普通高等教育“十三五”规划教材

Java 程序设计教程

Java CHENGXU SHEJI JIAOCHENG

赵 辉 郑山红 王 璐 编 著

中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十三五”规划教材

Java 程序设计教程

赵 辉 郑山红 王 璐 编 著

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书全面、系统地介绍了 Java 语言的基本概念、基本语法、程序设计方法以及一些企业级应用技术。全书共 12 章，第 1 章～第 2 章为预备与入门知识，介绍了 Java 语言相关的基本概念，描述了 Java 程序开发的一般过程和编译、运行环境，以及 Java 编程的基础知识；第 3 章～第 4 章为面向对象的编程知识，介绍了面向对象程序设计的基本概念、基本原理和基本特点，Java 语言的类、对象、接口和包的设计规范，以及封装、继承、多态等设计机制；第 5 章～第 9 章为 Java 编程的常用知识，介绍了常用实用类、泛型与集合框架、图形用户接口、输入输出处理、数据库编程等具体应用；第 10 章～第 12 章为 Java 编程的进阶知识，介绍了 Java 多线程、Java 网络编程以及 Java 反射机制、代理机制等高级编程知识，为开发企业级应用软件打下基础。

本书构思新颖、示例丰富，内容循序渐进、前后呼应，既重视基本理论和基本概念的阐述，又注重程序设计能力的培养，同时反映了 Java 语言的最新发展。

本书可以作为高等院校计算机及相关专业的学生学习“Java 程序设计”课程的教材，也可以作为广大工程技术人员和程序设计爱好者的自学教材。

本书提供有电子教案、源程序文件、习题答案，方便读者使用。读者可以从中国水利水电出版社网站和万水书苑上免费下载，网址为：http://www.waterpub.com.cn/softdown/和 http://www.wsbookshow.com。

图书在版编目（CIP）数据

Java程序设计教程 / 赵辉，郑山红，王璐编著. --
北京 : 中国水利水电出版社，2016.2
普通高等教育“十三五”规划教材
ISBN 978-7-5170-4047-7

Ⅰ. ①J… Ⅱ. ①赵… ②郑… ③王… 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(2016)第020072号

策划编辑：崔新勃　　责任编辑：李　炎　　加工编辑：封　裕　　封面设计：李　佳

书　　名	普通高等教育“十三五”规划教材 Java 程序设计教程
作　　者	赵　辉　郑山红　王　璐　编　著
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（发行部）、82562819（万水）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 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	三河市铭浩彩色印装有限公司
规　　格	184mm×260mm　16 开本　16.75 印张　410 千字
版　　次	2016 年 2 月第 1 版　2016 年 2 月第 1 次印刷
印　　数	0001—2000 册
定　　价	35.00 元

前　言

Java 语言是一种革命性的编程语言，具有简单、平台无关、面向对象、分布式、安全性、可移植、多线程以及强有力的网络支持等特点，因此，Java 已经成为编写各类应用程序，如安全的网络程序、图像处理、多媒体、Wed 客户机和服务器以及关键性任务的企业级系统的首选语言工具，并且“Java 程序设计”课程已成为高校计算机类专业的一门重要专业课。

本书是在作者多年从事 Java 程序设计教学改革与实践的基础上编写而成，主要具有以下五方面特色。

1．注重内容的合理选择与更新

Internet 网络技术的飞速发展以及软件开发模型的层次化趋势使得 Java 语言在不断发展、更新。本书在内容选取时除了包含 Java 语言最基本的知识外，还适当选择了一些新的、成熟的知识（例如：多线程、网络编程、反射机制、Annotation、代理机制等），体现了 Java 语言的发展、变化，保证了知识的先进性。

2．注重面向对象程序设计能力的培养

面向对象技术被认为是程序设计方法学的一场革命，是现代软件开发的主流方法，Java 语言是面向对象技术应用的最成功范例。本书以 Java 语言为载体，在介绍 Java 编程的同时讲解面向对象程序设计的基本原理和方法，将面向对象的思维方法贯穿于全书并加以强调，为初学者奠定扎实的面向对象程序设计基础，使其树立良好的编程思想。

