# Java数组的创建及使用

> 数组是具有相同数据类型的一组数据的集合，作为对象允许使用new关键字进行内存分配。

## 1. 一维数组的创建和使用

声明一维数组，有下列两种方式：

- 数组元素类型 数组名字[];
- 数组元素类型[] 数组名字；

数组元素类型据定了数组的数据类型，它可以是java中任意的数据类型，声明一维数组，如：int arr[]；String str[]; int arr[]是声明int型数组，数组中的每个元素都是int型数值，同理，str[]数组中的每个元素都是String类型。

*注：Java语言中声明数组时不能指定其长度（数组中元素的个数），这是因为数组是一种引用类型的变量，因此使用它定义一个变量时，仅仅表示定义了一个引用变量（也就是定一个了一个指针），这个引用变量还未指向任何有效的内存，所以定义数组时不能指定数组的长度。而且由于定义数组仅仅只是定一个引用变量，并未指向任何有效的内存空间，所以还没有内存空间来存储数组元素，因此这个数组也不能使用，只有在数组进行初始化后才可以使用。*

为数组分配内存空间的语法格式如下：

数组名字=new 数组元素的类型[数组元素的个数]；

例如，arr=new int[5];也可以在声明数组的同时为数组分配内存，例如：int arr[]=new int[5];

附:一旦使用new关键字为数组分配了内存空间，每个内存空间存储的内容就是数组元素的值，也就是数组元素就有了初始值，即使这个内存空间存储的内容是空，这个空也是一个值null。也就是说不可能只分配内容空间而不赋初始值，即使自己在创建数组对象（分配内容空间）时没有指定初始值，系统也会自动为其分配。

附：诸如基础数据类型的包装类，其默认的**初始化值均为null**，因为基础数据类型的包装类创建的数组属于引用数组（对象数组），**对象数组默认的初始化值都是null**。

理解了一维数组的声明和创建，二维数组的声明和创建大同小异。

二维数组的声明，int arr[][]；第一个下标表示行，第二个下标表示列。

为数组分配内存之后，就要学习如何初始化数组了。一维数组的初始化形式有两种：

int arr[]=new int[]{1,2,3,4};

或　int arr[]={1,2,3,4};

## 2. 二维数组的初始化方法

int myarr[][]={{12,2},{43,45}};

**注意，写成intmyarr[][]={12,2,43,45}是错误的。**

接下来，通过两段小的代码来解释如何使用一维数组和二维数组。

第一个例子是使用一维数组将1~12月各月的天数输出。代码如下：

```
public class shuZu{
 public static void main(String[] args){
   int day[]=new int[]{31,28,31,30,31,30,31,31,30,31,30,31};
   for (int i=0;i<12;i++){
     System.out.println((i+1)+"月有"+ day[i]+"天");
     }
   }
 }
```

第二个例子是使用二维数组将数组中的元素输出。代码如下：

```
public class shuZu{
public static void main(String args[]){
 int arr[][]=new int[][]{{1},{2,3},{4,5,6}};
 for(int i=0;i<arr.length;i++){
   for(int j=0;j<arr[i].length;j++){
     System.out.print(arr[i][j]);
     }
     System.out.println();
   }
 }
 }
```

简单介绍了一下数组的基本内容，接下来讲解一下经典的数组冒泡排序算法。

## 3. 数组冒泡排序

冒泡排序原理：相邻的两位数做比较，1和2比较 2和3比较 3和4比较 4和5比较，这样依次比较，如果前面的数小于后面的，不做操作，如果前面的数大于后面的数则调换两个数字的位置，列1>2 则数字顺序为2 1(注：此时的1 2 3 4 5代表索引而不是数组)，所以一层循环能挑出一个当前数组参与比较数字中的最大的数字，并将其排到数组的最末尾。
但是一层循环根本不够用，只能挑选出一个最大数，数组中其他的数字还没有正确排序。
这里遵循一个原理就是数组里面有length个数字，要进行length-1次循环。至于为什么要this.length-i，是因为第一次比较7个数字，第二个只要比较前6个就行了，第7个肯定是最大的了。这里要着重强调为什么会有2个for循环，为什么要用this.length-i。

下面用一个例子来做详细的说明：

```
/**
 * @description: 冒泡排序
 * @author: mqxu
 * @create: 2020-10-20 21:52
 **/
public class BubbleSort {
    /**
     * 冒泡排序的一种实现，没有任何优化
     *
     * @param a: 数组
     * @param n: 排序趟数
     */
    public static void bubbleSort1(int[] a, int n) {
        int i, j;
        //表示n次排序过程。
        for (i = 0; i < n; i++) {
            for (j = 1; j < n - i; j++) {
                //前面的数字大于后面的数字就交换
                if (a[j - 1] > a[j]) {
                    //交换a[j-1]和a[j]
                    int temp;
                    temp = a[j - 1];
                    a[j - 1] = a[j];
                    a[j] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] arr = {1, 1, 2, 0, 9, 3, 12, 7, 8, 3, 4, 65, 22};
```

```
            BubbleSort.bubbleSort1(arr, arr.length);
            for (int i : arr) {
                System.out.print(i + ",");
            }
        }
    }
```