“十三五”普通高等教育规划教材

Java 程序设计教程

崔 淼　赵晓华　主编

提供电子教案和素材文件

http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

“十三五”普通高等教育规划教材

Java 程序设计教程

崔 淼 赵晓华 主编

机械工业出版社

本书以面向对象程序设计的思想为主线，全面细致地介绍了 Java 程序设计的基础知识、特点及相关应用，注重引导读者从 C 语言的以函数为主的面向过程的程序设计，过渡到以类和对象为主的面向对象的程序设计。本书共分为 12 章，主要包括 Java 语言概述，类和对象，深入理解类及其成员，继承、抽象类、接口和多态，数组与集合，异常和异常处理，输入/输出与文件管理，数据库编程、多线程，Java 网络编程，JavaFX 基础和 JavaFX Scene Builder 等方面的内容。

本书适合作为高等院校计算机专业教材使用，同时也可作为广大计算机爱好者的学习用书和各类 Java 程序设计培训班的教学用书。

本书配套授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：2850823885，电话：010-88379739）。

图书在版编目（CIP）数据

Java 程序设计教程 / 崔淼，赵晓华主编. —北京：机械工业出版社，2019.5

“十三五”普通高等教育规划教材

ISBN 978-7-111-62467-7

Ⅰ. ①J… Ⅱ. ①崔… ②赵… Ⅲ. ①JAVA 语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 068185 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：胡 静 责任编辑：胡 静

责任校对：张艳霞 责任印制：郜 敏

北京圣夫亚美印刷有限公司印刷

2019 年 5 月第 1 版 • 第 1 次印刷

184mm×260mm • 20.25 印张 • 501 千字

0001－2500 册

标准书号：ISBN 978-7-111-62467-7

定价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：（010）88379833

读者购书热线：（010）68326294

网络服务

机 工 官 网：www.cmpbook.com

机 工 官 博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

金 书 网：www.golden-book.com

封面无防伪标均为盗版

前　言

Java 是一种经典的程序设计语言，它全面支持面向对象的程序设计方法。因此，Java 在国内外各个领域中得到了广泛的应用，有着极高的市场占有率。本教材以 Eclipse 4.7 + JRE 10.0 为开发平台，结合大量易于理解的实例，面向初步学习了 C 语言的读者，从面向对象程序设计的角度，循序渐进地展开了 Java 程序设计基础知识和编程技术的介绍。在内容讲述上以深入浅出的布局，结合直观的图示、演练、实训以及在源代码中尽可能多地添加注释等手段，使读者能够较轻松地理解面向对象编程的基本概念和思想。

本教材重点突出面向对象的程序设计思想，不仅在讲述内容上详细介绍了面向对象的相关概念及编程技术，而且在几乎所有演练、实训中采用“任务驱动”的方式，强调使用面向对象的程序设计方法来实现程序功能。注重引导读者从 C 语言的以函数为主的面向过程程序设计，过渡到以类和对象为主的面向对象的程序设计方法。

本教材共分为 12 章，主要包括 Java 语言概述，类和对象，深入理解类及其成员，继承、抽象类、接口和多态，数组与集合，异常和异常处理，输入/输出与文件管理，数据库编程、多线程，Java 网络编程，JavaFX 基础和 JavaFX Scene Builder 等方面的内容。

作者讲授程序设计语言课程多年，并参加过许多实际应用系统的开发，有丰富的教学经验和实践经验。在教材内容的处理上，紧紧抓住面向对象的程序设计思想这条主线，使学生通过本教材的学习，不但能学会 Java 程序设计的基本知识、设计思想和方法，还能很容易地过渡到其他面向对象程序设计语言的学习与使用上。

本书适合作为高等院校计算机专业教材使用，同时也可作为广大计算机爱好者和各类 Java 程序设计培训班的教学用书。本书配套有完整的教学用 PPT 课件，并提供所有演练、实训的源代码，需要的读者可从机械工业出版社教学服务网（http://www.cmpedu.com）中下载。