3．注重软件工程素质和能力的培养

注重学生编程习惯的培养，使学生能够站在现代软件开发和软件工程这个比较开阔的层面上学习程序设计，而不是局限于繁琐的程序设计语言规则上。为此全书贯穿了软件工程的思想，强调“自顶向下、逐步求精”“先分析后设计再编码”和“以需求为驱动”等软件工程方法的应用。

4．注重知识面的拓展与学习兴趣的激发

相比较而言，教材内容毕竟是有限的，学生要完全掌握 Java 语言和程序设计的精髓，还需要学习很多教材之外的知识，为此本书在正文中和正文后分别设置了“注意”和“拓展”等栏目，介绍一些相关的历史典故、发展动向、研究热点以及技术方法等知识，并指明学习的途径，以便对学生的进一步学习加以引导，从而开拓学生的知识面，激发学生学习的兴趣。

5. 注重教材的完整性

本书提供了电子教案、可执行的源程序文件和习题答案等电子资料，读者可以从出版社相关网站下载。

本书由赵辉、郑山红、王璐编著，陈满林、王国春、彭馨仪参与了部分章节的编写工作。

尽管书稿几经修改，但由于编者水平有限，书中难免有许多疏漏之处，敬请各位同行和广大读者批评指正。

编　者

2015 年 10 月

目　录

第 1 章　Java 语言概述

本章目标：

- 了解 Java 的特点
- 理解 Java 的平台无关性
- 掌握 JDK 的安装和环境变量的配置
- 掌握 Java 程序的开发流程

Java 语言是一种功能强大的面向对象程序设计语言。多年来，Java 语言以其独有的特点及与网络的紧密结合，成为当今应用最广泛的计算机编程语言。本章主要介绍 Java 语言的产生和发展过程、Java 语言的特点和优越性、Java 的体系结构、Java 程序的开发环境及其基本结构和开发流程。

1.1　Java 语言发展历史

Java 的历史可以追溯到 1991 年，当时 Sun Microsystem 公司的 James Gosling 领导的一个叫 Green 的项目组试图为消费类的电子产品（例如电视、冰箱等）开发一种交互式的软件系统，由于该类产品的内存小，并且不同厂商的产品可能选用不同的中央处理器，因此，这种应用需要的程序设计语言应该尽可能小巧，并且具有良好的移植性和实用性。开始时，项目组采用 C++语言编写程序，但是在开发中遇到了一系列难以克服的困难使他们最终不得不放弃，于是决定基于 C++语言开发一种新的编程语言。这种新型语言最初被命名为 Oak 语言，但是，Sun 公司很快发现当时已经存在了一种名为“Oak”的程序设计语言，后来开发小组在一次聚会中，从咖啡获得灵感想到了“Java”这个名字，于是它被正式命名为 Java 语言。

注意：印度尼西亚有一个重要的盛产咖啡的岛屿叫 Java，中文译名为爪哇，开发人员为这种新的语言起名为 Java，其寓意是为世人端上一杯热咖啡。

由于市场的因素，Green 项目组的交互式软件系统并没有在消费类电子产品市场中取得成功，Sun 公司放弃了对该项目的继续开发和研制。但随后由于互联网的迅速发展，Sun 公司看到了 Java 语言在 Internet 上的广阔前景，于是通过改造于 1995 年 5 月 23 日正式推出了 JDK（Java Development Kit，Java 开发工具集）1.0 版本，几个月后又推出了 JDK1.02 版本。但是，这些版本并不适合进行正规的应用软件开发，缺乏对很多功能的支持，例如不支持打印功能。它的后继者 JDK1.1 版本很快弥补了前者的不足，增加了许多 Java 类库，基本上实现了 Java 的一些重要特性，但是仍然具有很大的局限性，例如对文本的编辑限制。

1998 年 12 月，JDK1.2 版本正式发布，改进了原来的图形用户接口和图形工具包，使它的功能更加全面。为了表示与以前版本的明显不同，Sun 公司将其称为 J2SE1.2（Java 2 平台标准版，Java 2 Standard Edition）。2004 年 9 月发布了 J2SE1.5 后，又将其改为 Java SE5.0。值得一提的是，2006 年发布 Java SE6.0 后，由于不断的亏损和决策失误，Sun 公司的发展举步维

