# 第七章 一维数组

## 7.1 引言

1.要点提示：单个的数组变量可以引用一个大的数据集合。

2.这中被称作“数组”的数据结构，可以用它来存储一个元素个数固定且元素类型相同的有序集。

## 7.2 数组的基础知识

1.要点提示：一旦数组被建立，它的大小是固定的。使用一个数组引用变量，通过下标来访问数组中的元素。

### 7.2.1 声明数组变量

1.例如：下面的代码声明myList，它引用一个具有double型元素的数组。



### 7.2.2 创建数组

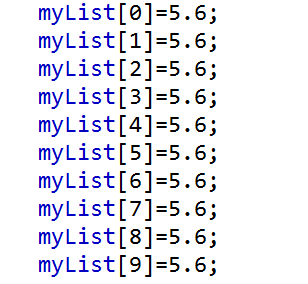
1.不同于基本数据类型的声明，声明一个数组变量时并不在内存中给数组分配任何空间，它只是创建一个对数组的引用的存储位置。

2.声明一个数组变量、创建数组、然后将数组引用赋值给变量这三个步骤可以合并在一条语句里，例如：



这条语句声明了数组变量myList，创建一个由10个double型元素构成的数组，并将该数组的引用赋值给myList。

可以通过以下方法初始化数组：



3.注意：一个数组变量看起来似乎存储了一个数组，但实际上它存储的是指向数组的引用。严格地讲，一个数组变量和一个数组是不同的，但多数情况下它们的差别是可以忽略的。因此，为了简化，通常可以说myList是一个数组，而不用更长的陈述：myList是一个含有10个double型元素数组的引用变量。

### 7.2.3 数组的大小和默认值

1.当给数组分配空间时，必须指定该数组能够存储的元素个数，从而确定数组的大小。创建数组之后就不能再创建它的大小。

2.当创建数组后，它的元素被赋予默认值，数值型基本数据类型默认值为0，char型的默认数值为‘\u0000’，boolean型的默认值为false。

### 7.2.4访问数组元素

1.数组元素可以通过下标访问。数组下标是基于0的，也就是说，其范围从0开始到arrayRefVar.length-1结束。例如，数组myList包含10个double值，而且下标从0到9。

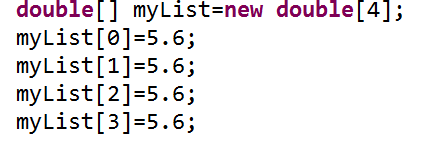
2.警告：一些语言使用圆括号引用数组元素，例如myList(9)。而Java语言使用方括号，例如myList[9]。

### 7.2.5数组初始化语法

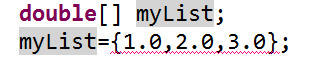
1.Java有一个简捷的标记，称作“数组初始化语法”，它使用下面的语法将声明数组、创建数组和初始化数组结合到一条语句中：



