21世纪高等院校计算机基础系列教材

Java 程序设计

Java Program Design

教程

邱仲潘　朱诗兵　朱小谷　编著

红旗出版社

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

21 世纪高等院校计算机基础系列教材

Java 程序设计

Java Program Design

教程

邱仲潘　朱诗兵　朱小谷　编著

红旗出版社

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

图书在版编目（CIP）数据

Java 程序设计教程/邱仲潘，朱诗兵，朱小谷编著.
—北京：红旗出版社，2005.3
（21 世纪高等院校计算机基础系列教材）
ISBN 7-5051-1119-1

Ⅰ.J… Ⅱ. ①邱…②朱…③朱… Ⅲ.Java 语言
—程序设计—高等学校—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 124216 号

内 容 简 介

自问世以来，Java 以其独特的优势迅速风靡了计算机界。经过数年的发展，它已日益显现出巨大的优势和潜力，成为当今主流的编程语言。本书共分为 13 章，分别介绍了 Java 基础入门、基本数据类型、Java 运算符与表达式、流程控制、面向对象程序设计、类的继承与多态、数组、字符串的处理、图形用户界面设计、图形处理、异常处理、多线程程序设计、网络编程等内容。另外，配以大量的习题供读者练习。

本书语言流畅，示例丰富，针对所阐述的理论列举了比较典型的实例，便于读者学习、掌握。本书可作为大学、高等职业技术教育教材，也可供从事计算机软件开发的读者学习使用。

需要本书或技术支持的读者，请与北京中关村 083 信箱（邮编：100080）发行部联系，电话：010-82702660，82702658，62978181（总机）转 103 或 238 传真：010-82702698 E-mail：tbd@bhp.com.cn。

系 列 名	21 世纪高等院校计算机基础系列教材
书 名	Java 程序设计教程
编 著	邱仲潘 朱诗兵 朱小谷
总 策 划	北京希望电子出版社
责任编辑	刘海芳 雷 铎
出 版	红旗出版社 北京希望电子出版社
发 行	北京希望电子出版社
地 址	红旗出版社 北京市沙滩北街 2 号（100727） 电话：（010）64037138 北京希望电子出版社 北京市海淀区上地三街 9 号金隅嘉华大厦 C 座 610
经 销	各地新华书店 软件连锁店
排 版	希望图书输出中心 王春桥
印 刷	北京双青印刷厂
版次/印次	2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷
开本/印张	787 毫米×1092 毫米 1/16 16 印张 362 千字
印 数	1～5000 册
书 号	ISBN 7-5051-1119-1
定 价	22.00 元

总　序

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。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。

一本好书，是人生前进的阶梯；一套好教材，就是教学成功的保证。为缓解计算机技术飞速发展与计算机教材滞后落伍的矛盾，我们通过调查多所院校的师生，并多次研讨，根据读者认识规律，开创出一种全新的方式，打破过去介绍原理——理论推导——举例说明的模式，增加实用操作性，通过上机实验与课上内容结合来增强可读性，用通俗易懂的语言和例子说明复杂的概念。

本套教材的特点一是“精”，精选教学内容；二是“新”，捕捉最新资讯；三是“特”，配备电子课件，力争达到基础性、先进性、全面性、典型性和可操作性的最大统一。

为保证教材质量，我们同时聘请了一批学术水平较高的知名专家、教授作为本套教材的主审和编委。全套教材包括必修课教材20多种，选修课教材和学习配套用书10余种，基本上涵盖了目前高等院校（含高等职业技术学院、高等专科学校、成人高等学校）计算机科学与技术专业所必修或选修的内容。各种教材编写时既注意到内容上的连贯性，又保证了教学上的相对独立性。

本套教材在内容的组织上，大胆汲取当今计算机领域最新技术，摒弃了传统教材中陈旧过时的内容。这些变化在各本教材中都得到了不同程度的体现。本套教材编写时既参照了教育部有关计算机科学与技术专业的教学要求，又参考了“程序员考试大纲”和“全国计算机水平等级考试大纲”的内容，因此既适合作为高等学校计算机科学与技术专业教材，也可作为计算机等级考试学习用书。

