陈国君　主编

Java
程序设计基础
(第6版)

清华大学出版社

陈国君　主编
陈　磊　副主编
李梅生　刘　洋　鲜征征　刘秋莲　编著

Java 程序设计基础

(第6版)

清华大学出版社
北京

内容简介

本书全面、系统地介绍Java语言的特点及应用技术，内容以Java的基础程序设计、面向对象程序设计和事件处理为三大主线，利用浅显易懂的语言、丰富简单的实例，完整地介绍了Java面向对象程序设计的重点和难点。本次改版增加了注解、反射、Lambda表达式等内容，去掉了小程序设计的内容。例题采用目前最新的Java 10技术进行重新编写，尤其是图形界面程序设计中例题采用目前最流行的JavaFX 2.2架构重新编写，充分体现了新技术的特点。本书共分18章，其中第1～5章介绍程序设计基础；第6～11章介绍面向对象程序设计；第12章介绍泛型和容器类；第13章介绍注解、反射、内部类、匿名内部类与Lambda表达式；第14、15章介绍图形界面设计和事件处理；第16章介绍绘图与动画程序设计；第17章介绍Java数据库程序设计；第18章介绍Java网络编程。

本书的特色：概念清楚、结构合理；深入浅出、条理分明；内容连贯，循序渐进；重点突出，分解难点；选材精细，通俗易懂。尤其在结构上特别注重前后内容的连贯性，力求抓住关键、突出重点、分解难点，体现“理论性、实用性、技术性”三者相结合的编写特色。对每个知识点不但能告诉读者怎么做，而且还能告诉读者这么做的原因和道理。

本书可以作为高等院校计算机及其相关专业的教学用书，也可作为各学校程序设计公共选修课的教材，还可用作职业教育的培训用书和Java初学者的入门教材或供具有一定Java编程经验的开发人员学习使用。

图书在版编目(CIP)数据

Java程序设计基础/陈国君主编. —6版. —北京：清华大学出版社，2019(2019.1重印)
ISBN 978-7-302-51551-7

Ⅰ. ①J…　Ⅱ. ①陈…　Ⅲ. ①JAVA语言－程序设计　Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第250575号

责任编辑：刘向威　张爱华
封面设计：文　静
责任校对：胡伟民
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：28.5　　**字　　数**：695千字
版　　次：2006年1月第1版　2019年1月第6版　　**印　　次**：2019年1月第2次印刷
印　　数：1501～5500
定　　价：69.00元

产品编号：081632-01

前言

本书自2006年面市以来，深受广大读者的好评，市场反应非常热烈，一直畅销，经久不衰。尤其是本书的第3版被“中国书刊发行业协会”评为全行业优秀畅销教材后，市场需求量更是迅速倍增。为了能适应Java技术的快速发展和计算机教学的需要，清华大华出版社和本书作者在征求广大读者的意见和建议的基础上，决定修订再版，以便更好地满足广大读者的需求。

本次改版，增加了注解、反射、Lambda表达式等内容，去掉了小程序设计的内容。书中例题采用目前最新的Java 10技术重新编写，尤其是图形界面程序设计中例题采用目前最流行的JavaFX 2.2架构重新编写，充分体现了新技术的特点。每个例题都突出一个编程的知识点，并保持原书的由浅入深、循序渐进、突出重点、分解难点的编写特色，使本版书在体系结构、内容组织、语言表达等方面都更加完善，同时使学生感到学习Java编程是一种兴趣，而兴趣又成为学习Java语言的动力，让学生在学习的乐趣中掌握Java的基本编程技巧。这种良性循环都归功于本书对内容的精选和组织结构的合理性，衷心地希望本书能成为广大读者的良师益友。

正是由于本书优化的知识体系，通俗易懂的讲解方式，对知识点的透彻分析和灵活实用的举例，因而深受读者的欢迎，这也是催生本书再版的主要原因。由于Java技术的内容庞大、结构复杂，所以从其中抽出基本的内容，并能以通俗的方式介绍给读者并非易事，所以本书难免存在不尽如人意的地方，因此希望广大读者继续能对本书提出合理化建议，使本书更加完善。由于计算机技术发展很快，加之作者水平有限，书中难免有不足之处，欢迎广大读者斧正。