这条语句声明、创建并初始化包含4个元素的数组myList，它等价于下列语句：



2.警告：使用数组初始化语法时，必须将声明、创建和初始化数组都放到一条语句中。将它们分开会产生语法错误。因此，下面的语句是错误的：

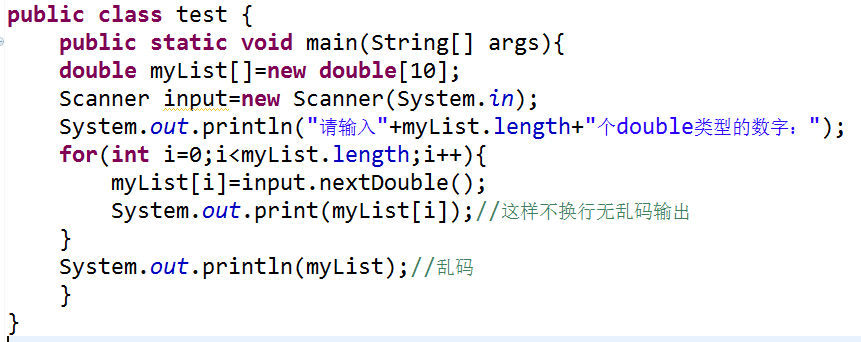


### 7.2.6处理数组

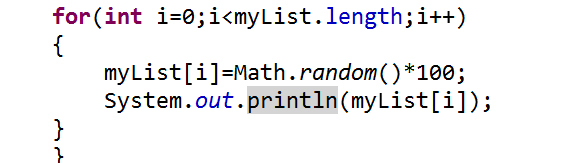
处理数组的时候，经常会用到for循环，理由有以下两点：

1. 数组中所有元素都是同一类型的。可以使用循环以同样的方式反复处理这些元素。
2. 由于数组的大小是已知的，所以很自然的使用for循环。

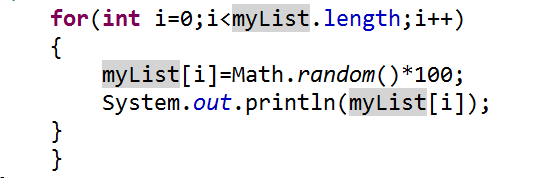
1.（使用输入值初始化数组）下面的循环使用用户输入的数值初始化数组myList。



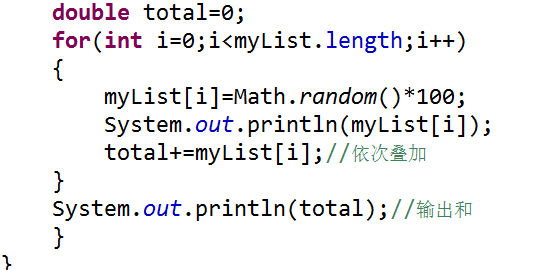
2.（使用随机数初始化数组）下面循环使用0.0到100.0之间，但小于100.0的随机值初始化数组myList。



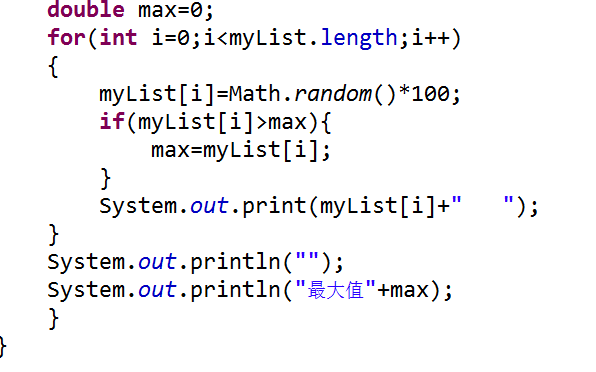
3.（显示数组）为了打印数组，必须使用类似下面的循环，打印数组中的每一个元素。



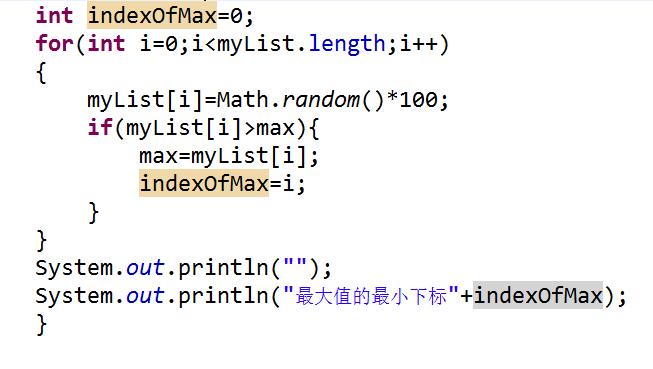
4.（对所有元素求和）使用名为total的变量存储和。Total的值初始化为0.使用如下循环讲数组中的每个元素加到total中



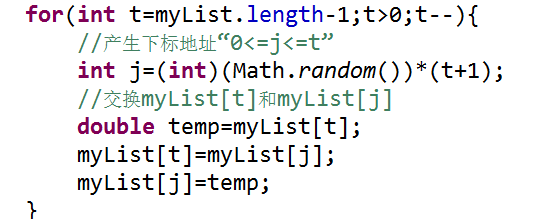
5.（找出最大元素）使用名为max的变量存储最大元素。将max的值初始化为myList[0]。为了找出数组myList中的最大元素，将每个元素与max比较，如果该值大于max则替换掉。