考虑到各校教学特点和计算机设备条件，我们本着“学以致用”的理念，在本套教材编写中自始至终贯彻“由浅入深，理论联系实际”的原则，以阐明要义为主，辅之以必要的例题、习题和上机实习，能够使学生尽快领悟和掌握。

在本套教材编写过程中，作者们付出了艰辛的劳动，教材编委会的各位专家、教授进行了认真的审定和悉心的指导。书中参考、借鉴了国内外同类教材和专著，在此一并表示感谢。

我们希望更多的优秀教师参与到教材建设中来，真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议。

若有投稿或建议，请发至本丛书出版者电子邮件：textbook@bhp.com.cn

21世纪高等院校计算机教材编委会

目　录

第 1 章　Java 基础入门

本章导读：

本章介绍了 Java 语言的发展史以及其特点，并对适合初学者的 Java 编程环境 J2SDK 的安装和相关环境变量的设置进行详细介绍，最后举例说明了 Java 的两类应用程序开发过程。

本章的主要内容：

- Java 语言概述
- Java 语言的特点
- Java 程序开发与运行环境
- Java 的两类应用程序开发过程举例

1.1　Java 语言概述

1.1.1　Java 语言的发展及应用前景

Java 语言是一种面向对象的程序设计语言，诞生于 1991 年，它的的设计起源于 SUN 公司的一个叫 GREEN 的项目，该项目最初的目的是为家用电器开发一个分布式的系统，这样我们就可以把 E-mail 发给家用电器，从而对它们进行控制，和它们进行信息交流。开始时，该项目组成员准备采用 C++,但 C++太复杂，而且安全性也差，因此最后决定基于 C++开发一种新的语言 Oak（一种橡树的名字）。Oak 是一种用于网络的精巧而安全的语言，在 1991～1993 年期间，Oak 一直被认为是开发消费类电子产品和交互式电视控制器的利器。但是，Sun 公司曾依此投标一个交互式电视项目，但结果是被 SGI 打败。可怜的 Oak 几乎无家可归，恰巧这时 Mark Ardreesen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的支持，触发了 Java 进军 Internet。随着全球信息网的推波助澜，这个功能强而且新颖的语言在全球信息网舞台上发出璀璨的光芒，正如广受全球咖啡族喜爱的爪哇咖啡一样，爪哇（Java）语言的魅力也迅速得到了肯定，成为在全球信息网上的超级语言，在短期内更是无人能出其右。据说 Java 的取名也有一趣闻，有一天，几位 Java 成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java（爪哇）咖啡，有一人灵机一动说就叫 Java 怎样，得到了其他人的赞赏，于是，Java 这个名字就这样传开了。

随着 Internet 的飞速发展与 WWW 的应用的快速增长，给 Java 语言带来无限生机，用 Java 编写的浏览器 HotJava 以及 Applet 在 Web 上的应用，使其成为 Internet 上最受欢迎的开发与编程语言。很多软件业的大公司也赶着这股潮流纷纷取得 Java 的使用权而进行开发各种产品。目前，全球有 67%的大型企业在采用 Java 开发自己的信息系统，Java 已经进入主流计算模式。其强大的图形、图像、动画、音频、视频、多线程和网络交互能力，使它在设计交互

式、多媒体网页和网络应用方面大显身手，成为当今推广最快的计算机编程语言。

1.1.2 Java 语言的特点

Java 语言是有 C++语言发展而来的，是一种纯面向对象的程序设计语言，作为当今最流行的一种程序设计语言，其主要有以下几个特点：

1. 简单

Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。其简单性主要体现在以下几个方面：

- Java 语言与 C++语言的风格极为相似，但比 C++语言简单。因此从某种意义上讲，Java 语言是 C 及 C++语言的一个变种，C++程序员可以很快就掌握 Java 编程技术。
- Java 语言摒弃了 C++中容易引发程序错误的地方，如指针和内存管理。
- Java 提供了丰富的类库，从而使编程工作变得更加简单。

