**数组、排序、多维数组**

**数组：**

一.为什么需要数组

因为程序中每一个变量只能存储一项数据,如果存储多项数据,使用多个变量效率太低,因此需要使用一种能够存储多项数据的数据类型即数组

二、数组定义

数组:可以存储多个同一类型的数据集合

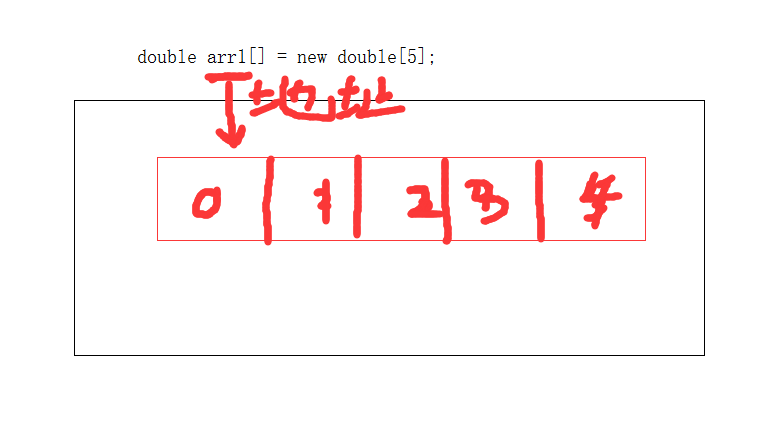
三、数组特点：

1、能够存多项数据

2、存储的数据必同种类型

四、数组的使用方法;

案例：养了五只小鸡，每只小鸡的体重分别为 2kg 、1.4kg、2.2kg、1.8kg、5kg，以一个数组来存放每只小鸡的体重，并计算它们的平均体重



1.声明

a. 数据类型[] 数组名称 = new 数据类型[数组的长度]

**int[] arr1 = new int[4];**

b. 数据类型 数组名称[] = new 数据类型[数组的长度]

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

2.初始化(给数组装数据)

1、直接赋值

数据类型[] 数组名称 = {数组中的数据,每项数据逗号分隔}

2、通过下标赋值

数组名称[下标] = 值

注：数组的下标是从0开始

五、数组的遍历

**for(int i = 0;i<6;i++){**

**int tem = arr[i];**

**}**

数组的大小arr.length

同学们思考一个问题，即然int char double这些基本类型都有数组来存放，那么可不可以有对象数组呢

Eg:Dogs dogs[] = new Dogs[4];

//使用输入流创建一个dogs数组

//使用set和get方法

找出体重最大的狗是第几只狗，打出它的体重，输入狗的名字，打出狗的体重

Eg:

总结：

1. 数组可以存放同一类型的数据(基本数据类型/引用数据类型)
2. 简单数据类型可直接赋值,引用数据类型需要创建实例
3. 对象数组在定义后，赋值时需要再次为每个对象分配空间(即 new 对象)
4. 数组大小必须事先指定,指定后不能更改
5. 数组也是一种引用数据类型(数组名存放的是一个引用地址)
6. 数组下标是从0开始编号的

**Java排序**

排序是指将一群数据，依照指定的规则进行排列的过程

排序分为内部排序和外部排序

1、内部排序：把数据加载到内存中进行排序(交换式排序、选择排序、插入排序)

2、外部排序：数据量过大无法全部加载到内存中，需要借助外部存储进行排序，包括(合并排序法、直接合并排序法)

我们常见的是内部排序

1. 交换式排序

交换式排序，是运用数据值比较后，依判断规则对数据位置进行交换，以达到排序的目的，我们来学习交换式排序中的冒泡排序

冒泡排序

冒泡排序是一种交换排序算法。使用相邻元素比对与交换的方式找到当前子序列（每一趟排序的序列比前一趟减少一个最末元素）中的最大或最小值并且将其移动到当前序列最末位置的方法最终获得有序序列

数组 第一次 第二次 第三次 第四次 第五次

23 75 75 75 75

56 23 23 56 56 56

46 56 23 46 46

12 46 46 23 23

22 12 22 22 22

75 22 12 12 12



eg: **public** **void** mao\_pao\_sort(**int** arr[]){

**int** tem = 0;

//外层循环，决定一共走几趟

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

//内层循环，开始逐个比较，如果符合前一个数大于后一个数则交换

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

**if**(arr[j]>arr[j+1]){

//使用临时变量进行交换

tem = arr[j];

arr[j] = arr[j+1];

arr[j+1] = tem;

}

}

}

}

1. 选择排序

选择排序也属于内部排序，是从欲排序的数组中，按指定的规则选出某一元素，和其它元素比较后，再依原则交换位置后达到排序的目的。

注：选择排序比冒泡排序速度快一些

初始状态： 34 12 35 56 16 42 54

第一次 ： 12 34 35 56 16 42 54

第二次 ： 12 16 35 56 34 42 54

第三次 ： 12 16 34 56 35 42 54

第四次 ： 12 16 34 35 56 42 54

第五次 ： 12 16 34 35 42 56 54

第六次 ： 12 16 34 35 42 54 56

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

**int** min\_tem = arr[i];

**int** minIndex = i;

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

**if**(min\_tem>arr[j]){

min\_tem = arr[j];

minIndex = j;

}

}

arr[minIndex] = arr[i];

arr[i] = min\_tem;

}

插入排序：

基本思想：把n个待排序的元素看成一个有序表和一个无序表，开始时有序表中只包含一个元素，无序表中包含n-1个元素，排序时每次从无序表中取出第一个元素，把它的排序码依次与有序表中的排序码进行比较，将它插入到有序表中的适当位置，使之成为新的有序表

初 始 值 (34) 12 45 3 54 22

第 一 次 (12 34) 45 3 54 22

第二次 (12 34 45) 3 54 22

第三次 (3 12 34 45) 54 22

第四次 (3 12 34 45 54) 22

第五次 (3 12 22 34 45 54)

**public** **void** sort\_charu(**int** arr[]){

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

**int** inserVal = arr[i+1];

**int** index = i;

**while**(index>=0&&inserVal<arr[index]){

//把arr[index]向后移动一位

arr[index+1] = arr[index];

//让index向前移动一位

index-- ;

}

//将inserVal插入到适当的位置

arr[index+1] = inserVal;

}

}

多维数组：

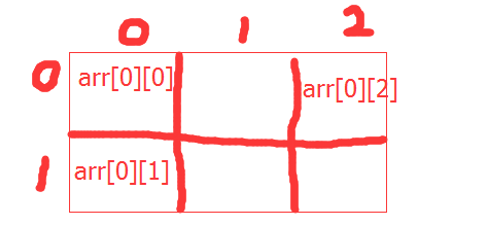
二维数组

定义：

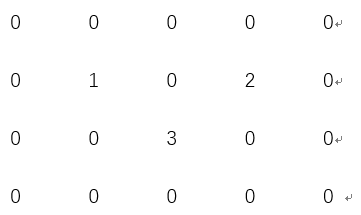
类型 数组名[][] = new 类型[大小][大小]

Eg: int arr[][] = new int[2][3];

分析：内存中状态



案例：请用二维数组输出以下图形：



习题：

1. 題目：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总共有多少对？ System.out.println(第几个月，有多少对兔子);

2、写一个程序完成二维数组的转置，并输出

1 2 3 0 1 4 7

4 5 6 0 转置后： 2 5 8

1. 8 9 0 3 6 9

0 0 0

3求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加由键盘控制。

4、题目：有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。