艰。2009 年 4 月，Oracle 公司宣布以 74 亿美元收购 Sun 公司，一个伟大的公司就这样走到了生命的尽头，但是，Sun 被 Oracle 收购并不代表着 Java 的毁灭，而是继续由 Oracle 推动 Java 向前发展。2011 年 7 月 Oracle 公司发布了 Java SE7.0，这是 Java 目前的最高版本。

自从 Java 语言产生以来，Java 技术得到了迅猛发展和广泛应用。由于提供了强大的图形、图像、动画、音频、视频、多线程、跨平台以及网络交互能力，Java 语言在实时交互式系统、网络应用系统以及嵌入式系统等方面大显身手，成为当今推广速度最快、应用最广泛的计算机编程语言。在TIOBE世界编程语言排行榜中，从 2002 年至今，Java 始终位于前两位，占有相当高的市场份额。

1.2 Java 语言的特点

Java 语言是一种跨平台，适合于分布式计算环境的面向对象的编程语言。具体来说，它具有简单、面向对象、安全、解释型、平台无关、分布式、多线程、动态性及开源性等特点。

1. 简单（Simple）

Java 语言由 C++语言发展而来，其风格与 C++类似，因而开发人员如果熟悉 C++语言和面向对象的概念，就可以很容易地掌握 Java。此外，Java 又去除了 C++的头文件、指针变量、结构、运算符重载、多重继承等特性，减少了编程的复杂性。

Java 语言增加了自动内存单元收集（周期性地释放未被使用的内存）功能，在程序设计时不需要考虑内存的分配与释放，不仅简化了程序设计工作，而且能够大幅度降低出错率。

Java 语言为开发者提供了丰富的类库（相当于 C 语言的函数库），提供了从构造图形用户界面等简单任务到建立网络连接和访问数据库等复杂任务的各种方法，通过直接调用或扩展类库中的类，引用类库中已经实现的功能，实现了代码的重复利用，使程序编写变得简单、容易。

2. 面向对象（Object-Oriented）

面向对象技术是当今世界软件开发中最为常用的技术之一。Java 语言是一种纯粹的面向对象语言，它集中于对象和接口的设计，提供简单的类机制和动态的接口模型。具有共同特征和行为的对象的集合被抽象为类，类中封装了状态数据以及响应的方法，实现了模块化和信息隐藏。类提供了建立对象的模板，并且通过继承机制，子类可以使用父类的数据和方法，实现代码的复用，不仅降低开发过程的复杂性，而且程序的代码量得以大大减少，使得开发大型、复杂的应用程序的成功机率大大提高。

3. 安全（Security）

Java 在设计之初就十分注重语言的健壮性，避免了一些不稳定的因素，防止了许多编程错误，在此基础上还采取了其他一些安全措施，主要体现在以下几个方面：

（1）Java 是强类型的语言，要求用显式的方法声明，允许在编译时进行深入的检查使得编译器可以发现方法调用错误，保证程序更加可靠。

（2）Java 不支持指针操作，杜绝了对内存的非法访问，同时其提供垃圾自动回收机制，防止了由于动态内存分配导致的内存丢失等问题。

（3）Java 提供了异常处理机制，可以在程序运行期间“捕捉”错误，例如可以在 Java 解释器运行时实施检查机制，发现数组和字符串访问的越界异常，这样可以简化错误处理，避免系统崩溃。

（4）当 Java 应用于网络应用程序开发时，可以通过自身的安全机制防止病毒程序对本地系统的破坏。

4. 解释型（Interpreted）

Java 是一种解释型的程序设计语言。Java 代码编译后不直接生成特定的机器码，而是生成 Java 字节码，在程序运行时借助 Java 解释器（即 Java 虚拟机，Java Virtual Machine，JVM）对编译后的字节码文件进行解释执行。Java 解释器是与平台有关的，它使得 Java 程序在某一特定的软、硬件平台上直接运行目标代码指令，所以程序员在编写程序时不必考虑运行环境。

