

Java 数组

一、什么是Java中的数组？

二、数组有哪些常见的操作？

三、数组的五种赋值方法和使用方法

声明数组

声明数组并且分配空间

声明数组同时赋值(1)

声明数组同时赋值(2)

从控制台输入向数组赋值

四、求总和平均

五、求数组中最大值最小值

六、对数组进行升序排序

六、不调用方法进行数组排序

七、Arrays 类常用方法

一、什么是Java中的数组？

数组是一种数据结构，可以存储多个相同类型的元素；在Java中，数组是一种对象，它可以存储基本数据类型和对象类型。

二、数组有哪些常见的操作？

- 创建数组：使用new操作符创建一个新的数组对象。
- 访问元素：使用索引访问数组中的元素。
- 遍历数组：使用循环结构遍历数组中的所有元素。
- 复制数组：使用System.arraycopy()方法或Arrays.copyOf()方法将一个数组复制到另一个数组中。
- 排序数组：使用Arrays.sort()方法对数组进行排序。
- 搜索数组：使用Arrays.binarySearch()方法在一个已排序的数组中搜索一个元素。

三、数组的五种赋值方法和使用方法

声明数组



Java

复制代码

```
1  int[] arr;  
2  //开辟三个空间  
3  arr = new int[3];  
4  //向数组中赋值, 0、1、2是下标  
5  arr[0] = 30;  
6  arr[1] = 97;  
7  arr[2] = 65;
```

声明数组并且分配空间



Java

复制代码

```
1  int[] arr = new int[3];  
2  //开辟三个空间  
3  //向数组中赋值  
4  arr[0] = 30;  
5  arr[1] = 97;  
6  arr[2] = 65;
```

声明数组同时赋值(1)



Java

复制代码

```
1  int[] arr = new int[]{30, 97, 65};
```

声明数组同时赋值(2)



Java

复制代码

```
1  int[] arr = {30, 97, 65};
```

从控制台输入向数组赋值

```
1  int[] arr = new int[3];
2  System.out.println("向数组中存入三个数据: ");

3  //arr.length是数组的长度
4  for (int i = 0; i < arr.length; i++) {
5      //i是下标
6      arr[i] = input.nextInt();
7  }
8  //把数组中的值输出
9  for (int i = 0; i < arr.length; i++) {

10     //输出数组中的值
11     System.out.println(arr[i]);
12 }
```

四、求总和平均

```
1  int i = 0; int zong = 0;
2  int[] arr = new int[3];
3  for (i = 0; i < arr.length; i++) {
4      System.out.print("输入第" + (i + 1) + "位同学的成绩: ");
5      arr[i] = input.nextInt();
6      zong = zong + arr[i];
7  }
8  System.out.println(i + "位同学的总分是: " + zong + "平均分是: " + zong * 1.0 /
    arr.length);
```

五、求数组中最大值最小值

```
1  int[] arr = { 90, 100, 98, 67, 45, 79 };
2  int max = arr[0];
3  int min = arr[0];
4  for (int i = 1; i < arr.length; i++) {
5      if (arr[i] > max) {
6          max = arr[i];
7      }
8      if (arr[i] < min) {
9          min = arr[i];
10     }
11 }
12 System.out.println("最大值为: " + max); System.out.println("最小值为: " + min);
```

六、对数组进行升序排序

```
1  数组升序输出
2  int[] arr1 = { 90, 100, 98, 67, 45, 79 };
3      Arrays.sort(arr1);
4  for (int i = 0; i < arr1.length; i++) {
5      System.out.println(arr1[i]);
6  }
7
8  数组降序输出
9  int[] arr2 = {90, 100, 98, 67, 45, 79};
10 Arrays.sort(arr2);
11 for (int i = arr2.length - 1; i >= 0; i--) {
12     System.out.println(arr2[i]);
13 }
```

六、不调用方法进行数组排序

```
1  int[] array = {10, 30, 17, 6, 29, 20};
2  int temp; for(int i = 0; i < array.length - 1; i++){
3      for (int j = 0; j < array.length - 1 - i; j++) {
4          if (array[j] < array[j + 1]) {
5              //<为降序, >为升序
6              temp = array[j];
7              array[j] = array[j + 1];
8              array[j + 1] = temp;
9          }
10     }
11 }
12 for(int i : array){
13     System.out.print(i + "\t");
14 }
```

七、Arrays 类常用方法

Arrays类位于 java.util 包中， import java.util.Arrays;

```
1  int[] array1 = {10, 20, 16, 19, 5};
2  int[] array2 = {10, 20, 16, 19, 5};
3  //boolean equals(array1, array2)比较两个数组是否相等
4  System.out.println(Arrays.equals(array1, array2));
5  //打印true
6
7  //String toString(array)将数组转换成字符串显示值
8  System.out.println(Arrays.toString(array1));
9  //打印[10, 20, 16, 19, 5]
10
11 //void fill(array, val)将数组的所有值填充为val Arrays.fill(array2, 20); System.out.println(Arrays.toString(array2));
12 //打印[20, 20, 20, 20, 20]
13
14 //数组copyOf(array, newLength)将array复制成指定长度的新数组 int[] array3 = Arrays.copyOf(array1, 10);
15 System.out.println(Arrays.toString(array3));
16 //打印[10, 20, 16, 19, 5, 0, 0, 0, 0, 0]
17
18 //int binarySearch(array, key)在array数组中查找指定的值，并返回其索引，前提是数组是有序的 Arrays.sort(array1);
19 System.out.println(Arrays.binarySearch(array1, 16));
20 //打印2
```