本书由崔淼、赵晓华主编，其中，李鸿雁编写第 1、10 章，崔淼编写第 2、3、4、5 章，刘瑞新、刘克纯、骆秋容、翟丽娟、徐维维编写第 6 章，彭姣编写第 7 章，许萌编写第 8 章，赵晓华编写第 9 章，苏继斌编写第 11、12 章，程序的上机调试、代码优化及教学课件由赵晓华制作完成。本书由刘瑞新教授统稿。编写过程中得到了许多一线教师的大力支持，提出了许多宝贵意见，使本书更加符合教学规律，在此感谢。

由于计算机信息技术发展迅速，书中难免有不足和谬误之处，恳请广大读者批评指正。

编　者

目　录

第1章 Java语言概述

Java是一种优秀的编程语言，具有面向对象、与平台无关、安全稳定和多线程等特点，在全球编程语言排行榜中，从2002到2018年，Java一直位于第一，是最受欢迎的语言。

1.1 Java语言的特点及相关概念

Java是一种功能强大和多用途的编程语言，可用于开发运行在移动设备、台式计算机以及服务器端的应用程序。

1.1.1 Java语言的特点

Java是由美国Sun公司于1995年推出的。Java最初被称为Oak（橡树），是1991年为消费类电子产品的嵌入式芯片而设计的。1995年更名为Java，并重新设计成用于Web应用程序的开发。2010年Sun公司被美国Oracle（甲骨文）公司收购，目前Java商标归Oracle公司所有。Java是目前使用最为广泛的编程语言之一，它具有简单、面向对象、与平台无关、解释型、多线程、安全、动态等特点。

1．简单

Java语言的语法与C/C++和C#语言十分接近，这使得程序员可以很容易地学习和使用Java语言。此外，Java丢弃了C++中很少使用的、很难理解的、令人迷惑的一些特性，如操作符重载、多继承、自动强制类型转换等。特别是，Java语言不再使用指针的概念，并提供了自动的内存垃圾回收机制，使得程序员不必再编写任何关于内存管理的代码。

2．面向对象

Java语言是一个完全面向对象的程序设计语言，它提供了类、接口和继承等面向对象编程技术。Java支持类之间的单继承，支持接口之间的多继承，支持类与接口之间的实现机制和全面支持动态绑定。

3．与平台无关

与程序运行平台无关是Java较其他一些传统的编程语言最大的优势。Java源程序（*.java）经过编译后生成能被Java虚拟机（JVM）识别和执行的字节码文件（*.class），这种机制使得任何支持JVM的平台都可以很好地运行Java程序。当平台的操作系统（Windows、Linux等）或处理器发生变化或升级时无须对程序进行任何修改，从而实现了Sun公司提出的“一次写成，处处运行”的设计目标。

4．解释型

使用C和C++等语言编写的应用程序，在运行前需要针对当前计算机的操作系统和CPU进行编译，编译后生成对应本计算机的二进制可执行代码文件。显然，这样的应用程序

的可移植性较差。而 Java 应用程序经过编译后生成的是针对 JVM 的字节码文件，该文件在 JVM 中以解释方式被执行，从而提高了 Java 程序的适应能力和可移植性。

5．多线程

使用 Java 可以设计出能同时处理多项任务的多线程应用程序。多线程机制使应用程序能够并行执行，而且 Java 的同步机制保证了对共享数据的正确操作。通过使用多线程程序，设计者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，这样就很容易地实现网络上的实时交互行为。

6．安全

为适应在网络环境中的应用，Java 提供了一套完善的安全机制，可以有效防止恶意代码的攻击。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，提供了一个安全防范机制（ClassLoader 类）。例如，分配不同的名字空间以防替代本地的同名类。字节代码检查和安全管理机制（SecurityManager 类）也为 Java 应用程序提供了一个“安全哨兵”。

7．动态

Java 语言的设计目标之一就是适应动态变化的环境。Java 程序需要的类能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，这对软件的升级十分有利。

Java 语言的优良特性使得 Java 应用具有强大的健壮性和可靠性，减少了应用系统的维护费用。Java 对面向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，降低开发成本。Java 的编译一次到处可运行的特性，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相关技术和丰富的类库。

1.1.2 与 Java 相关的几个概念

在使用 Java 语言进行程序设计之前，首先需要理解以下几个与 Java 相关的基本概念。

1．Java 语言规范

Java 语言规范是对 Java 语言的技术定义，规定了 Java 语言的语法和语义，如关键字、标识符、语法格式等。完整的 Java 语言规范可以在 Oracle 的官方网站（http://docs.oracle.com/javase/specs/）中找到。