2. 面向对象

面向对象是 Java 最重要的特性。传统的面向过程的思维方式并不符合人类习惯的思维方式，因此用面向过程方法开发软件的方法与过程不同于人类认识世界解决为题时习惯采用的方法与过程，面向对象程序设计开发模拟人类习惯的解题方法，用对象分解取代了面向过程思维方式中的功能分解，即把程序分解成许多对象，不同对象之间通过发送消息向对方提出服务要求，接受消息的对象主动完成指定功能，程序中所有对象分工协作，共同完成整个程序的功能。Java 的面向对象的特性代表了计算机程序设计的新颖的思维方法，成为解决目前软件开发面临的困难的最有效工具之一。

3. 分布式

Java 提供了一个支持 TCP/IP 协议的子库，可以处理像 HTTP 和 FTP 这样的协议。Java 应用程序可以通过 URL 来打开并访问网络上的对象，就像访问本地文件系统一样简单。Java 支持 WWW 客户/服务器计算模式，因此，Java 可利用一个特定的 URL 对象来打开并访问具有相同 URL 地址的对象，Java 的 Applet 小应用程序可以从服务器下载到客户端运行，从而实现了数据分布与操作分布。程序员可以充分利用 Java 提供的有关网络的类库进行网络程序设计，方便地实现 Java 的分布式特性。

4. 健壮性

传统的程序设计语言可以任意访问计算机的全部内存，导致程序可能修改内存中任意数据。Java 语言不支持指针，因而也就杜绝了内存的非法访问。用 Java 语言编写的程序只能访问内存中允许它们访问的部分，从而保证了内存中数据的安全性。Java 语言提供了自动垃圾收集器来进行内存管理，排除了人为因素，减少了内存出错的可能性。

同时，Java 语言通过面向对象的方法来解决程序运行过程中发生的异常事件，如除 0 溢出、数组越界、文件找不到等。大大加强了程序的健壮性。此外，Java 是强类型的语言，要求显式的方法声明，从而保证了编译器可以发现方法应用的错误，保证程序更加可靠。

5. 可移植性

Java 没有与具体环境有关的概念，Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行，这为 Java 应用软件的移植打下良好的基础。但仅仅如此还是远远不够的。同其他程序设计语言不同，Java 语言中基本数据类型设计不依赖于具体实现。通过定义独立于平台的基本数据类型及其运算，Java 数据得以在任何硬件平台上保持一致，如在 Windows 3.1 中整数（Integer）为 16bits，在 Windows 95 中整数为 32bits，在 DEC Alpha 中整数为 64bits，在 Intel 486 中为 32bits。Java 语言的基本数据类型及其表示方式如下：byte 8-bit 二进制补码，short 16-bit 二进制补码，int 32-bit 二进制补码，long 64-bit 二进制补码，float 32-bit IEEE 754 浮点数，double 32-bit IEEE 754 浮点数，char 16-bit Unicode 字符。

在任何 Java 解释器中，数据类型都是依据以上标准具体实现的。Java 运算系统的编制依据 POSIX 方便移植的限制，用 ANSI C 语言写成。Java 语言规范中没有任何“同具体实现相关”的内容。

6. 安全性

Java 虽然源于 C++，但它摒弃了 C++中很多不可靠因素。首先，Java 不支持指针，杜绝了内存的非法访问；其次，Java 的自动回收内存单元这一机制可有效防止内存丢失等动态内存分配导致的问题；第三，Java 解释器运行时进行检查，可以发现数组和字符串访问越界等问题。