书中所有例题全部在JDK 10环境下编译通过并运行。

本书由陈国君、陈磊、李梅生、刘洋、鲜征征、刘秋莲共同修改完成。

本教材的再版，得到了清华大学出版社的大力支持，在此本书全体作者对清华大学出版社的大力支持，尤其是编辑刘向威博士的热心关注、建议与指导，表示衷心的感谢！

作　者

2018年10月

目录

第 1 章　Java 语言概述

本章主要内容：

- Java 语言的特点；
- Java 源文件(.java)与 Java 字节码文件(.class)；
- Java 应用程序和 Java 小程序的主类；
- Java 虚拟机；
- Java 程序的种类和结构。

Java 语言是一种简单易用、完全面向对象、与平台无关、安全可靠、主要面向 Internet 的开发工具。

1.1　Java 语言的诞生与发展

Java 语言诞生于 20 世纪 90 年代初期，从它正式问世以来，它的快速发展已经让整个 Web 世界发生了翻天覆地的变化。Java 语言的前身是 Sun Microsystems 公司(Sun 公司于 2009 年 4 月被 Oracle 公司收购)开发的一种用于智能化家电的名为 Oak(橡树)的语言，它的基础是当时最为流行的 C 和 C++语言。但是，由于一些非技术上的原因，Oak 语言并没有得到迅速的推广。直到 1993 年，WWW(万维网)迅速发展，Sun 公司发现可以利用 Oak 语言的技术来创造含有动态内容的 WWW 网页，于是已受人冷落的 Oak 语言又被重新开发和改造，并将改造后的 Oak 语言改名为 Java 语言。Java 是太平洋上的一个盛产咖啡的岛屿的名字。终于，在 1995 年，Java 这个被定位于网络应用的程序设计语言被正式推出。

由于 Java 语言功能强大，其问世后不久，即被业界广泛接受，于是 IBM、Apple、DEC、Adobe、HP、Oracle、Toshiba、Netscape 和 Microsoft 等大公司均购买了 Java 语言的许可证。Microsoft 公司还从其 Web 浏览器 Explorer 3.0 版起开始增加了对 Java 语言的支持。同时，众多的软件开发商也开发了许多支持 Java 的产品。在目前以网络为中心的计算机时代，不支持 HTML 和 Java 语言，就意味着应用程序的应用范围只能限于同质的环境。

随着 Java Servlet 的推出，Java 语言极大地推动了电子商务的发展。Java Server Page (JSP)技术的推出，更是让 Java 语言成为基于 Web 应用程序的首选开发工具。Internet 的普及和迅猛发展，以及 Web 技术的不断渗透，使得 Java 语言在现代社会的经济发展和科学研究中占据越来越重要的地位。

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编程语言。它具有简单、面向对象、分布式、解释型、可靠性、安全、平台无关、可移植、高性能、多线程、动态性等特点。下面介绍 Java 语言的几个重要特性。

1. 简单易学

Java 语言虽然衍生自 C++语言，与 C++语言相比 Java 语言是一种完全面向对象的编程语言。出于安全性和稳定性的考虑，Java 语言去掉了 C/C++语言支持的三个不易理解和掌握的数据类型：指针(pointer)、联合体(unions)和结构体(structs)。而 C/C++语言中联合体和结构体的功能，完全可以在 Java 语言中用类及类的属性等面向对象的方法来实现，这不但更加合理规范，而且还降低了学习难度。

2. 面向对象

Java 语言最吸引人之处，就在于它是一种以对象为中心、以消息为驱动的面向对象的编程语言。面向对象的语言都支持封装、继承和多态三个概念，Java 语言也是如此。

3. 平台无关性

Java 语言是与平台无关的语言，这是指使用 Java 语言编写的应用程序不用修改就可在不同的软硬件平台上运行。平台无关有两种：源代码级和目标代码级。C 和 C++语言具有一定程度的源代码级平台无关，即用 C 和 C++语言编写的应用程序不用修改只需重新编译就可以在不同平台上运行。Java 语言是靠 Java 虚拟机(JVM)在目标代码级实现平台无关性的。

4. 分布式

分布式包括数据分布和操作分布。Java 语言支持这两种分布性。Java 语言提供了一整套网络类库，开发人员可以利用类库进行网络程序设计，方便地实现 Java 语言的分布式特性。