2．Java 虚拟机

任何一种可以解释并运行 Java 字节码的软件均可看成是 Java 虚拟机（JVM），如各类浏览器。可以将其理解为能解释并执行 Java 字节码的“软 CPU”，也就是说 JVM 是可以解释并运行 Java 字节码的假想软计算机。

3．API

API（Application Program Interface，应用程序接口）也称为“库”，其中包含有为开发 Java 程序而预定义的通用类和接口。使用这些具有工具性质的预定义类和接口，可以大幅度减轻开发人员的代码编写量，缩短软件的开发周期。

4．JRE

JRE（Java Runtime Environment，Java 运行环境），它包含了 Java 虚拟机、Java 基础类库等，是运行 Java 应用程序所必需的软件环境。计算机中只有正确安装和配置了 JRE 后，才能运行 Java 应用程序。

5. JDK

JDK（Java Development Kit，Java 开发工具包）由一系列独立的 Java 程序构成，是开发人员编写 Java 程序所需的开发工具包，它是提供给开发人员使用的预定义类、接口的集合。

JDK 包含了 JRE、Java 源代码的编译器 javac.exe 和运行 Java 程序的 java.exe，还包含了许多 Java 程序调试和分析的工具，如 jconsole.exe（Java 性能分析器）、jvisualvm.exe（Java 监控工具）等。此外，编写 Java 程序所需的文档和一些实例程序也包含在 JDK 中。

6. Java 版本

Java 是一种全面且功能强大的计算机程序设计技术，面对不同的用途 Java 提供了 Java SE、Java EE 和 Java ME 3 个不同的版本。

（1）Java SE

Java SE（Java Standard Edition，Java 标准版），可用来开发客户端应用程序。使用 Java SE 开发的应用程序可以独立运行，也可作为 Java Applet（Java 小程序）嵌入到 HTML 网页中在 Web 浏览器中运行。Java 基础学习一般都是在 Java SE 环境中进行的。

（2）Java EE

Java EE（Java Enterprise Edition，Java 企业版），可用来开发服务器端的应用程序，如 Java Servlet 和 JSP（Java Server Pages）。Java EE 提供了企业电子商务架构及 Web Services 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。目前，Java EE 已成为开发电子商务应用的首选平台。

（3）Java ME

Java ME（Java Micro Edition，Java 微型版）是一个精简的 Java 开发平台，可用于面向消费类产品和嵌入式设备的应用程序开发。无论是无线通信还是手机、PDA 等小型电子设备，均可采用 Java ME 作为开发工具及应用平台。它提供了对 HTTP 等 Internet 协议的支持，可以使手机等便携式设备以 C/S（客户端/服务器）的方式直接访问 Web 网站或本地存储的资源。

1.2 Java 与面向对象的程序设计

Java 是一个完全面向对象的程序设计语言。所谓“面向对象”是指将程序中遇到的所有实体都看作一个“对象”（Object），并将具有相同基本特征的对象归属到一个“类”（Class）中，可以将对象理解成类的一个具体实例。类是抽象的、模糊的；而类的对象确是具体的、明确的。例如，隶属于电视机类的某品牌，某型号具体的电视机对象。在使用 Java 以面向对象的方式进行针对电视机的程序设计时，通常需要先创建一个电视机类，定义出描述电视机所需的特征变量（如尺寸、分辨率、能耗等）和行为方法（如开机、关机、搜台、调整音量和色彩等）。而后，创建电视机类的实例，并通过该实例操作电视机类的特征变量或调用电视机类具有的方法，进而实现程序的预期功能。

1.2.1 Java 应用程序的构成

在使用 Java 进行程序设计时，开发人员的主要工作是进行类及其方法的设计，并通过代码控制类对象的特征变量（也称为字段变量或属性）、调用对象的方法，最终实现程序的

设计目标。

一个可以独立运行的 Java 应用程序由一个或多个相互关联的类组成，每个类中通常包含以下 3 个最基本的成员。

1）特征变量：也称为字段或属性。一个类中可以包含一个或多个特征变量，如描述一个圆对象的半径值、边线颜色、填充颜色、圆心位置等；描述一个学生对象的学号、姓名、性别、班级、成绩等。