由于 Java 是目前功能最强大的网络编程语言，因此，安全性是首要的。Java 字节码进入解释器时，要经过字节码校验器的检查；然后 Java 解释器在决定程序中类的内存布局；最后，类装载器将本机类与网络资源类分开，从而避免了应用程序之间的干扰。另外，客户端还可以对从网络上装载的类所访问的文件系统作出限制。这些机制，都保证了 Java 的安全性。

7. 解释型

Java 是解释型的语言，用 Java 语言编写的程序经编译后生成的是 Java 字节码（.class 文件）。在运行时，借助 Java 语言运行系统（Java 虚拟机 JVM），.class 文件是 JVM 中可执行的文件格式。Java 虚拟机规范为不同的硬件平台提供了不同的编译代码规范，从而使 Java 程序独立于平台；然后，Java 解释器负责将 Java 字节码文件（.class 文件）一边解释，一边运行。为了提高运行速度，Java 提供了另一种解释运行方法 JIT（Just In Time），可以一次解释完，再运行。

8. 多线程

传统的程序设计语言（如 C/C++）采用了单线程体系结构，而 Java 提供了多线程支持。多线程的优点是具有更好的交互性能和实时控制性能。线程（Process）是系统分配资源的单位，可并发执行的单位。Java 中多线程的概念给程序员提供了一条可维持用户界面不被冻结而又能同时处理琐碎工作的机会。Word 就是一个多线程程序，当使用 Word 的窗口界面执行编辑动作时，同时 Word 也有其他线程正在执行其他工作，如自动存盘、拼写检查等。Java 提供了一个类 Thread，它负责启动运行，终止线程，并可检查线程状态。Java 的线程还包括一组同步原语，负责对线程实行并发控制。程序开发人员可利用 Java 提供的多线程接口，方

便地设计出多线程应用程序。当然多线程的实时控制性能还取决于运行时支持系统（UNIX,Windows,Macintosh 等）。

9. 动态

Java 的动态特性是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类，这是 C++语言进行面向对象程序设计所无法实现的。 在 C++程序设计过程中，每当在类中增加一个实例变量或一种成员函数后，引用该类的所有子类都必须重新编译，否则将导致程序崩溃。Java 编译器不是将对实例变量和成员函数的引用编译为数值引用，而是将符号引用信息在字节码中保存下传递给解释器，再由解释器在完成动态连接类后，将符号引用信息转换为数值偏移量。这样，一个在存储器生成的对象不在编译过程决定，而是延迟到运行时由解释器确定。这样，对类中的变量和方法进行更新时就不至于影响现存的代码。

1.2 Java 程序开发与运行环境

1.2.1 Java 开发环境及安装

为了支持用户的 Java 程序开发，厂家已经开发了多种 Java 程序开发工具，如 Visual J++, Jbuilder, JpadPro, TextPad 等。现在市场上有很多 Java 语言开发环境，但由于 J2SDK 比较适合初学者使用，本书中使用的是 J2SDK（Java 2 SoftWare Development Kit），Java 是由 Sun 公司开发的，因此读者可以从 SUN 公司的 Internet 站点找到这一工具并下载 ffj30_ce_ml.exe, jmf-2_1_1a-win.exe 和 j2sdk-1_4_0-win.exe，根据提示进行安装。

1.2.2 开发环境设置

下面以 J2SDK1.4.0 Windows 为例来说明开发环境设置，假设 J2SDK 安装在 C:\j2sdk1.4.0 目录下。

（1）平台为 Windows 9X 时，需要修改系统根目录下的 autoexec.bat 文件的 path 和 classpath 变量，例如：

```
set Path=%path%;c:\ j2sdk1.4.0\bin
set Classpath=.; c:\ j2sdk1.4.0\lib
```

其中，classpath 中的“.”表示在任意当前目录下均可执行 J2SDK。存盘后重新执行 autoexec.bat.即可完成设置。

（2）平台为 Windows 200/NT/XP 时，须修改环境变量，即在环境变量中增加 classpath, 并修改原有的 path,设置方法为单击“开始”|“控制面板”|“系统”|“高级”|“环境变量”，在系统变量处，找到 Path。单击“编辑”，将 c:\ j2sdk1.4.0\bin 加到后面。在单击“新建”按钮，在加入新的环境变量 classpath,classpath=.; c:\ j2sdk1.4.0\lib,单击“确定”按钮即可。

