链滴

Java 数组的常用操作

作者：chenyuan995

原文链接：https://ld246.com/article/1587546309298

来源网站：链滴

数组遍历

- 数组遍历： 就是将数组中的每个元素分别获取出来，就是遍历。遍历也是数组操作中的基石。

```
public static void main(String[] args) {
    int[] arr = { 1, 2, 3, 4, 5 };
    for (int i = 0; i < arr.length; i++) {
        System.out.println(arr[i]);
    }
}
```

数组获取最大值元素

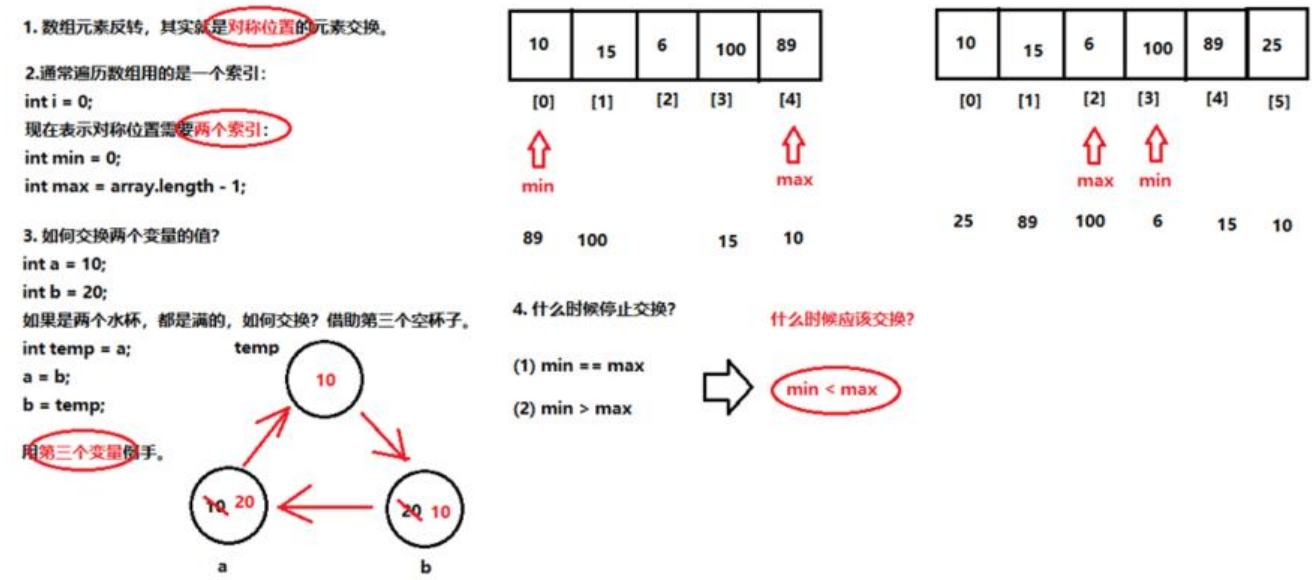
- 实现思路：
    - 定义变量，保存数组0索引上的元素
    - 遍历数组，获取出数组中的每个元素
    - 将遍历到的元素和保存数组0索引上值的变量进行比较
    - 如果数组元素的值大于了变量的值，变量记录住新的值
    - 数组循环遍历结束，变量保存的就是数组中的最大值

```
public static void main(String[] args) {
    int[] arr = { 5, 15, 2000, 10000, 100, 4000 };
    //定义变量，保存数组中0索引的元素
    int max = arr[0];
    //遍历数组，取出每个元素
    for (int i = 0; i < arr.length; i++) {
        //遍历到的元素和变量max比较
```

```
        //如果数组元素大于max
        if (arr[i] > max) {
            //max记录住大值
            max = arr[i];
        }
    }
    System.out.println("数组最大值是： " + max);
}
```

数组反转

- 实现思想：数组最远端的元素互换位置。
  - 实现反转，就需要将数组最远端元素位置交换
  - 定义两个变量，保存数组的最小索引和最大索引
  - 两个索引上的元素交换位置
  - 最小索引++，最大索引--，再次交换位置
  - 最小索引超过了最大索引，数组反转操作结束