2）方法：它是用于在程序中实现某些具体操作的代码段。一个类中可以包含一个或多个用于实现不同功能的方法，如一个圆对象可以有计算并输出圆面积值的方法、计算并输出周长的方法等；一个学生对象可以有修改学生成绩、计算总分、输出成绩单等方法。

3）构造方法：构造方法是类中一种特殊的方法，用于初始化类的对象，它没有返回值并且其名称必须与类名相同。

下列所示的是一个能根据用户输入的半径值，计算并输出圆面积值的 Java 应用程序的构成框架。

```
class 主类{
    主方法 main{
        用于接收用户输入的半径值的语句;
        创建圆类对象并为其半径赋以用户输入值的语句;
        通过圆对象调用圆类计算并输出面积方法的语句;
    }
}
class 圆类{
    声明圆半径变量的语句;
    计算并输出圆面积的方法{
        实现功能的代码;
    }
}
```

说明：

1）上述 Java 应用程序由两个类（主类和圆类）构成，每个类中又包含有各自的方法。

2）若希望一个 Java 应用程序能独立运行，则其中必须包含有一个命名为“main”的主方法，主方法是程序执行的起始点。它所在的类称为“主类”。

3）通常主方法仅用来提供用户的操作接口，它通过对象的特征变量存储数据，通过调用对象的方法实现程序功能，相当于应用程序的指挥中心。

4）Java 除了允许开发人员根据实际需要创建自定义类外，还在 JDK 中包含了众多可在程序中直接调用的、用于实现各种功能的预定义类。使用这些预定义类可以大幅度提高程序的开发效率，减轻开发人员的工作量。

1.2.2 创建、编译和执行 Java 应用程序

在 Windows 命令提示符窗口中编译和运行 Java 应用程序之前，首先需要在计算机中正确地安装和配置 JDK。

1. 下载、安装和配置 JDK

最新版的 JDK 安装包可以从 Oracle 公司的官方网站中免费下载（http://www.oracle.com/technetwork/java/javase/downloads/index.html），下载页面如图 1-1 所示。该页面中提供了当前 JDK 的最新版本和 Java 应用程序开发 IDE 平台 NetBeans with JDK（包含了 JDK 的 NetBeans）的下载链接。下载时应注意根据自己计算机中安装的操作系统选择相应的 32 位（i586）或 64 位（x86）版本。

下载并正确安装了 JDK 后还需要对其运行环境进行一些必要的配置。下面以 Windows 10 操作系统为例说明 JDK 的基本配置方法。

JDK 10 默认安装在 C:\Program Files\java\jdk-10 文件夹中，其中包含的 JRE 10 安装在 C:\Program Files\java\jre-10 文件夹中。为了使 Windows 能够找到需要执行的程序文件，需要将 JDK 的路径添加到 Windows 的环境变量中。

右击 Windows 10 桌面上“此电脑”图标，在弹出的快捷菜单中执行“属性”命令，在打开的窗口中单击左侧“高级系统设置”，在打开的对话框中单击“环境变量”按钮，在环境变量设置窗口中选择“系统变量”中的“Path”项，在图 1-2 所示的窗口中单击“新建”按钮，向环境变量列表中添加一条描述 JDK 相关文件所在位置的记录“C:\Program Files\java\jdk-10\bin”。

