Chapter09 数组

本章内容:

一、 为什么需要数组?
1. 简化繁琐的变量设计
2. 利于数据处理

二、 什么是数组?
1. 数组：是一个存储相同数据类型的一组数据的空间。
2. 数组基本要素
   1) 标识符
   2) 数组元素
   3) 元素下标
   4) 元素类型

三、 使用数组(重点)

整型数组，保存 4，5，6，7 这个四个数字?

1，声明数组: 数组的类型 数组名;（定义变量 int a;）

```
int[] arr;
```

2，定义数组：数组名=new 数据类型[长度];

```
arr = new int[4];
```

3，逐个赋值：数组名[下标]=值;（入住 a =100）

下标：数组中数据每个空间的编号。下标一定是从 0 开始。

```
arr[0] = 4;
arr[1] = 5;
arr[2] = 6;
arr[3] = 7;
```

4，(使用 System.out.println(a);)

```
for(int i=0;i<4;i++){
    System.out.println(arr[i]);

}
```

等价于：

```
System.out.println(arr[0]);

System.out.println(arr[1]);

System.out.println(arr[2]);

System.out.println(arr[3]);
```

数据类型 数组类型

int	int[]
double	double[]
char	char[]
String	String[]
boolean	boolean[]

四、 数组特性(重点)

1. 数组的特点

第 1 点：数组本身就是一个变量。

第 2 点：数组使用来保存大批量同类型数据的一个空间。

```
int a = 100;
String b = “a”;
int c = 33;
```

====➔上面的三组数据不能使用数组存。

第 3 点：数组的的每个空间在内存中一定是连续的。

第 4 点：数组的长度一旦确定，就不能修改了。

```
int[] arr = new int[4];
```

如果确定大小后，空间开小了，只能重新开数组。

```
arr = new int[20];
```

2. 数组有默认值的，变量没有默认值！

1) int[] arr ;
arr = new int [4];默认在数组中每个下标对应的空间放入 0
System.out.println(arr[0]);输出 0
数组 默认值
int[] ------ 0
double[]----0.0
String[]----null
char[]------ascII 码为 0 的字符
boolean[] --false

2) int a;
System.out.println(a);报错

3. 三合一

```
int[] arr;
arr = new int[4];
```

等价于：int[] arr = new int[4];

```
int[] arr;
arr = new int[4];
arr[0]=4;
arr[1]=5;
arr[2]=6;
arr[3]=7;
```

等价于：int[] arr = new int[]{4,5,6,7};
还可以等价于：int[] arr= {4,5,6,7};

4. 数组的下标与长度
   1) 数组的长度=数组名.length
      int[] arr = {4,5,6,7,8};
      arr.length=5;
   2) 数组的最大下标=数组的长度-1
      =数组名.length-1

数组下标不在 0～最大下标内就会出现一下错误：

ArrayIndexOutOfBoundsException 数组下标越界的异常

array 数组 index 索引 下标 outofbounds 越界 exception 异常

数据典型应用(重点、难点)

5. 求平均分，最大值
6. 排序：
   1) Arrays.sort(数组名).
      1. **注意地方：**必须要导入 java.util.Arrays 类
         import java.util.Arrays;
   2) 冒泡排序
      升序：11 22 33 44 55 66　　特点：相邻的 2 个数：前<后
      降序：66 55 44 33 22 11　　特点：相邻的 2 个数：前>后
   3) 选择排序
      从数组的第 1 开始，让其与它后面所有的数都比较一次。
   4) 插入排序
      升序
      原数组：8，4，2，1，23，344，12
      第 1 轮
      下标：i=1
      arr :8，|4，2，1，23，344，12
      如果 4<8. arr[i]<arr[i-1]交换
      结果：4，8，2，1，23，344，12

i—
伪代码
```
int i=1;
while(i>0){
    如果 arr[i]<arr[i-1] 交换
    否则 停止插入
    i--;
}
```

第 2 轮
下标：i=2
4，8，|2，1，23，344，12
如果 2<8 arr[i]<arr[i-1]，交换
4，2，|8，1，23，344，12
i—
否则 停止插入；
下标 i=1
如果 2<4.arr[i]<arr[i-1]交换
结果：2,4,8, 1，23，344，12
伪代码
```
int i=2;
while(i>0){
    如果 arr[i]<arr[i-1] 交换
    否则 停止插入
    i--;
}
```

第 3 轮
i=3
2,4,8,| 1，23，344，12
如果 1<8 交换
结果：2,4,1,| 8 23 344 12
i—
否则 停止插入
如果 1<4 交换
结果：2，1，4，|8，23 344 12
i—
否则 停止插入
如果 1<2 交换
结果：1，2，4，8，23 ，344 ，12
i--
否则 停止插入
伪代码：

```
int k ;
for(int i=1;i<=数组长度-1;i++{//i 表示轮数
    k=I;
    while(i>0){
        if(arr[i]<arr[i-1]){
            交换;
            k--;
        }else{
            停止插入
        }
    }

}
```

7. 求最值
   擂台法：
8. 插入算法
   升序数组：0 60 63 82 85 99
   插入的数：70

```
int i=0;
if(70<arr[0]){
    找到了；
    停止查找
    确定插入的位置为 i-1；
}
int i=1;
if(70<arr[1]){
    找到了；
    停止查找
    确定插入的位置为 i-1；
}

Index 表示插入的位置
Int index = arr.length-1;
for(int i=1;i<arr.length;i++){
    if(70<arr[i]){
        找到了插入的位置为：i-1；
        index=i-1；
        停止查找
    }
}
60 前移一位
```

63 前移一位
70 放入数组的下标为 2 的位置

9. 普通查找（标记）与二分查找
1,标记法 （不是最好的算法）
2，二分查找算法（要求：数组是有序的）
原数组： 60 63 82 85 99
查找的数 82
第 1 轮：

```
int i=0;
int j=arr.length-1;
int middle = (i+j)/2;
if(number>arr[middle]){
    i= middle+1;
}else if(number<arr[middle]){
    j= middle-1;
}else{
    这个数就在 arr[middle]
    停止查找
}
```

第 2 轮

```
Middle =(i+j)/2;
if(number>arr[middle]){
    i= middle+1;
}else if(number<arr[middle]){
    y= middle-1;
}else{
    这个数就在 arr[middle]
    停止查找
}
```

结束的条件：开始下标<=结束下标 ←=> i<=j

```
int i=0;
int j=arr.lenght-1;
while(i<=j){
    Middle =(i+j)/2;
    if(number>arr[middle]){
        i= middle+1;
    }else if(number<arr[middle]){
        y= middle-1;
    }else{
        这个数就在 arr[middle]
```

```
            停止查找
        }

    }
```

10. 反序
    1) 交换法
    原数组： 60 63 82 85 99 88
    反序后的数组：99 85 82 63 60
    分析：
        arr[0]与 arr[arr.length-1]交换
        arr[1]与 arr[arr.legnth-2]交换
        ….
        arr[?]与 arr[arr.length-?]交换
        ？=(arr.length)/2-1=2
11. 复制算法
    1) 全部复制
    Arr={1,2,3,4,5,7}将这个数组复制到另外一个数组中。
    2）部分复制

    Arr={1,2,3,4,5,7}将这个数组中所有奇数复制到另外一个数组中。

五、 本章单词

1. array-数组;阵列
2. new- 新的、新建立的
3. length-长度
4. index-索引;数组下标
5. out of bounds-越限的、越界的