

本文内容是对数组知识的专题梳理和总结，本文内容包括：

数组-相关的概念

数组-内存结构

数组-元素默认值

数组-下标越界

数组-去重问题

数组-扩容问题

数组-排序问题

数组-工具类Arrays

1 数组-相关的概念

什么是数组？

- 1 答：有三点
- 2 1. 存储多个数据
- 3 2. 这些数据的类型相同
- 4 3. 这些数据在内存中是连续存储的
- 5 对数组的理解：定义数组相当于一次定义多个变量，数组的每一个元素就是一个变量

什么是数组名称？

- 1 答：指向到数组元素的变量就是数组名称，例如 `int arr[] = {1,2,3}`, `arr` 就是数组名称

什么是数组元素？

- 1 答：数组中的每一个值就是一个元素

什么是数组下标（索引）？

1 | 答：下标是数组元素在数组中的序号，从0开始，依次加1

什么是数组下标越界？

1 | 答：不在有效范围内的下标叫做下标越界。有效范围 ≥ 0 并且 $< \text{length}$

什么是数组大小（长度）？

1 | 答：数组中元素的个数是数组的长度，也叫做数组大小，`length`属性获取数据大小

使用数组的步骤是什么？

1 | 答：四步
2 | 1.定义数组名称 `int arr[];`
3 | 2.给数组元素分配内存 `arr = new int[3];`
4 | 3.数组元素初始化 `arr[0]=60;`
5 | 4.使用数组，如排序，求最大值等

数组的定义？

- 对于元素为 1, 2, 3 的整型数组 `arr`，你知道有哪些种定义数组的方式？

```
1 | 方式1:
2 | int arr[];
3 | arr = new int[3];
4 | arr[0] = 1;
5 | arr[1] = 2;
6 | arr[2] = 3;
7 |
8 | 方式2:
9 | int arr[] = new int[3];
10 | arr[0] = 1;
11 | arr[1] = 2;
12 | arr[2] = 3;
13 |
14 | 方式3:
15 | int arr[] = new int[]{1,2,3};
16 |
```

```
17 | 方式4:  
18 | int arr[] = {1,2,3};
```

什么是二维数组？

1 | 答：数组元素的值还是一个数组，也就是一维数组的一维数组

什么是不规则数组？

```
1 | 答：不规则数组是每一行的列数不一样的二维数组。  
2 | 例如：  
3 | int arr[][] = {  
4 |     {1,2,3}, //第一行3个数据  
5 |     {1,2}    //第二行2个数据  
6 | }
```

2 数组-内存结构

一：堆和栈

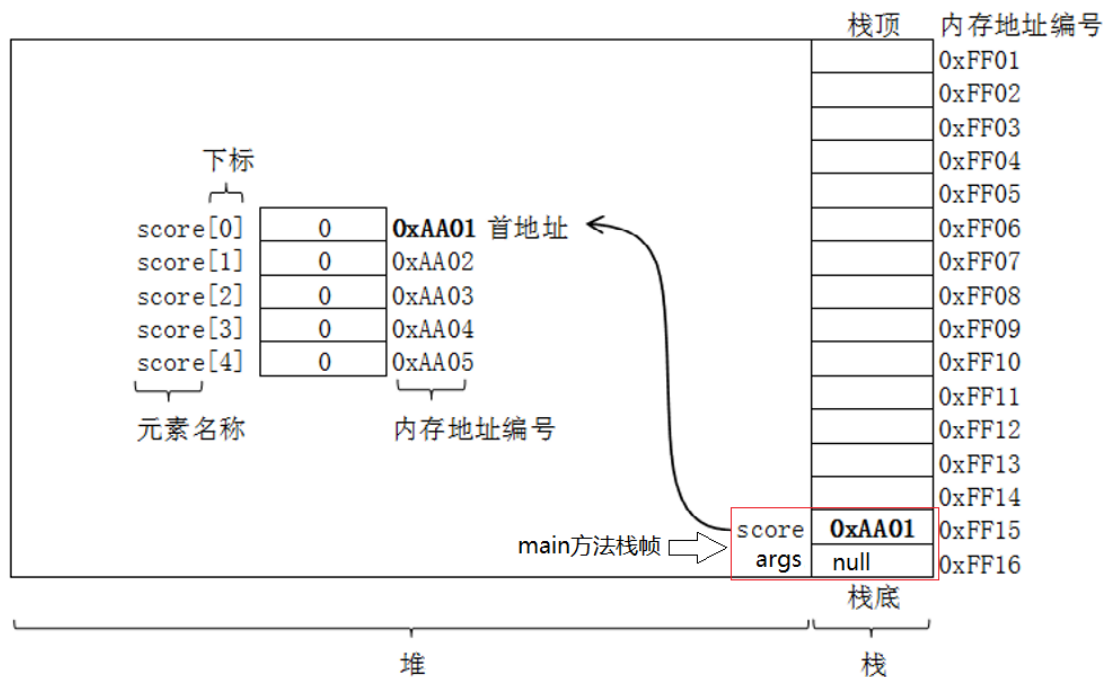
1. JVM内存分成几个区域，其中有两个区域分别叫做堆和栈
2. 数组元素分配在堆中，数组名称分配在栈中