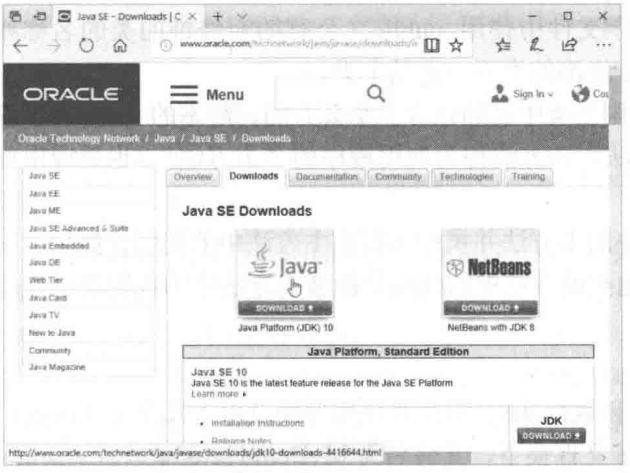

图 1-1　下载 JDK 安装包

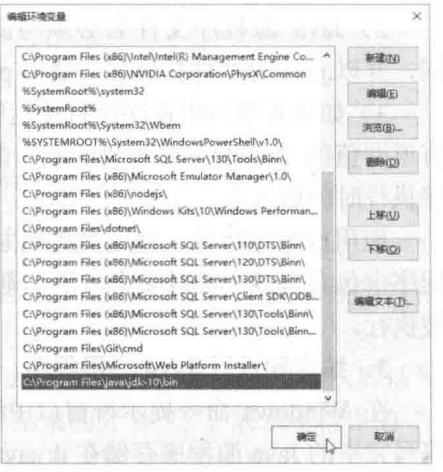

图 1-2　设置 Windows 环境变量

设置完毕后，打开 Windows 命令提示符窗口，在窗口中输入命令“java -version”后按〈Enter〉键，该命令表示要在命令提示符窗口中显示当前计算机中安装的 JDK 版本，若能在窗口中显示出图 1-3 所示的信息，则表明 JDK 的基本配置正确。

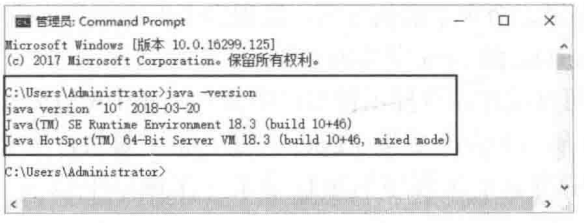

图1-3　检查环境变量设置是否正确

2．创建 Java 应用程序

创建一个 Java 应用程序最直接的方法，就是在类似于 Windows 记事本的纯文本编辑软件中直接按 Java 语言规范写出程序的源代码。图 1-4 所示的就是在 Windows 记事本中编写的，可在屏幕上显示一段文本的，一个简单 Java 应用程序。代码编写完毕后，需要将源代码文件以主类名为文件名，以.java 为文件扩展名保存。本例将源程序文件以 MyDemo.java 为文件名，保存在 d:\javacode 文件夹中。

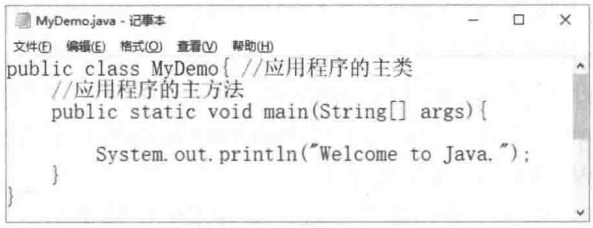

图 1-4　在 Windows 记事本中创建 Java 应用程序

需要注意以下几点：

1）在一个 Java 应用程序中至少要包含一个类（class），该类称为应用程序的主类。由于主类是执行程序的入口，所以它不能使用 private（私有的）修饰符。

2）Java 源程序文件名必须与源程序文件中使用 public（公有的）修饰的类的名称相同，并以.java 为文件扩展名。使用 public 修饰的类不一定是主类。

3）如果希望应用程序能直接运行，则主类中必须包含一个公有的、静态的（static）、没有返回值的（void）、名为“main”的方法。该方法称为应用程序的“主方法”，也是应用程序执行时的切入点。

应用程序运行时，首先要找到主类中的主方法并执行书写在主方法中的代码。如果应用程序中包含有多个类或多个方法，则其他类或方法中的代码只能被主方法中的代码调用才能被执行。

3．编译和运行 Java 应用程序

在 Windows 命令提示符窗口中编译和运行 Java 应用程序需要经过以下几个步骤（设：编写完毕的 Java 源程序存储在 d:\javacode 文件夹下，文件名为 MyDemo.java）。

1）打开 Windows 命令提示符窗口，输入“d:”后按〈Enter〉键，将当前驱动器变更为 d 盘。

2）输入“cd javacode”后按〈Enter〉键，将当前目录变更为前面已编写完成的 Java 源程序文件所在目录 d:\javacode。

3）输入“javac MyDemo.java”后按〈Enter〉键（javac 是 Java 源程序的编译命令），对 Java 源程序文件进行编译。命令正确执行后，源程序文件夹中将生成一个可被 JVM 识别并执行的、名为 MyDemo.class 的 Java 字节码文件。