与传统的解释型语言（例如 HTML、JavaScript）相比，Java 语言可以看作是解释型与编译型的折衷，Java 源程序先被编译，然后再被解释执行，编译后的字节码文件更接近机器指令，从而有效地克服了传统解释型语言的性能瓶颈（效率低、速度慢），同时又保证了解释型语言的可移植性。

5. 平台无关（Architecture-Neutral）

Java 是一种目标代码级的平台无关语言，用 Java 编写的应用程序不用修改就可以在不同的软硬件平台上运行。Java 主要通过 Java 虚拟机实现其目标码级的平台无关性。JVM 是一种用软件编写的抽象机器，它运行在具体操作系统之上，本身具有一套虚拟机器指令，并有自己的栈、寄存器组等。Java 源程序经过编译之后产生 JVM 可以解释的字节码文件，任何一台机器只要安装了 JVM，就可以运行 Java 应用程序，而无论这种字节码是在哪一种平台上生成的。

注意：可以登录http://www.oracle.com 下载针对各种平台的 Java运行环境。

6. 分布式（Distributed）

分布式分为数据分布和操作分布两种形式，Java 同时支持这两种分布形式。对于数据分布，Java 提供了一个 URL 对象，利用此 URL 对象，可以开启并访问具有相同 URL 地址的对象，访问方式与访问本地文件系统相同；对于操作分布，Java 提供了 Applet 小程序，可以从服务器下载到客户端运行，使得部分计算在客户端完成，从而提高系统的执行效率。

7. 多线程（Multi-Thread）

Java 提供多线程支持，主要体现在两个方面：其一，Java 环境本身就是多线程的，若干个系统线程并发运行负责必要的无用单元回收、系统维护等系统级操作；其二，Java 语言内置了多线程控制，可以大大降低多线程应用程序开发的难度。

8. 动态性（Dynamic）

Java 允许程序动态地装入运行中所需要的类，或者在原有的类中自由地加入新的方法和实例变量而不影响用户程序的运行。Java 语言不支持多重继承，但它通过接口间接实现多重继承，接口告诉连接对象它可以做什么而不是怎么做，使之比严格的类继承具有更灵活的方式和扩展性。Java 语言的类有运行时的表述，可以根据运行类型信息分辨类之间的关系并自动进行转换。

9. 开源性（Open Source）

2007 年 5 月，Sun 公司在其 OpenJDK Web 站点上正式发布 JDK 的源代码，这一举措兑现了 Sun 公司使 Java 成为开放源代码软件的承诺，这意味着以 Linux 为核心技术的其他厂商能够更方便地在它们的产品中集成 Java，使 Java 赢得编程人员的更多关注。

1.3 Java 体系结构

Java 不仅是开发各种应用程序的编程语言，还是支持 Java 应用程序进行编译、运行的平台。完整的 Java 体系结构由 4 部分组成，包括 Java 源程序、Java 类文件、Java API（Application Programming Interface，应用程序接口）以及 JVM。图 1-1 显示了 Java 不同部分之间的相互关系，以及它们与应用程序、与操作系统之间的关系。

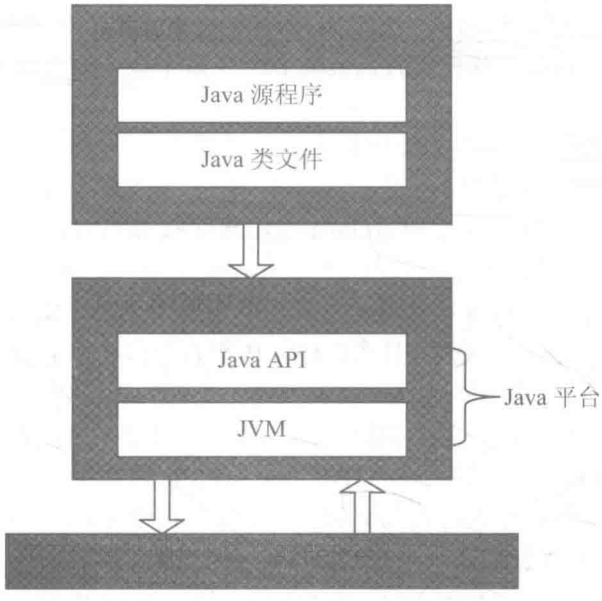

图 1-1 Java 体系结构

Java API 是已编译的可在任何 Java 程序中使用的代码库（即 Java 类库）。它们作为可定制的现成功能可以随时添加到 Java 应用程序中，以节约编程时间，实现代码复用。