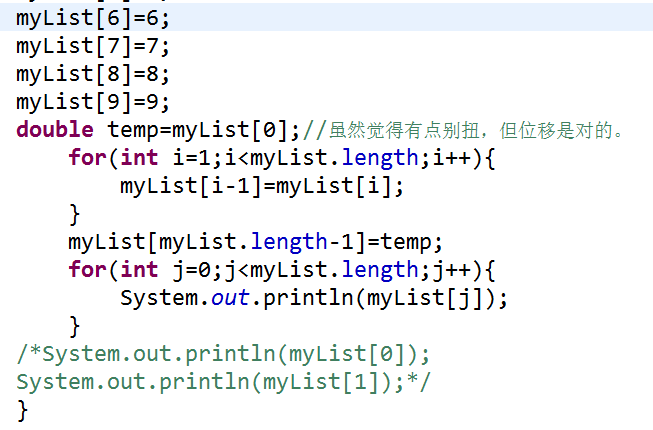
6.（找出最大元素的最小下标—不大明白，例子随机数也不符合要求，需要自定义数组）经常需要找出数组中最大元素。如果数组中含有多个最大元素，那么找到最大元素的最小下标值



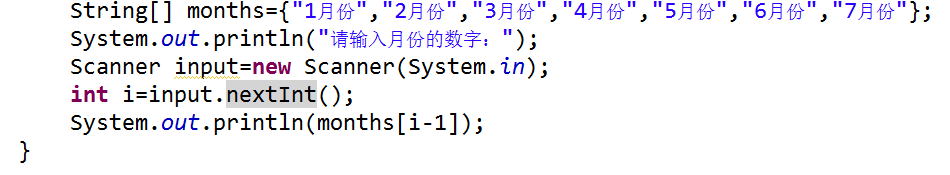
7.（随机乱打）在很多应用程序中，需要对数组中的元素进行任意的重新排序。这称作乱打。为了完成这种功能，针对每个元素myList[i]，随机产生一个下标j，然后将myList[i]和myList[j]互换。



8.（移动元素）



9.（简化编码）对于某些任务来说，数组可以极大化简化编码。例如，假如你想通过给定数字的月份来获得该月份的英文名。如果月份名称保存在一个数组中，给定月份的的月份英文名可以通过下标来实现。下面的代码提示用户输入一个月份然后显示月份名（而不用条件语句实现）