二：数组内存结构

下面的代码创建了5个元素的整型数组

```
1 | public static void main(String[] args) {  
2 |     int []score = new int[5];  
3 | }
```

这个数组的内存结构如下图



3 数组-元素默认值

一：数组元素不赋值时，元素有值吗？

当数组创建后，如果没有给元素赋值，会使用默认值，不同类型的数组默认值不一样。

- byte,short,int,long类型，默认值都是0
- float,double类型，默认值是0.0
- boolean类型，默认值为false
- char类型，默认值为'\u0000'，表示空白字符
- 其他类型，默认值为null

二：测试一下

1. 执行完以下代码int []x = new int[25];后，以下哪项说明是正确的（）

- A. x[24]为0
- B. x[24]未定义
- C. x[25]为0
- D. x[0]为null

2. 执行完以下代码后，输出结果是什么（）

```
1 double arr1[] = new double[3];
2 Double arr2[] = new Double[3];
3 System.out.println(arr1[0]);
4 System.out.println(arr2[0]);
```

三：double和Double

- double和Double是不一样的，double是基本数据类型，默认值是0.0，Double是引用数据类型，默认值是null
- Double的double的包装类，两者之间可以通过装箱和拆箱进行转换，后续会学到，先了解一下即可

4 数组-下标越界

一：下标越界时会报出什么异常？

- 会报出 `ArrayIndexOutOfBoundsException`

二：如何查看下标越界异常？

在IDEA中，下标越界异常信息中有两个重要信息，如下图所示

- 异常原因，这个是英文的，你要能看懂
- 异常位置，图中显示Test.java类的第13行发生了异常，鼠标点击链接，光标自动定位到异常代码处

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException Create breakpoint: Index 6 out of bounds for length 6
at com.kaifamiao.oop.Test.distinct([Test.java:13](#))
at com.kaifamiao.oop.Test.main([Test.java:7](#))
Process finished with exit code 1