JVM 为 Java 虚拟机，它类似于可运行 Java 程序的抽象计算机。任何一个平台只要安装 JVM，就可以保证经过编译的任何 Java 代码能够在该平台上运行。

JVM 与 Java API 共同构成了 Java 平台，也称为 JRE（Java Runtime Environment，Java 运行时环境），该平台可以建立在任何操作系统上。字节码文件需要借助 JRE 才能在各种不同操作系统和计算机平台中执行。Java 平台发展到 Java 2 版本后，为了适应不同级别应用程序开发的需要，又分为三个分支：

1. Java SE（Java 2 Platform Standard Edition）

Java SE 是 Java 平台标准版，包含核心 Java 类和 GUI 类，提供基础 Java 开发工具、执行环境与 API，主要适用于桌面应用系统的开发。

2. Java EE（Java 2 Platform Enterprise Edition）

Java EE 是 Java 平台企业版，它是由 Sun 公司提出的一组技术规格，规划企业用户以 Java 2 技术开发、分发、管理多层式应用结构。该企业版的开发包中包含开发 Web 应用程序所需的类和接口，如 Servlet、JSP 以及 JavaBean 等，主要适用于分布式的网络程序的开发，如电

子商务网站和 ERP 系统。

3. Java ME（Java 2 Platform Micro Edition）

Java ME 是 Java 平台微型版，是经过高度优化的 Java 运行环境，提供嵌入式系统所使用的 Java 开发工具、执行环境和 API，适用于消费类电子产品，如手机和 PDA 的嵌入式系统编程。

无论上述哪种 Java 运行平台，都包括了相应的 Java 虚拟机，虚拟机负责将字节码文件（包括程序使用的类库中的字节码）加载到内存，然后采用解释方式来执行字节码文件，即根据相应平台的机器指令翻译一句执行一句。

1.4　Java 开发环境

开发 Java 应用程序，首先要建立 Java 开发环境。Java 语言自身提供了开发环境 JDK（Java Development Kit，Java 开发工具集），目前的最新版本是 JDK7.0。之前的版本仍然可以使用，本书使用的 JDK 版本是相对稳定的 JDK6.0 版本。

1.4.1　JDK 简介

JDK 是一种用于构建在 Java 平台上编译和发布 Java 程序的开发和运行环境。JDK 包含了一系列编写、运行 Java 程序所需要的工具，其核心 Java API 是一些预定义的类库，开发人员需要通过这些类来访问 Java 语言预先定义的功能。在这些工具当中，常用的有以下几种：

（1）javac：编译器，输出结果为 Java 字节码文件（即.class 文件）。

（2）java：字节码解释器，直接解释执行字节码文件。

（3）javadoc：API 文档生成器，根据 Java 源代码及说明语句生成 HTML 格式的 API 文档。

（4）appletviewer：小应用程序浏览器，用于调试运行 Java 小应用程序。

（5）jar：Java Archiver 文件归档工具。

（6）jdb：Java 语言调试器。

（7）javah：头文件生成器，用于从 Java 类中调用 C++代码。

（8）javap：Java 反编译器，将 Java 的字节码文件转换为 Java 源文件。

1.4.2　JDK 的安装

JDK 下载结束后，执行 JDK 安装程序，系统将显示 JDK 安装向导画面。默认情况下，JDK 及其所包含的 JRE 将被安装到 C:\Program Files\Java 文件夹中。将 JDK 安装到默认目录下后，会形成如图 1-2 所示的目录结构。