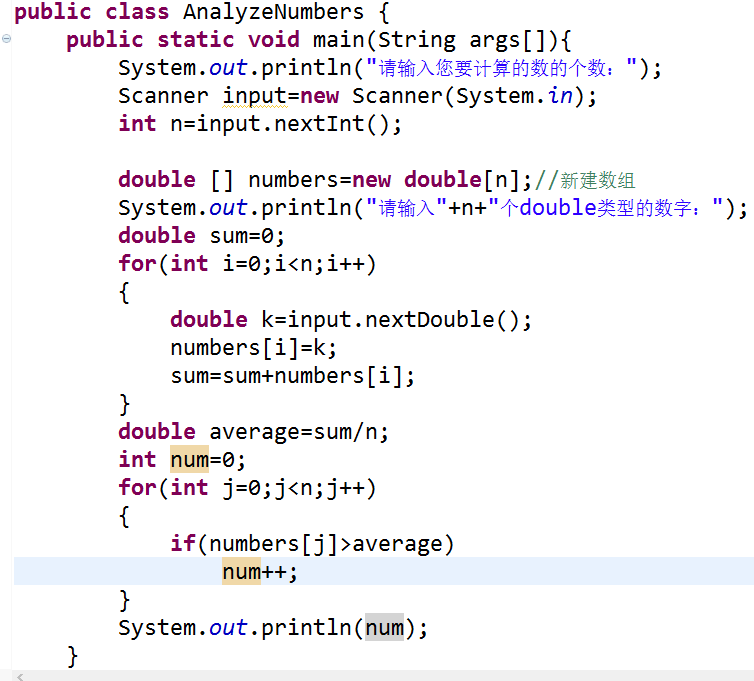


### 7.2.7 foreach循环

1.Java支持一个简便的for循环，称为foreach循环，即不使用下标变量就可以顺序的遍历整个数组。（略）

## 7.3 示例学习：分析数字

要点提示：编写一个程序，找到大于所有平均值的那些项。



注意：新建数组

## 7.4 示例学习：一副牌

编写一个程序，从一副牌中随机选出四张牌。

0-12表示13张黑桃；

13-25表示13张红桃；

26-38表示13张方块；

39-51表示13张梅花；

CardNumber/13决定牌的花色，cardNumber%13决定具体花色中哪张牌。



## 7.5数组的复制

1.要点提示：要将一个数组中的内容复制到另一个中，你需要将数组的每个元素复制到另外一个数组中。在程序中经常需要复制一个数组或数组的一部分。在这种情况下，你可能会尝试使用赋值语句（=），如下所示：

List2=list1;

该语句并不能将list1引用的数组内容复制给list2，而只是将list1的引用值复制给list2。在这条语句之后，list1和list2都指向同一个数组，list2原先所引用的数组不能再引用，就变成了垃圾，会被Java虚拟机自动回收（这个过程称为垃圾回收）。

2.在Java中，可以使用赋值语句复制基本数据类型的变量，但是不能复制数组。将一个数组变量赋值给另一个数组变量，实际上是将一个数组的引用复制给另一个变量，使两个变量都指向相同的内存地址。

3.复制数组有三种方法：

（1）使用循环语句逐个的复制数组的元素；

（2）使用System类中的静态方法arraycopy（该方法命名不当，应大写）

（3）使用clone方法复制数组；

## 7.6 将数组传递给方法

1.匿名数组

new int[ ]{1,2,3,4,5,6,7}，该数组没有显式的引用变量，这样的数组称为“匿名数组”。

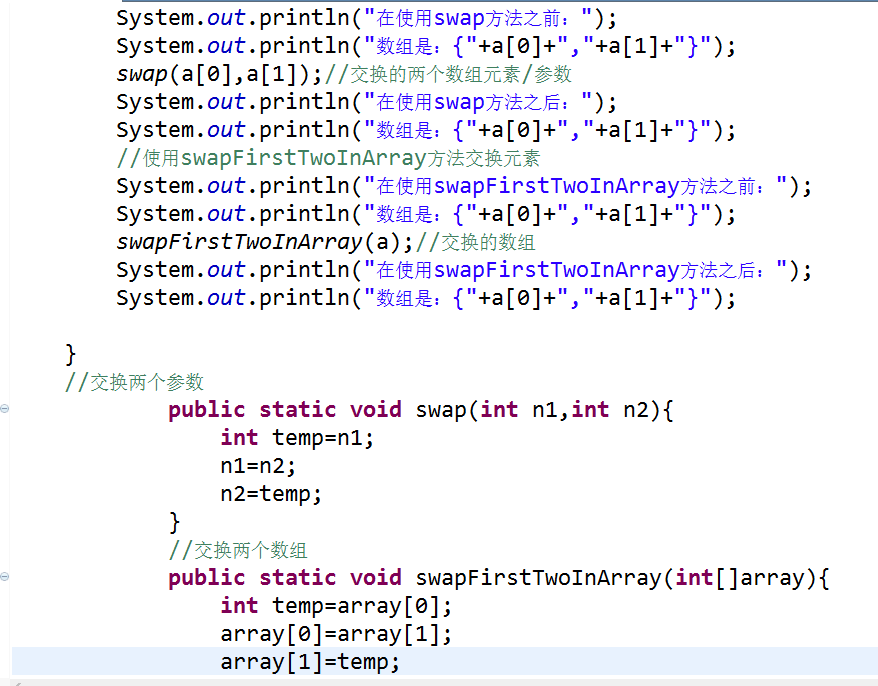
2.Java使用按值传递的方式将实参传递给方法。传递基本数据类型变量的值与传递数组的值有很大的不同。