5. 可靠性

Java 语言具有很高的可靠性。Java 解释器运行时实施检查，可以发现数组和字符串访问的越界；另外，Java 语言提供了异常处理机制，可以把一组错误的代码放在一个地方，这样可以简化错误处理任务，便于恢复。

6. 安全性

Java 语言具有较高的安全性。当 Java 字节码进入解释器时，首先必须经过字节码校验器的检查；其次，Java 解释器将决定程序中类的内存布局；再次，类装载器负责把来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏；最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。Java 语言综合了上述几种机制，成为安全的编程语言。

7. 支持多线程

Java 语言在两方面支持多线程：一方面，Java 环境本身就是多线程的，若干系统线程运行，负责必要的无用单元回收、系统维护等系统级操作；另一方面，Java 语言内置多线程机制，可以大大简化多线程应用程序开发。

8. 支持网络编程

Java 语言通过它所提供的类库可以处理 TCP/IP，用户可以通过 URL 地址在网络上很方便地访问其他对象。

9. 编译与解释并存

Java 语言的编译器并不是把源文件(.java)编译成二进制码，而是将其编译成一种独立于机器平台的字节码文件(.class 文件)。字节码文件可以被 Java 解释器执行，由解释器将字节码文件再翻译成二进制码，使程序得以运行。

1.3 Java 语言规范

Java 语言有严格的使用规范。Java 语言规范是对语言的技术定义，包括 Java 程序设计语言的语法和语义。如果编写程序时没有遵守这些规范，计算机就不能理解程序。Java 语言还为开发 Java 程序而预定义了类和接口，称为应用程序接口(Application Program Interface，API)。

目前，Java 技术主要包括如下三个方面。

Java SE(Java Platform Standard Edition)：Java 平台的标准版，可以用于开发客户端应用程序。应用程序可以独立运行或作为 Applet 在 Web 浏览器中运行。

Java ME(Java Platform Micro Edition)：Java 平台的精简版，用于开发移动设备的应用程序。不论是无线通信还是手机、PDA 等小型电子装置，均可采用 Java ME 作为开发工具及应用平台。

Java EE(Java Platform Enterprise Edition)：Java 平台的企业版，用于开发服务器端的应用程序，为企业提供了 e-Business 架构及 Web 服务。其优越的跨平台能力与开放的标准，深受广大企业用户的喜爱。

由于 Java SE 是基础，其他 Java 技术都基于 Java SE，所以本书采用目前最新版本 Java SE 10 介绍 Java 程序设计。与 Java SE 10 对应的 Java 开发工具包称为 JDK 10。

1.4 Java 虚拟机

大部分的计算机语言程序都必须先经过编译(compile)或解释(interpret)的操作后，才能在计算机上运行，然而，Java 程序(.java 文件)却比较特殊，它必须先经过编译的过程，然后再利用解释的方式来运行。通过编译器(compiler)，Java 程序会被转换成与平台无关(platform-independent)的机器码，Java 称之为“字节码”(byte-codes)，字节码文件的扩展名为.class。通过 Java 的解释器(interpreter)便可解释并运行 Java 的字节码。图 1.1 说明了 Java 程序的执行过程。

图 1.1 Java 程序的运行过程：先编译，后解释

字节码是 Java 虚拟机(Java Virtual Machine,JVM)的指令组,和 CPU 上的微指令码很相像。Java 程序编译成字节码后文件尺寸较小,便于网络传输。

字节码最大的好处是可跨平台运行,即 Java 的字节码可以编写一次,到处运行。用户使用任何一种 Java 编译器将 Java 源程序(.java)编译成字节码文件(.class)后,无论使用哪种操作系统,都可以在含有 JVM 的平台上运行。这种跨越平台的特性也是让 Java 语言急速普及的原因之一。

任何一种可以运行 Java 字节码的软件均可被看成 Java 的"虚拟机"(JVM),如浏览器与 Java 的开发工具等皆可被视为一部 JVM。很自然地,可以把 Java 的字节码看成 JVM 上所运行的机器码(machine code),即 JVM 中的解释器负责将字节码解释成本地的机器码。所以从底层上看,JVM 就是以 Java 字节码为指令组的"软 CPU"。可以说,JVM 是可运行 Java 字节码的假想计算机。它的作用类似于 Windows 操作系统,只不过在 Windows 上运行的是.exe 文件,而在 JVM 上运行的是 Java 字节码文件,也就是扩展名为.class 的文件。JVM 其实就是一个字节码解释器。