图 1-2　JDK 目录结构

JDK 目录的主要内容如下：

（1）bin：开发工具，包括常用工具和实用程序，可帮助开发、执行、调试 Java 程序，例如，编译器 javac.exe 和解释器 java.exe 都位于该目录中。

（2）demo：演示程序，是 Java 平台的编程示例（带源代码）。这些示例包括使用 Java Swing 和其他 Java 基类以及 Java 平台调试器体系结构的示例。

（3）include：C 头文件，是支持使用 Java 本机界面、JVM 工具界面以及 Java 平台的其

他功能进行本机代码编程的头文件。

（4）jre：Java 运行环境，包括 Java 虚拟机、类库以及其他支持执行 Java 程序运行的文件。

（5）lib：附加库，是开发工具所需的其他类库和支持文件。

注意：JDK 安装目录下的文件 src.zip 中提供了 JDK 类库的全部源代码。

1.4.3 开发环境配置

要顺利使用 JDK 提供的各种工具，必须正确配置相应的环境变量，一个是 Path 环境变量，保证操作系统能够寻找到本地 Java 运行工具；另一个是 classpath 环境变量，提供程序使用的类库的位置。

为此，可右击桌面上“我的电脑”图标，从弹出的快捷菜单中选择“属性”，打开“系统属性”对话框，然后单击“高级”选项卡，进行环境变量的配置。

首先配置 Path 环境变量：单击“环境变量”按钮，打开“环境变量”对话框（如图 1-3 所示）。单击对话框下方“系统变量”设置区中的 Path 系统变量，然后单击“编辑”按钮，打开“编辑系统变量”对话框（如图 1-4 所示）。在“变量值”编辑框中单击并向左侧拖动，使光标定位在变量内容的最前面，然后输入“D:\Java\jdk1.7.0_40\bin”，为 Path 系统变量增加 JDK 程序所在路径。

图 1-3 “环境变量”对话框

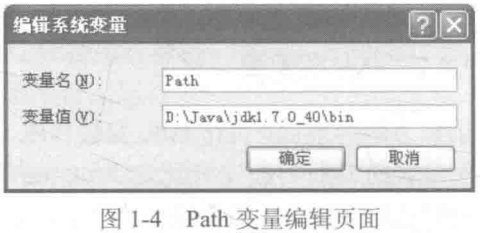

图 1-4 Path 变量编辑页面

接下来配置 classpath 环境变量：单击“系统变量”设置区中的“新建”按钮，打开“新建系统变量”对话框。新建系统变量名为“classpath”，设置其值为“D:\Java\jdk1.7.0_40\jre\lib\rt.jar;.;”（如图 1-5 所示），其中，位于两个分号中间的“.”表示在当前目录中查找 Java 类库。

为检验 Java 开发环境是否配置成功，可以采取如下的方法：单击“开始”按钮，选择“运行”，打开“运行”对话框。在“打开”编辑框中输入“cmd”，按 Enter 键，打开仿真 DOS 窗口。接下来在 DOS 提示符下输入“javac”并按 Enter 键，执行 javac 命令，如果能看到如图 1-6 所示的 javac 命令帮助信息，则表明系统变量设置已经生效了。

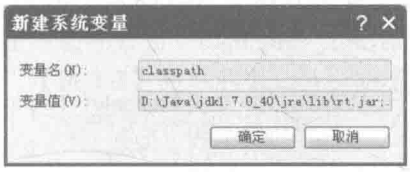

图 1-5　classpath 环境变量配置页面

图 1-6　javac 命令帮助信息

注意：修改了系统环境变量后，必须重新打开 DOS 窗口，新设置的环境变量才能生效。

1.5　Java 程序开发实例

根据运行环境不同，Java 程序可以分为两类，分别是 Java Application 和 Java Applet。这两类程序的开发原理相同，但应用于不同的场合。Java Application 在 Java 平台上独立运行，通常被称为 Java 应用程序，而 Java Applet 则必须嵌在 HTML 编写的 Web 网页中，通过浏览器运行，通常被称为 Java 小程序。