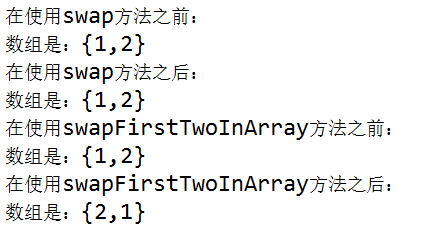
（1）对于基本数据类型参数，传递的是实参的值。

（2）对于数组类型参数，参数值是数组的引用，给方法传递的是这个引用。

3.JVM将对象存储在一个称作堆得内存区域内，堆用于动态内存分配。

4.程序：用于说明“传递基本数据类型值与传递数组引用变量方法的不同之处”





结果：

使用swap方法没能对换两个元素，但是，swapFirstTwoInArray方法就实现了对换。

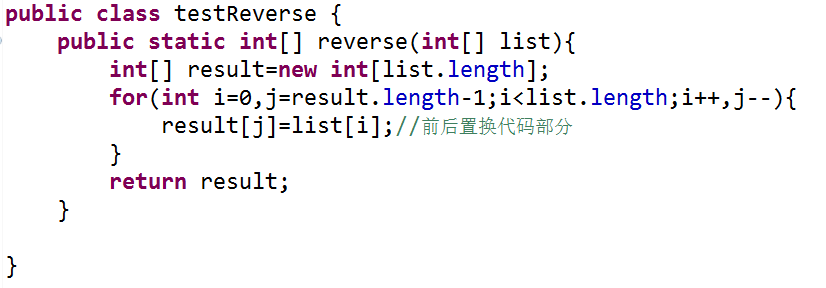
因为swap方法中的参数为基本数据类型，所以调用swap（a[0],a[1]）时，a[0]和a[1]的值传给了方法内部的n1和n2。N1和n2的内存位置独立于a[0]和a[1]的内存位置。这个方法的调用没有影响到数组的内容。