（3）设置完成后，在 DOS 窗口下，键入 Java 或 Javac 并按回车键后，如果出现用法参数提示信息，则安装正确。如果有问题，请检查路径设置是否正确。

1.3　Java 的两类应用程序开发过程举例

Java 程序有两种：Java　Application（Java 应用程序）和 Java　Applet（Java 小应用程序），下面我们通过两个例子来介绍这两种程序的开发过程并对其进行分析。

1.3.1　Java　Application 开发过程举例

开发一个 Java　Application 的过程可分为三步。

1. 编辑源程序

编辑源程序是程序开发的第一步。源程序是用 Java 程序设计语言直接编写的程序，通常是由一个或多个类组成的文件，其扩展名为.java。创建、修改、保存源程序文件的过程称为源程序文件的编辑，编辑工作可以使用任何文本编辑器来完成，如 Edit、Notepad、UltraEdit 等。对于初学者来说，使用普通文本编辑器有利于我们更专注于 Java 语言本身，而不会陷入复杂的集成工具中。编辑完成后，将源文件存为扩展名为.java 的文件。

2. 编译

编译是用 Java 编译器（Javac）将 Java 源程序翻译成 Java 虚拟机能理解的指令，并将这些指令组织为字节码文件（扩展名为.class）。如果出错，则返回源程序，改正源程序中的错误。

3. 运行

运行是 Java 虚拟机解释、运行字节码文件的过程，如果运行出错，则返回源程序（.java 文件），改正错误，再重新编译，运行。

以下举例证明开发 Java Appliation 的过程。

第一步：编辑源文件，打开记事本，写入如下所示的源程序，并存盘为 Welcome1.java。

```
1.   // Fig. 1.1: Welcome1.java
2    // A first program in Java.
3
4    public class Welcome1 {
5
6     // main method begins execution of Java application
7    public static void main( String args[] )
8    {
9        System.out.println( "欢迎来到java世界！!" );
10
11       } // end method main
12
13 } // end class Welcome1
```

以上就是一个完整的 Java 应用程序，本程序的作用是在屏幕上输出字符串"欢迎来到 java

世界!!”。程序首先用保留字 class 声明了一个新的类，类名为 Welcome1，它是一个公共类(public)。整个类定义由大括号{}括起来（4-13 行）。在该类中定义了一个 main()方法，其中 public 表示访问权限，指明所有的类都可以使用这一方法，static 指明该方法是一个类方法，它可以通过类名直接调用，void 则指明 main()方法不返回任何值。对于一个应用程序来说，main()方法是必需的，而且必须按照如上的格式来定义。Java 解释器在没有生成任何实例的情况下，以 main()作为入口来执行程序。main()方法只能有一个。String args[]是传递给 main()方法的参数，参数名为 args[]，它是类 String 的一个实例。在 main()方法的实现（大括号中），语句

```
System.out.println(“欢迎来到java世界！”);
```

是用来实现字符串的输出。//表示本行“//”后面的部分为程序的注释。

编辑完这一源程序后，把它存到一个名为 Welcome1.java 的文件中，如图 1-1 所示。

注意：文件名和类名必须相同，只是 Java 解释器的要求，公共类必须放在与其同名的文件中。

图 1-1　保存文件

保存为 Welcome1.java 文件,在文件名上加引号，可确保系统不会在文件名后再自动加上.txt 扩展名（因为我们选择的保存类型是“纯文本”），而是按双引号内的形式来保存。