Java 程序的开发必须经过编写、编译、运行三个步骤，如图 1-7 所示。

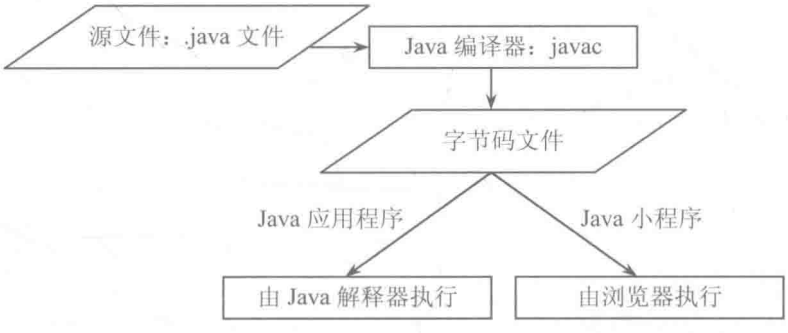

图 1-7　Java 程序开发步骤

1. 编写

编写是指在 Java 开发环境中进行程序代码的输入，形成后缀名为“.java”的 Java 源文件。

2. 编译

编译是指使用 Java 编译器对源文件进行语法错误排查的过程，编译后生成后缀名为.class 的字节码文件。字节码文件是一种与任何具体机器环境及操作系统无关的中间代码，它是一种二进制文件。

3. 运行

运行是指使用 Java 解释器将字节码文件翻译成机器代码，执行并显示结果。

1.5.1 Java Application

1. 编写源程序

Java 程序的源程序是以“.java”为后缀的文本文件，可以使用任何文本编辑器来编写，如 Windows 95/98/2000/XP/7 的记事本。例 1-1 利用记事本编写如下所示的 Java 程序代码，并将这段代码保存成名为“Example1_1.java”的文件。

【例 1-1】简单的 Java 应用程序实例

```
1    public  class  Example1_1 {
2        public static void main(String[] args){                    //定义一个类 Example1_1
3            System.out.println("This is my first Java program!");   //向屏幕输出字符串
4        }
5    }
```

2. 编译 Java 应用程序

打开一个命令提示符窗口，然后进入 Example1_1.java 文件所在的目录，输入如下命令开始编译源程序，Java 源程序必须带上扩展名（.java），否则编译程序会提示出错。

```
javac Example1_1.java
```

Java 源程序经过编译后得到的字节码文件为“.class”文件，一个源程序可以编译成一个或多个字节码文件，每个字节码文件对应源程序中定义的一个类，文件名与对应的类名相同。例 1-1 的源程序编译后生成的文件名为“Example1_1.class”，该文件被自动保存在源程序所在的目录下。

3. 运行 Java 应用程序

运行编译好的 Java 字节码文件需要调用 Java 的解释器 java.exe。在编译后使用如下命令运行已经生成的 Example1_1.class 文件。

```
java Example1_1
```

运行结果：

```
This is my first Java program!
```

注意：运行时在类名后面不要增加字节码文件的后缀.class，否则系统会提示找不到一个名为 Example1_1.class 的类。

说明：

（1）本程序仅实现一个简单的屏幕输出功能，其作用就是在显示器上输出一行信息：“This is my first Java program!”。

（2）所有的 Java 程序都是由类的定义组成的，一个 Java 源程序文件中可以定义多个类，但只能存在一个 public 类，并且该类的类名要与源文件名相同（包括字母的大小写）。

（3）Java 用关键字 class 来声明一个新的类，class 前面可以有若干标志该类属性的限制性关键字，如本例中的 public 表示该类是公有的，可以在任何场合使用。class 关键字后面是该类的类名，接下来是类的定义，整个类定义由大括号{}括起来。

（4）在本类中定义了一个 main()方法，对于一个 Java 应用程序来说，main()方法是必须的，而且必须按照以上的格式定义方法头。Java 解释器在没有生成任何实例的情况下，以 main()方法作为程序的入口。

（5）Java 的语句必须用“;”结束，“//”符号后面的内容为注释，它是对程序的解释，注释的内容在编译时是被忽略的，适当的注释可以提高程序的可读性。