SwapFirstTwoInArray方法的参数是一个数组。数组指向的地址是数值。

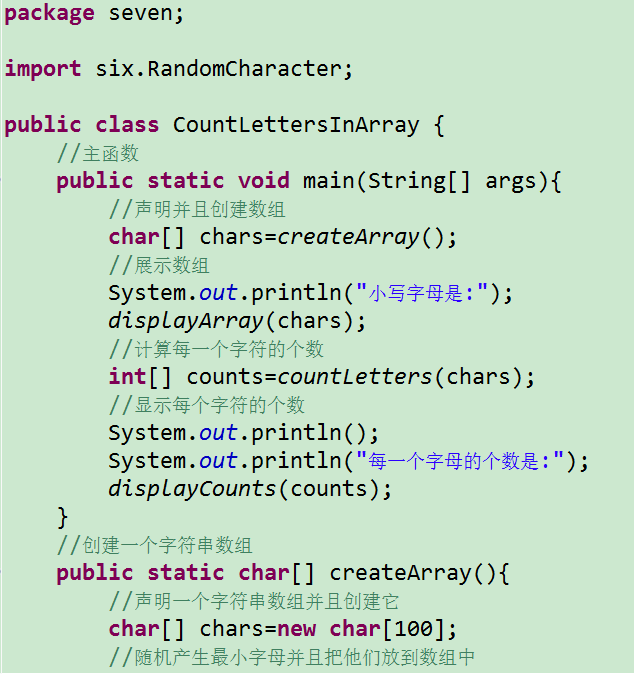
归根到底是内存地址指向的问题。

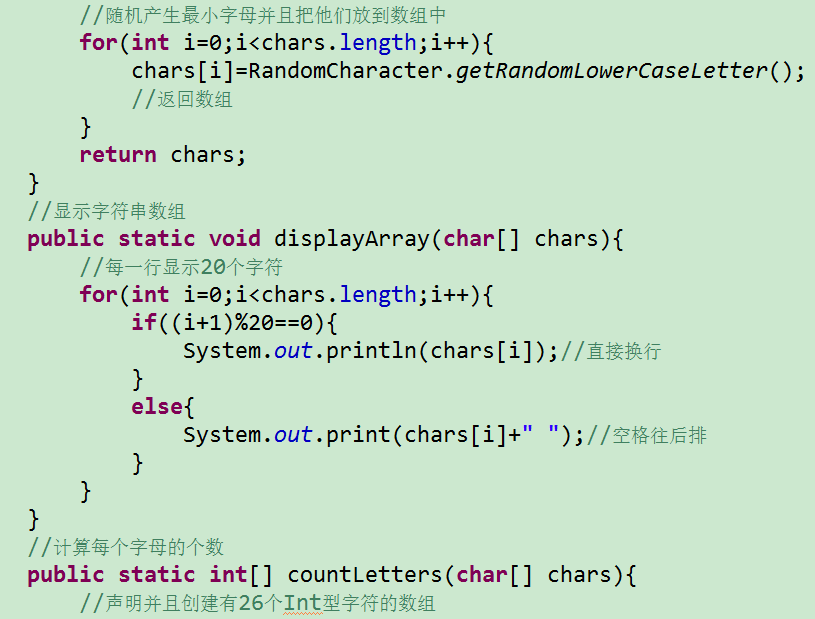
## 7.7 从方法中返回数组

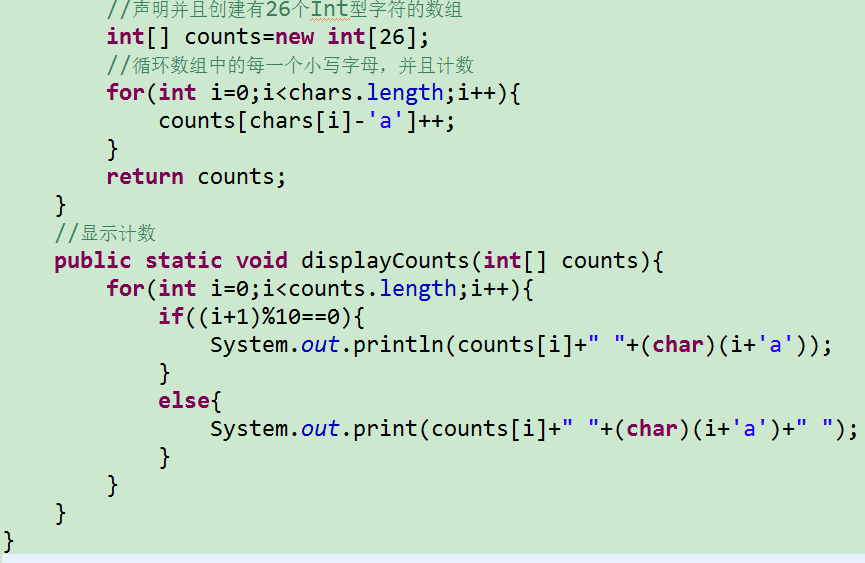
返回一个数组，返回值的形式返回。

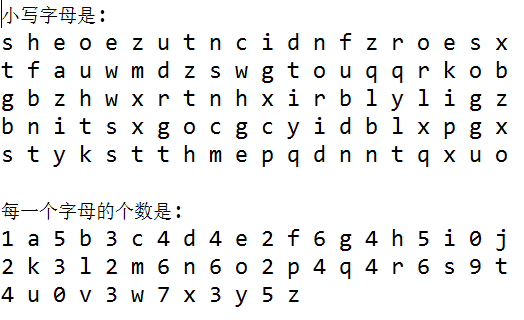


## 7.8 示例学习：统计每个字母出现的次数





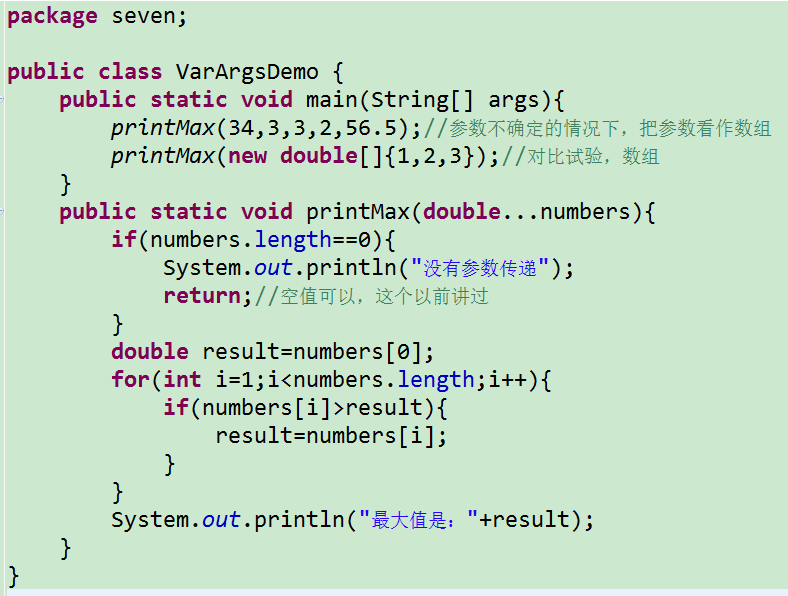




经验：输出矩阵或者数列时。整除，println换行；不整除，print+“ ”。

## 7.9 可变长参数列表

1.Java将可变长参数当成数组对待。