第二步：编译及运行。在 Dos 平台上的整个编译及运行过程如图 1-2 所示。

键入命令：

```
javac  Welcome1.java
```

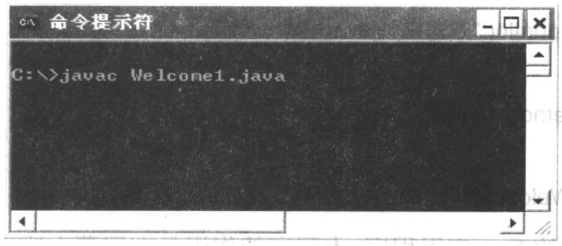

图 1-2　命令提示

如果编译成功，则会生成 Welcome1.class 文件，否则会报告错误。编译成功后，在 Dos 平台上键入命令：

```
java  Welcome1
```

就会在屏幕上出现运行结果，显示我们在程序中编写的字符串：

欢迎来到 java 世界！

如图 1-3 所示。

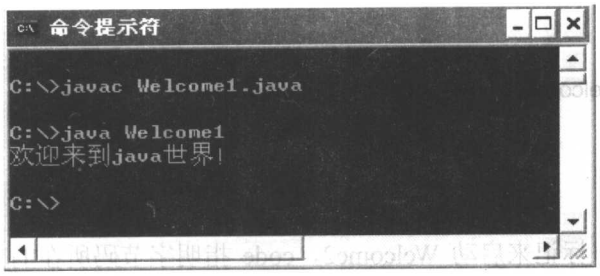

图 1-3　Welcome1.java 运行结果

1.3.2　Java　Applet 开发过程举例

开发一个 Java Applet 过程也分为三步：编辑，编译和运行。前两个步骤与 Java Application 完全相同。但由于 Applet 是一种嵌入到 HTML 的 Java 程序，所以运行之前应该先创建一个可以嵌入该 Applet 字节码的 HTML 文件，然后由支持 Java 的 WWW 浏览器或者 AppletViewer 来加载执行。

第一步：打开记事本，写入源程序并存为 Welcome2.java。

```
import java.awt*;
import java.applet.*;

    public class Welcome2 extends Applet {          //an applet
        public void paint(Graphics g){
        g.drawString ("Hello World!",20,20);

        }

    }
```

这是一个简单的 Applet（小应用程序）。程序中，import 语句输入 java.awt 和 java.applet 下所有的包，使得该程序可以使用这些包中所定义的类，然后声明一个公共类 Welcome2，用 extends 指明它是 Applet 的子类。在类中，我们重写父类 Applet 的 paint()方法，其中参数 g 为 Graphics 类，它表明当前作画的上下文。在 paint()方法中，调用 g 的方法 drawString(), 在坐标（20,20）处输出字符串 Hello World!其中坐标是用像素点来表示的。

第二步：编译及运行。

编译与例 1 相同，在 Dos 平台上键入命令：

```
javac  Welcome2.java
```

生成字节码文件 Welcome2.class，由于 Applet 中没有 main()方法作为 Java 解释器的入口，我们必须编写 HTML 文件，把该 Applet 嵌入其中，然后用 appletviewer 来运行，或在支持 Java 的浏览器上运行。它的 HTML 文件如下：

```
<HTML>
<HEAD>
<TITLE> Hello World </TITLE>
</HEAD>
<BODY>
<applet code="Welcome2.class" width=200 height=40>
</applet>
</BODY>
</HTML>
```

其中用<applet>标记来启动 Welcome2，code 指明字节码所在的文件，width 和 height 指明 applet 所占的大小，我们把这个 HTML 文件存入 Welcome2.html，然后运行。

```
C:\>appleviewer Welcome2.html
```

这时，屏幕上就出现一个小窗口，显示运行结果，如图 1-4 所示。

图 1-4　Welcome2.java 运行结果

或者再 WWW 浏览器中打开这个 HTML 页面来运行。

1.4　小　结

本章介绍了 Java 语言的背景知识及其运行环境，重点介绍了 Java 语言的特点、J2SDK 的安装和环境变量的设置，并举例说明了两种 Java 程序的不同之处。

