

# 数组

2018年11月4日 11:12

## 1. 数组的定义：

- a. 一组数据的集合，数据类型；
- b. 特点：
  - i. 数据类型相同或者兼容
  - ii. 数据是有序的
  - iii. 下标 (0, 长度-1)

## 2. 数组的声明及初始化

- a. **数据类型[ ] 变量名**； 数据类型 变量名[ ] ；  
**int [ ] ages;**            **int ages [ ];**  
数据类型int表示数组里边的数据的数据类型

### b. 创建数组（申请内存）

**ages = new int [20];**

- i. new到堆里申请空间，int (4) \*20字节

#### ii. 整形：默认值：0

浮点型：默认值：0.0

布尔：默认值：false

字符型：默认值：\u0000

引用类型：默认值：null

#### iii. 访问数组元素（下标）

**ages [下标]**

#### iv. 数组的其他定义格式

**char [ ] chs = {'a','b','c','d'};**

没有new，按照{ }里的个数，在堆里申请内存，char (2) \*4个字节

**char [ ] ch = new char[4];**

**ch = {'a','b','c','d'}; //error**

★ **System.out.println(Arrays.toString(ch)); //把数组以字符串的形式输出**

### v. 案例1：

```
/**
 * 随机生成10个数，打印输出
 * 把10个数存到数组里面
 * 定义最大值变量max
 * 使用max和数组里每一个元素比较，用max存储最大值
 */
void test4() {
    int[] ch = new int[10];
    for(int i = 0; i < ch.length; i++) {
        ch[i] = (int)(Math.random()*100);
        System.out.print(ch[i] + "\t");
    }
    //打印
    System.out.println();
}
```

```

        System.out.println(Arrays.toString(ch)); //把数组以字符串的形式输出
        //最大值
        int max = ch[0];
        for(int j = 0; j < ch.length; j++) {
            if(ch[j] > max) {
                max = ch[j];
            }
        }
        System.out.println("最大值是: " + max);
    }
}

```

```

/*
 * 定义一个数组长度为20，元素均为0-9之间的任意数字
 * 统计每个数字出现的次数
 */
void test5() {
    int[] nums = new int[20];
    int[] count = new int[10];
    //生成数组
    for(int i = 0; i < nums.length; i++) {
        nums[i] = (int)(Math.random()*10);
    }
    System.out.println(Arrays.toString(nums));
    //统计个数方法1
    for(int j = 0; j < nums.length; j++) {
        for(int m = 0; m < count.length; m++) {
            if(nums[j] == m) {
                count[m]++;
            }
        }
    }
    //统计个数方法2(简单)
    for(int n = 0; n < nums.length; n++) {
        count[nums[n]]++;
    }
    //输出结果
    for(int k = 0; k < count.length; k++) {
        System.out.println(k + "出现了" + count[k] + "次");
    }
}
}

```

### 3. 数组的复制和扩容

#### a. 数组的复制 ArrayList能自动扩容.

//arraycopy方法是static修饰的，static修的成员（变量和方法）可以使用类名进行调用  
 //第一个参数：src表示原数组  
 //第二个参数：srcPos表示从原数组的哪个下标开始复制  
 //第三个参数：dest表示目标数组  
 //第四个参数：destPos表示目标数组的下标，从该下标开始放  
 //第五个参数：length表示从原数组里复制多少个到目标数组  
 System.arraycopy(src, srcPos, dest, destPos, length);  
 System.arraycopy(arr1, 0, arr2, 0, 4);

#### b. 数组扩容

//第一个参数：需要扩容的数组  
 //第二个参数：扩容后的长度  
 Arrays.copyOf(original, newLength);

```
void test7() {
    int[] array1 = {1,2,3,4};
    array1 = Arrays.copyOf(array1, array1.length+1);
    System.out.println(Arrays.toString(array1));
}
```

#### 4. 数组的排序

```
int [] = {43,24,6,3,7}
```

##### ★ a. 冒泡排序：两个相邻的数比较，大的数沉下去，小的数浮上来

```
void test1() {
    int[] num = {43,24,6,3,7};
    //冒泡排序
    for(int i = 0; i < num.length; i++) { // i=0 1 2 3
        for(int j = 0; j < num.length-1-i; j++) { // j=4 3 2 1 j<num.length-1-i
            if(num[j] > num[j+1]) {
                int temp = 0;
                temp = num[j];
                num[j] = num[j+1];
                num[j+1] = temp;
            }
        }
    }
    System.out.println(Arrays.toString(num));
}
```

##### ★ b. 选择排序：选择一个数和其余的所有数比较，找到最大（最小）值

```
void test2() {
    int[] num = {43,24,6,3,7};
    //选择排序
    for(int i = 0; i < num.length-1; i++) {
        for(int j = i+1; j < num.length; j++) {
            if(num[i] > num[j]) {
                int temp = 0;
                temp = num[i];
                num[i] = num[j];
                num[j] = temp;
            }
        }
    }
    System.out.println(Arrays.toString(num));
}
```

##### ★ c. 系统提供的排序方法

```
//系统提供的排序方法
void test3() {
    int[] array = {43,24,6,3,7};
    Arrays.sort(array);
    System.out.println(Arrays.toString(array));
}
```

#### 5. 二维数组

##### a. 二维数组的声明创建，及其使用

i. `int [][] array`; 几个[]就是几维数组    `int array [][]`    `int [] array []`

ii. `array = new int[行][列]`; 申请内存

iii. `array[0][1]`; //第一行，第二列

##### b. 案例

	大学物理	计算机原理	数据结构
0001	100	99	100
0002	90	60	70

//二维数组

```
void test4() {
    int[][] array = new int[2][3];
    array[0][1]=100;
    ★ for(int i =0;i<array.length;i++) {
        ★ for(int j =0;j<array[i].length;j++) {
            System.out.print(array[i][j]+"\\t");
        }
        System.out.println();
    }
}
```

//二维数组对角线值的和

```
void test5() {
    int [][] array = {{1,2,3,4},{5,6,7,8},{1,2,3,4},{5,6,7,8}};
    int sum = 0;
    for(int i =0;i<array.length;i++) {
        for(int j =0;j<array[i].length;j++) {
            System.out.print(array[i][j]+"\\t");
            if(i==j || i+j==3) {
                sum+=array[i][j];
            }
        }
        System.out.println();
    }
    System.out.println(sum);
}
```