1.5 Java 程序的种类和结构

使用 Java 语言可以编写两种类型的程序:Application(应用程序)和 Applet(小程序)。这两种程序的开发原理是相同的,但是在运行环境和计算结构上却有着显著的不同。

应用程序是从命令行运行的程序,它可以在 Java 平台上独立运行,通常称为 Java 应用程序。Java 应用程序是独立完整的程序,在命令行调用独立的解释器软件即可运行。另外,Java 应用程序的主类包含有一个定义为 public static void main(String[] args)的主方法,这个方法是 Java 应用程序的标志,同时也是 Java 应用程序执行的入口点,在应用程序中包含有 main()方法的类一定是主类,但主类并不一定要求是 public 类。

小程序是嵌入在 HTML(超文本标记语言)文档中的 Java 程序,需要搭配浏览器来运行,因此称为小程序。由此可见,当运行一个 Java 小程序时,同时还要为它编写一个 HTML 文件,然后在 WWW 浏览器中运行这个 HTML 文件,就可以激活浏览器中的 Java 解释器。另外,也可以调用一些能够模拟浏览器环境并执行 Java 小程序的软件来直接运行 Java 小程序。由于浏览器受安全控制的限制,所以 Java 小程序一般使用模拟浏览器环境的软件来执行。

一个复杂的程序可以由一个或多个 Java 源文件构成,每个文件中可以有多个类定义。下面的程序是一个 Java 应用程序文件。

说明:为了便于对程序代码的解释,本书在每行代码之前加一标号,它们并不是程序代码的一部分。

```
1   package ch01;                                  //定义该程序属于 ch01 包
2   import java.io.*;                              //导入 java.io 类库中的所有类
3   public class App1_1                            //定义类: App1_1
4   {
5     public static void main(String[] args)       //定义主方法
6     {
7       char c = ' ';
8       System.out.print("请输入一个字符: ");
```

```
9        try{
10        c = (char)System.in.read();
11       }catch(IOException s){   }
12       System.out.println("您输入的字符是: " + c);
13     }
14   }
```

从这个程序可以看出，一般的 Java 源程序文件由以下三部分组成：

- package 语句(0 个或 1 个)；
- import 语句(0 个或多个)；
- 类定义(1 个或多个类定义)。

其中，package 语句表示该程序所属的包。它只能有一个或者没有。如果有，必须放在最前面；如果没有，表示本程序属于默认包。

import 语句表示引入其他类库中的类，以便使用。import 语句可以有 0 或多个，它必须放在类定义的前面。

类定义是 Java 源程序的主要部分，每个文件中可以定义若干类。

Java 程序中定义类使用关键字 class，每个类的定义由类头定义和类体定义两部分组成。类体定义部分用来定义属性和方法这两种类的成员，其中方法类似于其他高级语言中的函数，而属性则类似于变量。类头部分除了声明类名之外，还可以说明类的继承特性，当一个类被定义为是另一个已经存在的类(称为父类)的子类时，它就可以从其父类中继承一些已定义好的类成员而不必自己重复编码。

在类体中通常有两种组成成分：一种是域，包括变量、常量、对象、数组等独立的实体；另一种是方法，类似于函数的代码单元块。这两种组成成分通称为类的成员。在上面的例子中，类 App1_1 中只有一个类成员，即第 5 行定义的方法 main()。用来标志方法头的是方法名后面的一对小括号，小括号里面是该方法使用的形式参数，方法名前面的 public 用来说明这个方法属性的修饰符，其具体语法规定将在第 6 章中介绍。方法体部分由若干以分号";"结尾的语句组成，并由一对大括号{}括起来，在方法体内部不能再定义其他的方法。

同其他高级语言一样，语句是构成 Java 程序的基本单位之一。每一条 Java 语句都以分号";"结束，其构成应该符合 Java 语言的语法规则。类和方法中的所有语句应该用一对大括号{}括起来。除 package 及 import 语句之外，其他执行具体操作的语句，都只能存在于类的大括号之中。

