解了 Java 开发环境的搭建，以及如何使用该开发环境进行 Java 程序开发。最后讲解了 Eclipse 这一集成开发工具的基本功能及如何在集成工具中进行程序开发。