

内容

- 一维数组的声明和初始化
- 数组元素的引用
- 数组的遍历
- 数组的常用方法及算法
- 二维数组的声明和使用
- 数组拷贝

一、 数组的概述

数组可以看成是**多个相同类型**数据的组合，对这些数据的统一管理。

所谓数组(**Array**)，就是相同数据类型的元素按一定顺序排列的集合，就是把有限个类型相同的变量用一个名字命名，然后用编号区分他们的变量的集合，这个名字称为 数组名，编号称为 **下标**。组成数组的各个变量称为数组的**分量**，也称为数组的 **元素**，有时也称为下标变量。数组是在程序设计中，为了处理方便，把具有相同类型的若干变量按有序的形式组织起来的一种形式。这些按序排列的同类数据元素的集合称为数组“**[]**”。

数组变量属于**引用类型**，数组也可看成是**对象**，数组中的每个元素相当于该对象的成员变量

数组中的元素可以是任何数据类型，包括基本类型和引用类型。

数组是相同数据类型的有序集合。数组描述的是相同类型的若干个数据,按照一定的先后次序排列组合而成。其中,每一个数据称作一个数组元素,每个数组元素可以通过一个下标来访问它们。

数组具有以下**特点**：

1、 定长

长度一旦确定不可以改变。

2、相同数据类型

其元素必须是 相同 类型,不允许出现混合类型。数组中的元素可以是任何数据类型，包括基本类型和引用类型。

3、位置有序

元素所在的位置是有序的。

4、数组本身属于引用类型

数组变量属引用类型，数组也可以看成是对象，数组中的每个元素相当于该对象的成员变量。数组本身就是对象，Java 中对象是在堆中的，因此数组无论保存基本类型还是其他对象类型，数组对象本身是在堆中的。

二、 一维数组的声明方法

`type var[];` 或 `type[] var;` 推荐使用后一种

例如：

```
int a1[];
```

```
int[] a2;
```

```
Person[] p1;
```

Java 语言中声明数组时不能指定其长度（数组中元素的个数），例如：

```
int a[5]; // 非法
```

三、 数组对象的创建

Java 中使用关键字 new 创建数组对象：

```
数组名 = new 数组元素的类型[数组元素的个数]
```

```
int[] s;
```

```
s = new int[5];
```

注意：元素为引用数据类型的数组中的每一个元素都需要实例化

```
Person[] people;
```

```
people = new Person[3];
```

```
people[0] = new Person();
```

```
people[1] = new Person();
```

```
people[2] = new Person();
```

四、 数组的初始化

1. 动态初始化

数组定义与为数组元素分配空间和赋值的操作分开进行

```
int[] a;
```

```
a = new int[3];
```

```
a[0] = 1; a[1] = 2; a[2] = 3;
```

2. 静态初始化

在定义数组的同时就为数组元素分配空间并赋值

```
int[] a = {1,2,3};
```

数组元素的默认初始化

数组是引用类型，它的元素相当于类的成员变量，因此数组分配空间后，每个元素也被按照成员变量的规则被隐式初始化

```
int[] a = new int[5];
```

五、 数组元素的引用

定义并用运算符 new 为之分配空间之后，才可以引用数组中的每个元素，数组元素的引用方式为：

```
arrayName[index];
```

index 为数组元素下标，可以是整型常量或整型表达式。如：

```
a[3], b[i], c[5*i]
```

数组元素下标从 0 开始；长度为 n 的数组的合法下标取值范围为

0~n-1

每个数组都有一个属性 length 指明它的长度

a.length 的值为数组 a 的长度（**元素的个数**）

六、 一维数组的遍历

1、普通 for

```
for(int 索引=0 ; 索引<长度; 索引++){  
    数组名称[索引] = 值 ;  
}
```

2、增强 for

```
for(元素类型 局部变量 : 数组){  
    局部变量  
}
```

七、数组的界限

1、必须开辟空间才能使用否则出现 `java.lang.NullPointerException`

2、索引的范围 `[0,length)` 否则出现

`java.lang.ArrayIndexOutOfBoundsException`

3、长度:`[0,∞)` **理想状态**

0 表示空数组,确定了存放的数据类型,该数组不能直接使用,如果使用

`java.lang.ArrayIndexOutOfBoundsException`

-1:编译通过,运行错误,如果使用

`java.lang.NegativeArraySizeException`

数组的排序及指定元素的查找

数组内容的比较可以使用 `equals()`方法吗?