比语句更小的语言单位是表达式、变量、常量和关键字等，Java 的语句就是由它们构成的。其中，声明变量与常量的关键字是 Java 语言语法规定的保留字，用户程序定义的常量和变量的取名不能与保留字相同。

Java 源程序的书写格式比较自由，如语句之间可以换行，也可以不换行，但养成一种良好的书写习惯比较重要。

注意：Java 是严格区分字母大小写的语言。书写时，大小写不能混淆。

一个程序中可以有多个类，但只能有一个类是主类。在 Java 应用程序中，这个主类是指包含 main()方法的类。在 Java 小程序里，这个主类是一个继承自系统类 JApplet 的子类。应用程序的主类不一定要求是 public 类，但小程序的主类一定要求是 public 类。主类是 Java 程序执行的入口点。

本章小结

1. Java 程序比较特殊，它必须先经过编译的过程，然后再利用解释的方式来执行。即首先要将源程序(.java 文件)通过编译器将其转换成与平台无关的字节码文件(.class 文件)，然后再通过解释器来解释执行字节码文件。

2. 字节码(byte-codes)最大的好处是可跨平台执行，可让程序“编写一次，到处运行(write once，run anywhere)”的梦想成真。

3. Java 程序可分为两种：一种是 Application，称为 Java 应用程序；另一种是 Applet，称为 Java 小程序。Java 应用程序是指可以在 Java 平台上独立运行的一种程序；而 Java 小程序则是内嵌在 HTML 文件里，需要在浏览器的支持下才能运行。

4. 无论是应用程序还是小程序都必须有一个主类，主类是程序执行的入口点，应用程序的主类是包含有 main()方法的类，但应用程序的主类并不一定要求是 public 类；小程序的主类是一个继承自系统类 JApplet 的子类，且该类必须是 public 类。

第 1 章习题

1.1 Java 语言有哪些特点？

1.2 什么是 Java 虚拟机？

1.3 什么是字节码？采用字节码的最大好处是什么？

1.4 什么是平台无关性？Java 语言是怎样实现平台无关性的？

1.5 Java 语言程序有几种？每种程序的结构包含哪几个方面？

1.6 什么是 Java 程序的主类？应用程序与小程序的主类有何不同？

第 2 章　Java 语言开发环境

本章主要内容：

- Java 开发工具的下载与安装；
- JDK 开发环境的配置；
- Java 源文件的命名规则；
- 在 JDK 环境中编译与运行 Java 应用程序。

Java 开发工具早年是 Sun 公司所开发的一套 Java 程序开发软件，由于 Sun 公司于 2009 年 4 月被 Oracle 公司收购，所以现在它可在 Oracle 公司的网站免费取得。它与 JDK 的帮助文档(Java docs)一起是编写 Java 程序必备的工具。

2.1　Java 开发工具

Java 开发工具(Java SE Development Kits，JDK)是许多 Java 程序员使用的开发环境。尽管许多编程人员已经使用第三方的开发工具，但 JDK 仍被当作 Java 程序开发的重要工具。

JDK 由 Java API、Java 运行环境和一组建立、测试工具的 Java 实用程序等组成。其核心是 Java API，所谓 API(Application Programming Interface)就是 Java 提供的标准类库供编程人员使用，开发人员需要用这些类来实现 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 等。

作为 JDK 的实用程序，工具库中的主要程序都放在 JDK 安装文件夹下，其中 bin 子文件夹中包含了所有相关的可执行文件，下面是 bin 文件夹下的常用命令。

- javac.exe：Java 编译器，将 Java 源代码文件转换成字节码文件；
- java.exe：Java 解释器，执行 Java 程序的字节码文件；
- appletviewer.exe：小程序浏览器，执行嵌入在 HTML 文件中的 Java 小程序的 Java 浏览器；
- javadoc.exe：根据 Java 源代码及注释语句生成 Java 程序的 HTML 格式的帮助文档；
- jdb.exe：Java 调试器，可以逐行执行程序、设置断点和检查变量；
- jar.exe：创建扩展名为.jar(Java Archive，Java 归档)的压缩文件，与 zip 压缩文件格式相同；
- jmod.exe：创建扩展名为.jmod 的压缩文件。