答疑：

1. 开一个新的project，取名叫Arraytest。开一个主类，取名ArrayTest。
2. 循环重新讲
3. 声明变量和类型转换

数组：

所谓数组，是无序的元素序列。若将有限个类型相同的变量的集合命名，那么这个名称为数组名。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。用于区分数组的各个元素的数字编号称为下标。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元素按无序的形式组织起来的一种形式。这些无序排列的同类数据元素的集合称为数组。

一个数组中的所有元素具有相同的数据类型（在C、C++、Java、pascal中都这样。但也并非所有涉及数组的地方都这样，比如在Visual Foxpro中的数组就并没这样的要求）。当然，当数据类型为 Variant 时，各个元素能够包含不同种类的数据（对象、字符串、数值等等）。可以声明任何基本数据类型的数组，包括用户自定义类型和对象变量。

### 特点

1.数组是相同数据类型的元素的集合

2.数组中的各元素的存储是有先后顺序的，它们在内存中按照这个先后顺序连续存放在一起

3.数组元素用整个数组的名字和它自己在数组中的顺序位置来表示。例如，a[0]表示名字为a的数组中的第一个元素，a[1]代表数组a的第二个元素，以此类推。

对于VB的数组，表示数组元素时应注意;

1下标要紧跟在数组名后，而且用圆括号括起来（不能用其他括号）

2下标可以是常量，变量，或表达式，但其值必须是整数（如果是小数将四舍五入为整数)

3下标必须为一段连续的整数，其最小值成为下界，其最大值成为上界。不加说明时下界值默认为1

**内存：**

Java把内存划分成两种：一种是栈内存，一种是堆内存

**栈内存**

在方法中定义的一些基本类型的变量和对象的引用变量都在方法的栈内存中分配，当在一段代码中定义一个变量时，java就在栈内存中为这个变量分配内存空间，当超出变量的作用域后，java会自动释放掉为该变量所分配的内存空间。

**堆内存**

堆内存用来存放由new运算符创建的对象和数组，在堆中分配的内存，由java虚拟机的自动垃圾回收器来管理。在堆中创建了一个数组或对象后，同时还在栈内存中定义一个特殊的变量。让栈内存中的这个变量的取值等于数组或者对象在堆内存中的首地址，栈中的这个变量就成了数组或对象的引用变量，引用变量实际上保存的是数组或对象在堆内存中的地址（也称为对象的句柄），以后就可以在程序中使用栈的引用变量来访问堆中的数组或对象。

**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

例:

int[] arr1={1,2,3,4,5,6};

for(int j = 0; j<=arr1.length;j++){

System.out.println(arr1[j]);

}

或：

for(int element:arr1){

System.out.println(element);

}

int[] arr2 = new int[6];

for(int I = 0; i<=5;i++){

arr2[i] = i+1;

System.out.println(arr2[i]);

}

**数组应用：**

查找：

顺序查找：

for(int i = 0;i<=arr.length-1;i++){

if(arr[i]==1){

System.out.println(i+1);

}

}

二分查找：

int n = 0;

int m = arr.length;

int p = (m+n)/2;

int i= 4;

while(i!=arr[p]){

if(i>arr[p]){

n = p+1;

}

If(i<arr[p]){

m = p-1;

}

P = (m+n)/2;

}

System.out.println(p);

如何互换值：

int temp;

temp = a;

a = b;

b = temp;

排序算法：

1. 选择排序
2. 插入排序
3. 归并排序
4. 快速排序

插入排序：

**int**[] arr = {9,9,1,0,1,2};

**for**(**int** x = 0;x<=arr.length-1;x++){

**for**(**int** y = x+1;y > 0 && y <= arr.length-1; y--){

**if**(arr[y]<arr[y-1]){

**int** temp;

temp = arr[y];

arr[y] = arr[y-1];

arr[y-1] = temp;

}

}

}

**for**(**int** i = 0;i<=arr.length-1;i++){

System.***out***.println(arr[i]);

}

黑板演示一步一步。

选择排序：

**for**(**int** i=0;i<=arr.length-1;i++){

Integer max = arr[i];

**int** maxPo = i;

**for**(**int** j=i+1;j<=arr.length-1;j++){

**if**(max>a[j]){

max = a[j];

maxPo = j;

}

}

int temp;

temp = arr[i];

arr[i] = arr[maxposition];

arr[maxposition] = temp;

}