

Java 数组

Java语言中提供的数组是用来存储**固定大小**的**同类型**元素。也是对象

- 其长度是确定的
- 其元素必须是相同类型
- 可以存储**基本数据类型**和引用数据类型
- 数组变量属于引用类型

直接打印数组是地址。

声明、创建和初始化

声明数组变量

首先必须声明数组变量，才能在程序中使用数组。下面是声明数组变量的语法：

```
dataType[] arrayRefVar;    // 首选的方法
```

或

```
dataType arrayRefVar[];    // 效果相同，但不是首选方法
```

创建数组

Java语言使用new操作符来创建数组，语法如下：

```
arrayRefVar = new dataType[arraySize];
```

上面的语法语句做了两件事：

- 一、使用dataType[arraySize]创建了一个数组。
- 二、把新创建的数组的引用赋值给变量 arrayRefVar。

数组变量的**声明**，和**创建**数组可以用一条语句完成，如下所示：

```
dataType[] arrayRefVar = new dataType[arraySize];
```

另外，你还可以使用如下的方式创建数组。

```
dataType[] arrayRefVar = {value0, value1, ..., valuek};
```

数组的元素是通过**索引**访问的。

实例

下面的语句**首先声明**了一个**数组变量**myList，接着**创建**了一个包含10个double类型元素的数组，并且把它的**引用赋值**给myList变量。

```
double[] myList = new double[10];
```

处理数组

数组的元素**类型**和数组的**大小**都是**确定的**，所以当处理数组元素时候，我们通常使用**基本循环**或者**foreach循环**。

示例

该实例完整地展示了如何创建、初始化和操纵数组：

```
public class TestArray {

    public static void main(String[] args) {
        double[] myList = {1.9, 2.9, 3.4, 3.5};

        // 打印所有数组元素
        for (int i = 0; i < myList.length; i++) {
            System.out.println(myList[i] + " ");
        }
        // 计算所有元素的总和
        double total = 0;
        for (int i = 0; i < myList.length; i++) {
            total += myList[i];
        }
        System.out.println("Total is " + total);
        // 查找最大元素
        double max = myList[0];
        for (int i = 1; i < myList.length; i++) {
            if (myList[i] > max) max = myList[i];
        }
        System.out.println("Max is " + max);
    }
}
```

以上实例编译运行结果如下：

```
1.9
2.9
```

```
3.4
3.5
Total is 11.7
Max is 3.5
```

foreach循环

JDK 1.5 引进了一种新的循环类型，被称为foreach循环或者加强型循环，它能在不使用下标的情况下遍历数组。

示例

该实例用来显示数组myList中的所有元素：

```
public class TestArray {

    public static void main(String[] args) {
        double[] myList = {1.9, 2.9, 3.4, 3.5};

        // 打印所有数组元素
        for (double element: myList) {
            System.out.println(element);
        }
    }
}
```

以上实例编译运行结果如下：

```
1.9
2.9
3.4
3.5
```

数组作为函数的参数

数组可以作为参数传递给方法。例如，下面的例子就是一个打印int数组中元素的方法。

```
public static void printArray(int[] array) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
}
```

下面例子调用printArray方法打印出 3 , 1 , 2 , 6 , 4和2：

```
printArray(new int[]{3, 1, 2, 6, 4, 2});
```

数组作为函数的返回值

```
public static int[] reverse(int[] list) {  
    int[] result = new int[list.length];  
  
    for (int i = 0, j = result.length - 1; i < list.length; i++, j--)  
    {  
        result[j] = list[i];  
    }  
    return result;  
}
```

以上实例中result数组作为函数的返回值。

获取数组长度：

```
a.length ;
```

数组的拷贝：

System类里：

```
static void arraycopy(object src, int srcpos, object dest, int  
destpos, int length)
```

该方法可以将src数组里的元素值赋给dest数组的元素，

srcpos指定从src数组的第几个元素开始赋值，

length参数指定将src数组的多少个元素赋给dest数组的元素