点击这里定位异常

异常原因

三：如何防止下标越界？

- 让下标的值 ≥ 0 并且 $< \text{length}$

5 数组-去重问题

一：去重问题

去重：就是去除数组中重复的值

去重的核心问题是算法问题，去重的方法有很多种，就看那种算法更好

二：双循环去重

去重的常规思路是使用双循环去重，下面去重代码的重算法是

1. 使用变量len记住元素总数
2. 从第一个数开始，找第一个数后面所有的数是否与第一个数重复
3. 如果发现重复的数，就将数组最后一个数移动到重复数的位置，并将len减去1
4. 然后从第二个数开始，重复第一个数的过程

```
1 private static void distinct() {
2     // 原始数组
3     int[] array1 = {1, 2, 3, 4 ,1, 2};
4
5     /*
6         1. 计算不重复数据的个数，存储在len中
7         2. 当有重复数据时，用数组最后面的数据覆盖重复的数据
8     */
9     int len = array1.length;
10    for (int i = 0; i < len; i++) {
11        for (int j = i + 1; j < len; j++) {
12            if (array1[i] == array1[j]) {
13                array1[j] = array1[len - 1];
14                len--;
15                j--;
16            }
17        }
18    }
19    // 去重后的元素放到新数组newArr中
20    int[] newArr = new int[len];
21    for (int i = 0; i < len; i++) {
22        newArr[i] = array1[i];
23    }
24    for (int i = 0; i < newArr.length; i++) {
25        System.out.print(newArr[i] + " ");
26    }
27 }
```

三：更牛的去重

- JDK8的新特性stream操作，能更方便的去重，后面会学到，提前见识一下吧

```
1 public static void distinct() {
2     int []array1 = new int[]{1, 2, 3, 4 ,1, 2};
3     //distinct()去重
4     int []array2 = Arrays.stream(array1).distinct().toArray();
5     System.out.println(Arrays.toString(array2));
6 }
```

6 数组-扩容问题

一：数组大小可以变化吗？

- 要知道数组大小是不可变的
- 如果要改变数组大小，只能重新定义数组

二：手动扩容

- 手动扩容：自己编写代码，重新定义数组实现扩容

```
1 public static void extendCapacity() {
2     // 需求：给数组 int []array1 =new int[]{34,45,56}中再增加一个
    元素67。
3
4     // 原始数组，长度为3
5     int array1 [] = {34,45,56};
6
7     // 扩容数组，长度为4
8     int array1New[] = new int[array1.length + 1];
9
10    //将原数组元素赋值到新数组中
11    for (int i = 0; i < array1.length; i++) {
12        array1New[i] = array1[i];
13    }
14
15    //向新数组中添加新数据
16    for (int i = array1.length; i < array1New.length; i++) {
```

```

17         array1New[i] = 67;
18     }
19
20     //遍历新数组
21     for (int i = 0; i < array1New.length; i++) {
22         System.out.println(array1New[i]);
23     }
24 }

```

三：工具类扩容

- 工具类扩容：Arrays工具类的 `copyOf(原始数组, 新数组长度)` 方法实现扩容

```

1     public static void extendCapacity() {
2         // 原始数组，长度为3
3         int score[] = {34,45,56};
4
5         // 扩容数组，长度为4
6         int scoreNew[] = Arrays.copyOf(score, score.length +
7 1);
8
9         // 向新数组中添加新数据
10        scoreNew[scoreNew.length-1] = 67;
11
12        // 输出新数组
13        System.out.println(Arrays.toString(scoreNew));
14    }

```

7 数组-排序问题

一：冒泡排序

- 核心算法
 - 相邻的比较
 - 大数往后移动
- 核心代码


```
for (int i = 0; i < len - 1; i++) {  
    for (int j = 0; j < len - 1 - i; j++) {  
        if (array[j] > array[j + 1]) {  
            int temp = array[j];  
            array[j] = array[j + 1];  
            array[j + 1] = temp;  
        }  
    }  
}
```

二：工具类排序

```
1 Arrays.sort(arr)
```

8 数组-工具类Arrays

一：什么是工具类？

- 大多数开发人员都有可能用到的相同的功能，比如数组排序，是不是每个开发人员都要写一遍代码呢？
- 对于开发人员共性的需求，JDK已经提供相应的功能，我们把JDK提供的共性需求功能的类称为工具类
- JDK中的工具类有很多，我们会陆续学习

二：Arrays工具类

- java.util.Arrays就是JDK提供的专门用于操作数组的工具类
- Arrays.sort() 排序
- Arrays.toString() 显示数组元素
- Arrays.binarySearch() 在数组中查找
- Arrays.copyOf() 复制并创建新数组
- Arrays的其他功能参见JDK文档