`Arrays.equals()`:比较数组

`Arrays.sort()`: 排序

`Arrays.binarySearch(数组, 元素)` : 二分法查找

数组 `copyOf(源数组, 长度)`

使用 `java.lang.System` 类的静态方法

<code>static void</code>	<code>arraycopy(Object src, int srcPos, Object dest, int destPos, int length)</code> 从指定源数组中复制一个数组,复制从指定的位置开始,到目标数组的指定位置结束。
--------------------------	--

src - 源数组

srcPos - 源数组中的起始位置

dest - 目标数组

destPos - 目标数据中的起始位置

length - 要复制的数组元素的数量

如果源数据数目超过目标数组边界则会抛出 `IndexOutOfBoundsException` 异常

八、 二维数组

二维数组可以看成以**数组为元素的数组**

```
int[][] a = {{1},{2,3},{4,5,6}};
```

Java 中多维数组的声明和初始化应按**从高维到低维**的顺序进行

```
int[][] a = new int[3][];
```

```
a[0] = new int[2];
```

```
a[1] = new int[1];
```

```
a[2] = new int[4];
```

```
int[][] a1 = new int[][4]; //非法
```

九、 二维数组初始化

静态初始化

```
int[][] intA = {{0},{3},{2,1}};
```

```
int[3][2] intB = {{1,2},{3,4},{5,6}}; // 非法
```

动态初始化

```
int[][] a = new int[3][5];
```

```
int[][] b = new int[2][];
```

```
b[0] = new int[5];
```

```
b[1] = new int[2];
```

二维数组的遍历

十、 总结

数组的内存布局

常见算法

思考作业

- 1.数组的类型可以为任意类型吗?
- 2.数组中的元素的类型必须是相同的吗? 数组中的元素是有序的吗?
- 3.数组的长度可变不? 数组的长度使用哪个属性?
- 4.数组元素下标(或索引)的范围是?
- 5.数组也是引用类型吗? 数组也是对象, 这句话对吗?
- 6.数组中的元素就像对象中的成员变量一样, 初始化方式也一样。这句话对吗?
- 7.编写程序, 练习使用数组类型对象的 length 属性, 测试并体会数组元素的默认初始化机制.
- 8.数组的三种初始化方式是什么?
- 9.编写一应用程序练习数组对象的两种初始化方式, 并输出各元素的值。
10. 数组的静态初始化代码, 下面代码有没有不合理的地方:

```
Dog[] dogs2 = {new Dog("大黄",1), new Dog("小强",2)};
```

- 11.阅读如下代码, 该代码 的作用是:

```
char[] s;  
s = new char[26];  
for ( int i=0; i<26; i++ ) {  
    s[i] = (char) ('A' + i);  
    System.out.println(s[i]);  
    // System.out.println("s[" + i + "]=" + s[i]);  
}
```

12. 下面的数组定义哪些是正确的?

```
A: int a[] = new int[3];
```

B: `int a[3]= new int[];`

C: `int[] a= new int[3]{1,2,3};`

D: `int[] a = new int[] {1,2,3,4};`

上机操作

- 1、练习课堂上的所有代码
- 2、完成 10-20 的平方，并将结果保存在一个数组中
- 3、定义一个长度为 10 的一维字符串数组，在每一个元素存放一个单词；然后运行时从命令行输入一个单词，程序判断数组是否包含有这个单词，包含这个单词就打印出 "Yes"，不包含就打印出 "No"。
- 4、将 1-20 存放到对应的数组中，求和
- 5、编写方法实现 将数组转成字符串，效果为:[10,10,10,7,4,7,9,5,0,8]
- 6、.数组扩容与元素添加:存在数组 { "a" , "b" , "c" } ,将 "X" 插入其中，变成 { "a" , " b" , " x" , " c" }
- 7、`int[] a={2,9,13,28,30,45,58,60}`，随机输入一个数插入到数组中，让数组仍然有序

寄语

人生最大的错误是不断担心会犯错



谢谢大家