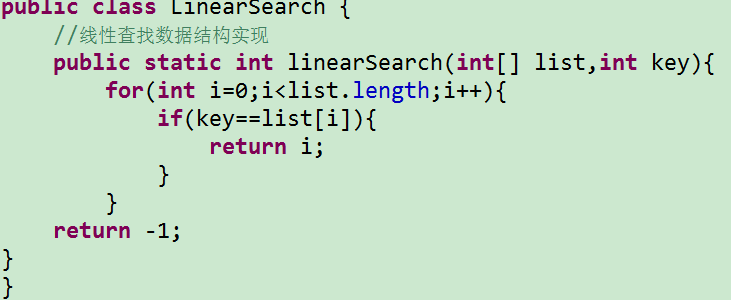


## 7.10 数组的查找

1.要点提示：如果一个数组排序好了，对于寻找数组中的一个元素，二分查找比线性查找更加高效。（数据结构相关内容）

### 7.10.1 线性查找法

1.简述：线性查找法将要查找的关键字key与数组中的元素逐个进行比较。这个过程持续到在列表中找到与关键字匹配的元素，或者查完列表也没找到关键字为止。如果匹配成功，线性查找法返回与关键字匹配的元素在数组中的下标。如果没有匹配成功，则返回-1。



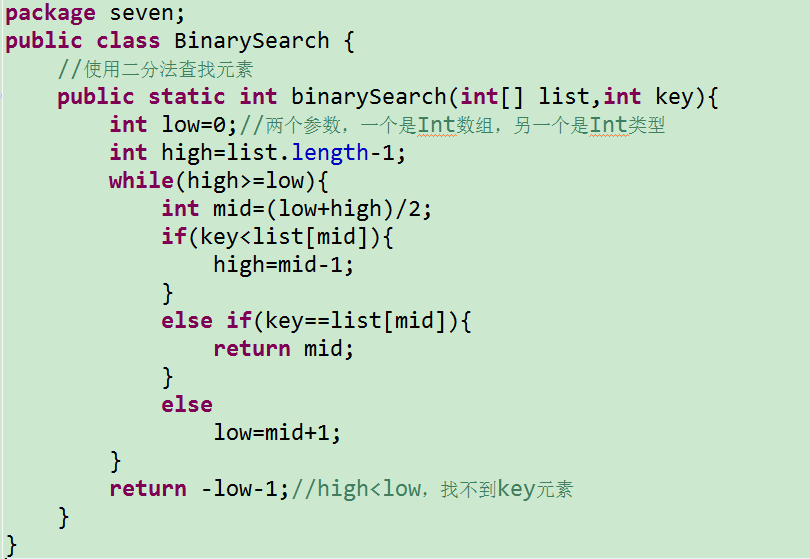
### 7.10.2 二分查找

1.简述：二分查找法是另一种常见的对数值列表的查找方法。使用二分查找的前提条件是数组中的元素必须已经排好序。假设数组已经按照升序排列。二分查找法首先将关键字与数组的中间元素进行比较。考虑下边三种情况：