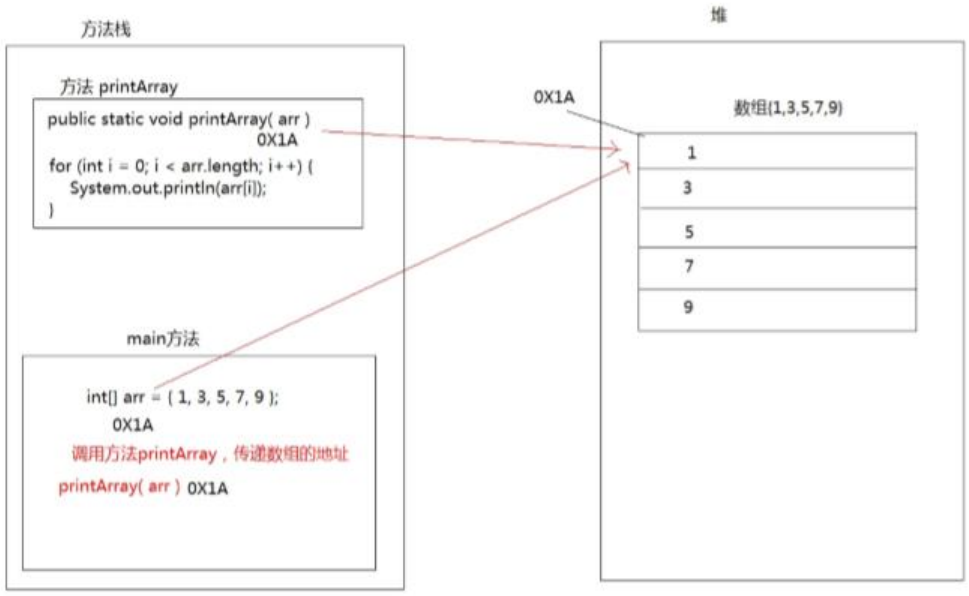

```
/*
数组元素的反转：
本来的样子：[1, 2, 3, 4]
之后的样子：[4, 3, 2, 1]
要求不能使用新数组，就用原来的唯一一个数组。
 */
public class Demo07ArrayReverse {
    public static void main(String[] args) {
        int[] array = { 10, 20, 30, 40, 50 };
        // 遍历打印数组本来的样子
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i]);
        }
        System.out.println("============");
        /*
```

```
        初始化语句：int min = 0, max = array.length - 1
        条件判断：min < max
        步进表达式：min++, max--
        循环体：用第三个变量倒手
         */
        for (int min = 0, max = array.length - 1; min < max; min++, max--) {
            int temp = array[min];
            array[min] = array[max];
            array[max] = temp;
        }
        // 再次打印遍历输出数组后来的样子
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i]);
        }
    }
}
```

数组作为方法参数

- 数组作为方法参数传递，传递的参数是数组内存的地址。

```
public static void main(String[] args) {
    int[] arr = { 1, 3, 5, 7, 9 };
    //调用方法，传递数组
    printArray(arr);
}
/*
创建方法，方法接收数组类型的参数
进行数组的遍历
*/
public static void printArray(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        System.out.println(arr[i]);
    }
}
```

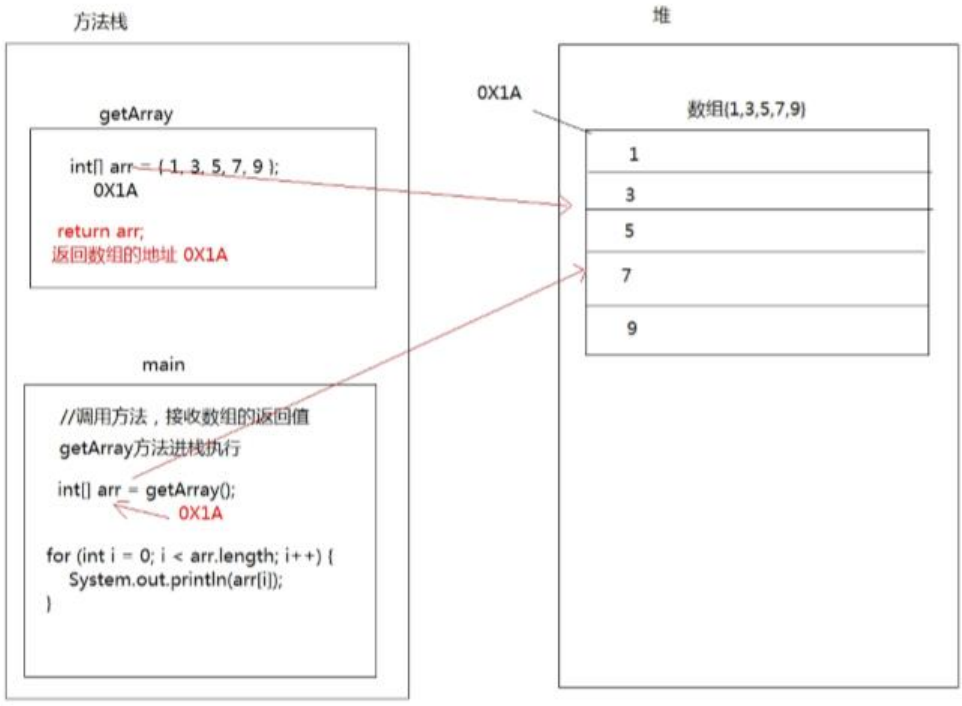


原文链接：Java 数组的常用操作

数组作为方法返回值

- 数组作为方法的返回值，返回的是数组的内存地址

```
public static void main(String[] args) {
    //调用方法，接收数组的返回值
    //接收到的是数组的内存地址
    int[] arr = getArray();
    for (int i = 0; i < arr.length; i++) {
        System.out.println(arr[i]);
    }
}
/*
创建方法，返回值是数组类型
return返回数组的地址
*/
public static int[] getArray() {
    int[] arr = { 1, 3, 5, 7, 9 };
    //返回数组的地址，返回到调用者
    return arr;
}
```



基本类型和引用类型作为参数类型区别

基本类型：

```
public static void main(String[] args) {
    int a = 1;
    int b = 2;
    System.out.println(a);
    System.out.println(b);
```

```
    change(a, b);
    System.out.println(a);
    System.out.println(b);
}
public static void change(int a, int b) {
    a = a + b;
    b = b + a;
}
```

引用类型：

```
public static void main(String[] args) {
    int[] arr = {1,3,5};
    System.out.println(arr[0]);
    change(arr);
    System.out.println(arr[0]);
}
public static void change(int[] arr) {
    arr[0] = 200;
}
```

总结:

方法的参数为基本类型时,传递的是数据值. 方法的参数为引用类型时,传递的是地址值.