4）编译完成后，可继续在命令提示符窗口中输入“java MyDemo”（这里表示的是主类名区分大小写，不带扩展名）命令后按〈Enter〉键（java 是执行字节码文件的命令），命令被正确执行后，窗口中将显示出程序的运行结果（在窗口中显示一段文字“Welcome to Java.”）。

5）主方法中的“String[] args”是一个字符串型参数数组，用于接收 Java 程序运行语句传递给主方法的若干个字符串型数据。例如，在运行 Java 程序时可以使用下列所示的命令格式向主方法传递两个数据 12 和 13。

```
java MyDome 12 13↙ //向主方法传递两个数据 12 和 13（用空格分隔），↙表示按〈Enter〉键
```

在主方法中可以使用类似下列所示的语句接收传递来的数据。

```
//显示“接收到的数据是：12 和 13”，args[0]表示数组中第 1 个数据，args[1]表示第 2 个数据
System.out.println("接收到的数据是：" + args[0] + "和" + args[1]);
```

1.2.3 Java 源程序的编写要求

在编写 Java 源程序时有以下一些注意事项。

1）Java 中所有变量、类、方法等的名称和 Java 关键字区分大小写。

2）一条语句原则上要书写在同一行中，行尾使用英文分号表示结束。

3）代码中二元操作符的左右应当各留一个空格。

4）块结构语句要使用缩进格式。

如：

好的风格

```
c = a + b;
if(a > 3){
    b = a + 1;
    System.out.println(b);
}
```

不好的风格

```
c=a+b;
if(a>3){
b=a+1;
System.out.println(b);
}
```

5）Java 源程序中的注释语句分为以“//”开头的“行注释”和以“/**”开头以“*/”结束的、可以书写在多行中的“块注释”。在程序中恰当地使用注释可有效提高程序的可读性，是一种良好的编程习惯。

6）书写大小括号时要注意它们都是成对出现的，最好输入前括号后立即输入后括号，而后再书写括号中的内容。此外，还要注意表示语句行结束的分号“;”、表示字符串的“""”和用于分隔数据列表的逗号“,”都是英文符号，要注意中英文输入法的切换。

1.3 Java 的数据类型

任何一个计算机应用程序都是在不断的数据分析、处理中实现其功能的。所以，在真正开始设计 Java 应用程序之前，首先需要建立 Java 使用的各种数据类型的概念。

计算机程序运行时的主要工作实际上是进行一系列各类数据的读取、分析、处理和输出。这些数据在程序运行过程中需要存储在计算机内存中，并以变量或常量的形式进行管理。

为了准确地为各类数据分配所需的内存空间并提供合理的运算方法，就需要将数据分为不同的类型。每种类型的数据均被预定义了能够占用的存储容量（字节数），这就意味着任何一种类型的数据都被规定了可取值的范围，超出这个范围将会导致数据的“溢出”。

1.3.1 基本类型和引用类型

Java 将数据分为基本类型和引用类型两大类。基本类型中主要包括整型、浮点型、布尔型和字符型。引用类型中最常用的有字符串型、数组、类的对象等。本节仅介绍基本类型和引用类型中的字符串类型，其他将在后续章节中介绍。

1. 基本类型

基本类型数据主要包括数值型（整型、浮点型）、字符型和布尔型 3 大类。

（1）整型

整型数据包括正整数、负整数和零。Java 中将整数分为 byte（字节型）、short（短整型）、int（整型）和 long（长整型）4 种类型，每种类型可占用不同大小的存储空间。不同的整型表示的数值范围也不同，这为开发人员根据实际需要进行灵活选择提供了方便，同时设置适当的类型对节约系统资源，提高程序运行效率也是十分重要的。常用整数类型、占用的存储空间及取值范围见表 1-1。

表 1-1 整型数据及取值范围

数据类型	占用存储空间（字节）	取值范围
byte（字节型）	1	-2^7(−128)～2^7-1(127)
short（短整型）	2	-2^{15}(−32768)～2^{15}-1(32767)
int（整型）	4	-2^{31}(−2147483648)～-2^{31}-1(2147483647)
long（长整型）	8	-2^{63} (−9223372036854775808)～2^{63} -1(9223372036854775807)