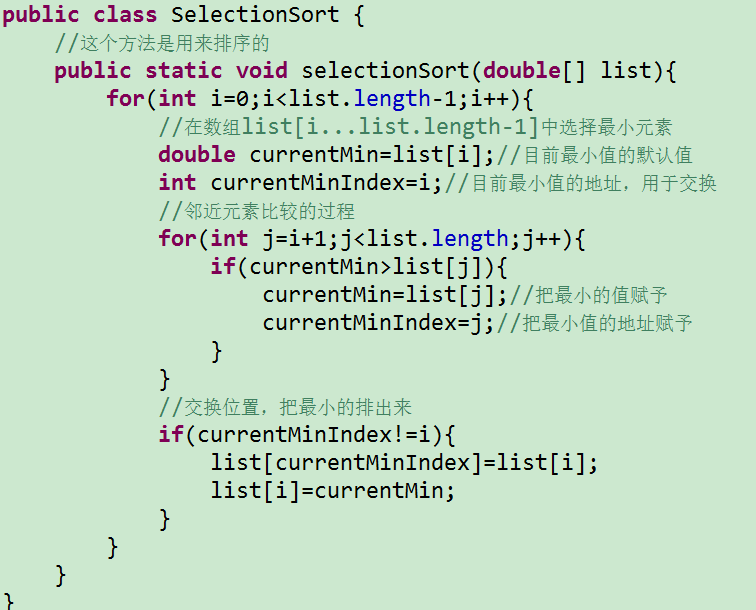
（1）如果关键字小于中间元素，需要在数组前一半元素中继续查找关键字。

（2）如果关键字和中间元素相等，则匹配成功，查找结束。

（3）如果关键字大于中间元素，只需要在数组的后一半元素中继续查找关键字。



## 7.11 数组的排序



## 7.12 Arrays类

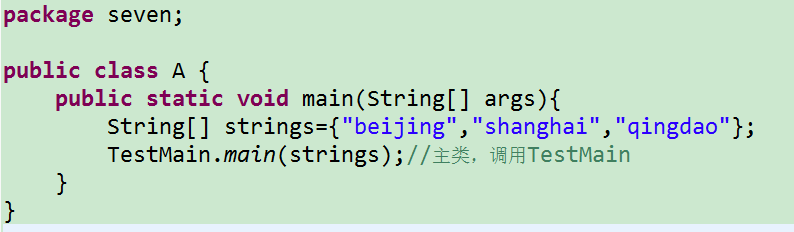
1.要点提示：Java.util.Arrays类包含了一些实用的的方法用于常见的数组操作，比如排序和查找。

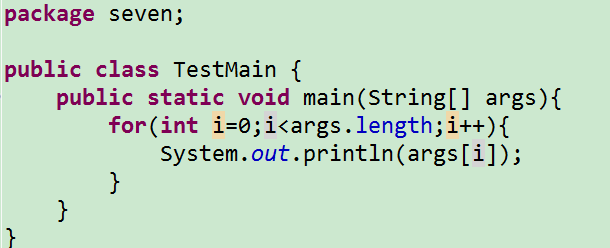
2.Java.util.Arrays类包括各种各样的静态方法，用于实现数组的排序和查找、数组的比较和填充数组元素，以及返回数组的字符串表示。这些方法都有对所有基本类型的重载方法。

## 7.13命令行参数

1.要点提示：main方法可以从命令行接收字符串参数。

你可以注意到main方法的声明与众不同，它具有String[]类型参数args。很明显，参数args是一个字符串数组。Main方法就像一个带参数的普通方法。可以通过传递实参来调用一个普通方法。





主函数其实是定义了一个数组，还可以传参数。

### 7.13.1向main方法传递字符串

### 7.13.2示例学习：计算机

命令行计算机。