1.5　习　题

1．为什么说 Java 语言具有可移植性？
2．Java 语言有哪些特点？
3．简述开发 Java 程序的步骤。
4．简述两类 Java 程序的不同之处。

第 2 章　基本数据类型

本章导读：

数据类型是程序设计中最基本的概念，Java 语言中的所有变量必须有一个数据类型，变量的数据类型决定了变量能存储的值的种类以及可能进行的操作方式。Java 语言提供了两种类型，一种是简单类型，如整型、布尔型等；另一种是复合数据类型，如数组、类等。另外，程序在数值运算之前，必须先将所有数值转化成同一类型。本章将对基本类型的数据作一个详细地介绍，在最后一节里，将介绍数据类型转换。

本章的主要内容：

- 数据类型综述
- 变量和常量
- 基本数值数据类型
- 非数值数据类型
- 数据类型综合应用举例
- 数据类型转换

2.1　数据类型综述

Java 语言中的数据类型和其他高级语言类似，Java 的语言规范提供了两种类型。一种是简单数据类型，如整数、浮点数、布尔型、字符型等，用来实现基本的数据类型，通常用户是不能修改的；另一种是复合数据类型，包括数组、类、接口等，是由简单数据类型复合而成的，如图 2-1 所示。

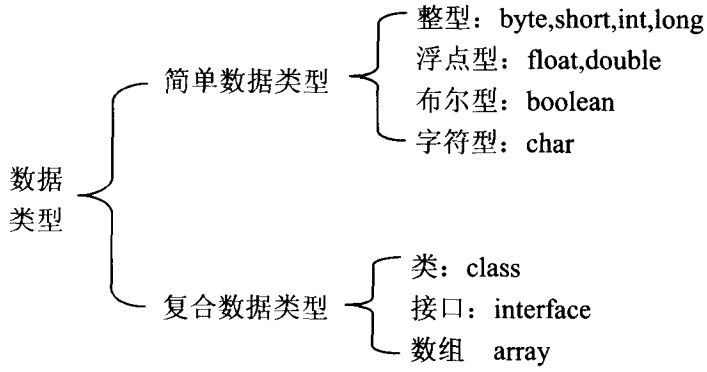

图 2-1　java 中的数据类型

类与接口又称为引用。本章将介绍 Java 语言中简单数据类型以及各数据类型的优先级和类型转换关系。

Java 语言中的类型与其他许多语言中的类型并不完全相同，Java 语言中的类型不影响存储分配，只用于控制变量的算术运算性质和值的合法范围，超出合法范围的部分被自动除去。

2.2 变量和常量

2.2.1 标识符

在 Java 语言中，标识符的首字符必须是一个字母，下划线("_")或美元符号("$")。后面的字符可以是数字也可以是字母，这里所谓的“字母”包括以下几种：

- 大写字母 A～Z
- 小写字母 a～z
- 泛代码（Unicode）中所有字符序号大于十六进制数 00C0 的所有字符

Unicode 字符集涵盖的不仅仅只有 ASCII 字符集，而且涵盖了像汉语、日语、德语等多种语言中的符号，这样，“字母”这一概念得到了很大程度的扩展。

下面举例证明。

合法字符：Txy,　_rid, $bik, var6

不合法字符：5txy　(首字符不能为数字)

　　txy#　(含有非法字符#)

　　default (保留字)

在 Java 语言中，有一部分标识符被保留作为关键字或保留字，它们有专门意义和用途，用户不能使用，下面列出了所有的关键字。

abstract						
boolean	break	byte	byvalve			
case	catch	char	class	const	continue	
default	do	double				
else	extends					
false	final	finally	float	for		
goto						
if	implements	import	instanceof	int	interface	
long	length					
native	new	null				
package	private	protected	public			
return						
short	static	super	switch	synchronized		
this	threadsafe	throw	throws	transient	true	try
void	volatile					
while						