需要说明以下两点：

1）从表 1-1 中可以看出，就算是数值范围最大的 long 类型，也会有无法表示的超小或超大的整数。此时，应当使用 Java 提供的 BigInteger 类来处理数据。

2）Java 将一个整数默认为 int 类型。若需要将一个整数表示为 long 类型时，则需要在数字的后面加上 l 或 L（大写或小写的 L）。

（2）浮点型

浮点型用于表示一个实数（既有整数又有小数的数），浮点型数据分为标准计数法（如 3.0、4.156 等）和科学计数法（如 123.45 表示为 1.2345E+2）两种。

Java 根据所需数值范围不同将浮点型数据分为 float（单精度）和 double（双精度）两种类型，它们占用的字节数和取值范围见表 1-2。

表 1-2 浮点型数据及取值范围

数据类型	占用存储空间（字节）	取值范围
float（单精度浮点型）	4	负数范围：-3.408235E+38～-1.4E-45 正数范围：1.4E-45～3.408235E+38
double（双精度型浮点型）	8	负数范围：-1.7976931348623157E+308～-4.9E-324 正数范围：4.9E-324～1.7976931348623157E+308

需要说明以下两点：

1）无论是 float 还是 double 都是一种近似数据。其中，float 型最多能有 7 位有效数字，能保证的精度为 6 位，也就是说 float 的精度为 6～7 位有效数字；double 类型最多能拥有 16

位有效数字，能保证的精度为 15 位，也就是说 double 的精度为 15～16 位。在使用浮点数进行高精度计算时应注意上述问题。例如，下列语句执行后的输出结果不是 2.7，而是 2.6999999999999997。

```
System.out.println(3.0 - 0.1 - 0.1 - 0.1);        //输出结果为 2.6999999999999997
```

若需要高精度的计算，可使用 Java 提供的 BigDecimal 类。

2）Java 将一个浮点数默认为 double 类型，若希望表示一个 float 类型数值，则应在数值的后面加上一个 f 或 F。

（3）布尔型

布尔型（boolean）也称为逻辑型，用来表示一个布尔值。布尔型数据的取值只能是 true（真）或 false（假），占用 1 个字节的存储空间，通常用来表示一个关系表达式（如 a > b）或逻辑表达式（如 a > b & a < c）的运算结果。

（4）字符型

字符型（char）用来存储单个字符，占用两个字节的存储空间。Java 语言中的字符采用的是 Unicode 字符集编码方案，每个字符占用两个字节的存储空间（16 位无符号整数），共包含有 65536 个字符。由于 Unicode 字符集支持英文、中文等多国文字，所以也被称为“万国码”。在 Java 中 Unicode 码用“\uxxxx”表示，前面的“\u”表示这是一个 Unicode 值，后面 xxxx 是 4 位 16 进制数。Unicode 码的范围是\u0000～\uFFFF。

2．String 类型

前面介绍过的基本数据类型在计算机内存中保存的是数值本身，而引用数据类型在内存中存储的则是数值的内存地址，它往往由多个基本数据组成。因此，常将对引用数据类型的引用称为“对象引用”，引用数据类型也被称为复合数据类型，在有的程序设计语言中将其称为“指针”。引用数据类型中最常用的有字符串类型、数组和类的对象等。

字符串类型（String）表示一个由若干字符组成的字符序列。严格地讲，String 是 Java 的一个预定义类，字符串类型数据实际上是 String 类的一个实例化对象中存储的数据。例如：

```
String str = new String();      //声明一个 String 类的对象 str
str = "Welcome to Java.";       //为 str 对象赋值
```

上面两条语句可以简化为：

```
String str = "Welcome to Java.";
```

1.3.2 变量与常量

对用户来说，变量是用来描述一条信息的名称，在变量中可以存储各种类型的信息。而对计算机来说变量代表一个存储地址，变量的类型决定了存储在变量中的数据的类型。简单地讲，变量就是在程序运行过程中，其值可以改变的数据。程序是通过变量的名称来访问相应内存空间的。

常量存储的是在程序运行中不能被修改的固定值。Java 中常量与变量相同，也分为整型、浮点型、布尔型、字符型和字符串型。