**数组：<类型>[] <名称> = new <类型> [元素个数]**

**数组变量是数组的管理者，不是数组本身（数组变量是管理者，其他变量类型是所有者）**

**数组必须创建出来然后交给数组变量来管理**

**数组变量之间的赋值是管理权限的赋予**

**数组之间的比较是判断是否管理同一数组**

**for each的语句格式：**

**for(元素类型 元素变量x : 遍历对象obj)**

**{**

**引用了x的java语句;**

**}**

**主要用于遍历，将数组中的元素一次次拿出来给x，改变x不能改变数组**

### 例一：

**初识数组**

**代码**

**package** Array;

**import** java.util.Scanner;

**public** **class** 初识数组 {

**private** **static** Scanner *in*;

**public** **static** **void** main(String[] args) {

//初识数组

*in* = **new** Scanner(System.***in***);

//声明数组变量，创建数组

//数组名称my，数组大小5，数组下标范围（0-4）

**double**[] my = **new** **double**[5];

//输出数组长度

System.***out***.println("数组长度：" + my.length);

//输入数组中的元素

**for**(**int** i=0; i<my.length; i++)

{

System.***out***.print("请输入第" + i + "个数字：");

my[i] = *in*.nextInt();

}

//输出数组中的元素

**for**(**int** i=0; i<my.length; i++)

System.***out***.print("\t第" + i + "个数字：" + my[i]);

// 计算所有元素的总和

**double** sum = 0;

**for** (**int** i=0; i<my.length; i++)

sum += my[i];

System.***out***.println();

System.***out***.println("总和为： " + sum);

//查找最大元素

**double** max = 0;

**for**(**int** i=0; i<my.length; i++)

**if**(my[i] > max)

max = my[i];

System.***out***.println("最大值为： " + max);

}

}

### 例二：

**数组变量**

**代码**

**package** Array;

**public** **class** 数组变量 {

**public** **static** **void** main(String[] args) {

//数组变量

// 数组变量是管理者

// a与b共同管理同一数组

// 开始只有a对数组进行管理，当int[] b = a;后，a与b共同管理

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

//输出数组a中的元素

System.***out***.print("数组a中的元素：");

**for**(**int** i=0; i<a.length; i++)

System.***out***.print("\t" + a[i]);

System.***out***.println();

**int**[] b = a;

b[0] = 10;

//输出数组a中的元素

System.***out***.print("数组a中的元素：");

**for**(**int** i=0; i<a.length; i++)

System.***out***.print("\t" + a[i]);

System.***out***.println();

//输出数组b中的元素

System.***out***.print("数组b中的元素：");

**for**(**int** i=0; i<b.length; i++)

System.***out***.print("\t" + b[i]);

System.***out***.println();

// 数组之间的比较是判断是否管理同一数组

**int**[] c = {0,1,2,3,4};

System.***out***.println(a == b);

System.***out***.println(a == c);

// 复制数组必须遍历源数组将每个元素逐一拷贝给目的数组

**for**(**int** i=0; i<c.length; i++)

c[i] = a[i];

//输出数组c中的元素

System.***out***.print("数组c中的元素：");

**for**(**int** i=0; i<c.length; i++)

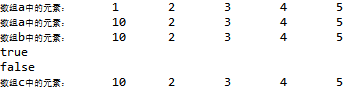
System.***out***.print("\t" + c[i]);

System.***out***.println();

}

}

**结果**



### 例三：

**投票统计**

**代码**

**package** Array;

**import** java.util.Scanner;

**public** **class** 投票统计 {

**private** **static** Scanner *in*;

**public** **static** **void** main(String[] args) {

//投票统计

*in* = **new** Scanner(System.***in***);

**int**[] num = **new** **int**[10];

System.***out***.println("请输入：");

**int** x = *in*.nextInt();

**while**(x != -1)

{

**if**(x>=0 && x<=9)

num[x]++;

x = *in*.nextInt();

}

**for**(**int** i=0; i<num.length; i++)

System.***out***.println(i + ": " + num[i] + "次");

}

}

### 例四：

**For each循环**

**代码**

**package** Array;

**public** **class** foreach循环 {

**public** **static** **void** main(String[] args) {

//for each循环

**double**[] num = {1.2,2.3,3.4,4.5};

**for**(**double** x : num)

System.***out***.println(x);

}

}

### 例五：

**二维数组 井字棋**

**代码**

**package** Array;

**import** java.util.Scanner;

**public** **class** 二维数组井字棋 {

**private** **static** Scanner *input*;

//i表示行数，j表示列数

**static** **int**[][] *x*=**new** **int**[3][3];

**static** **int**[][] *o*=**new** **int**[3][3];

**static** **int** *row*;

**static** **int** *rank*;

//判断

**public** **static** **boolean** judge(**int**[][] m,**int** i,**int** j)

{

**int** num0=0;

**int** num1=0;

**int** num2=0;

**int** num3=0;

**for**(**int** l=0;l<3;l++)

{

//正对角线

**if**(i==j)

**if**(m[l][l]==1)

num0+=1;

//反对角线

**if**(i==2-j)

**if**(m[l][2-l]==1)

num1+=1;

//行

**if**(m[i][l]==1)

num2+=1;

//列

**if**(m[l][j]==1)

num3+=1;

}

**if**(num0==3||num1==3||num2==3||num3==3)

**return** **true**;

**else**

**return** **false**;

}

//输入位置

**static** **void** position(**int**[][] xo,**int** m,**char** player)

{

//记录玩家是o或x

//判断玩家

*input*=**new** Scanner(System.***in***);

System.***out***.print("玩家"+player+"输入行数：");

*row*=*input*.nextInt()-1;

System.***out***.print("玩家"+player+"输入列数：");

*rank*=*input*.nextInt()-1;

//判断该坐标上是否有棋子,如果有就重新输入,没有就赋值并输出

**if**(*row*>2||*rank*>2||*row*<0||*rank*<0)

{

System.***out***.println("行数列数必须为1到3之间整数，请重新输入");

*position*(xo,m,player);

}

**else**

{

**if**(*x*[*row*][*rank*]+*o*[*row*][*rank*]>0)

{

System.***out***.println("坐标重复，请重新输入");

*position*(xo,m,player);

}

**else**

{

xo[*row*][*rank*]=1;

*printMap*();

}

}

}

//打印棋盘

**public** **static** **void** printMap()

{

**for**(**int** i=0;i<3;i++)

{

**for**(**int** j=0;j<3;j++)

{

**if**(*x*[i][j]==1)

System.***out***.print(" x |");

**else**

**if**(*o*[i][j]==1)

System.***out***.print(" o |");

**else**

System.***out***.print(" |");

}

System.***out***.println("");

}

}

**public** **static** **void** main(String[] args)

{

**boolean** res=**false**;

**char** player;

//调用方法输入并判断胜利否

**for**(**int** i=1;i<10;i++)

{

**int** m=i%2;

//输入数值

**if**(m==0)

{

player='x';

*position*(*x*,m,'x');

res=*judge*(*x*,*row*,*rank*);//判断胜利否

}

**else**

{

player='o';

*position*(*o*,m,'o');

res=*judge*(*o*,*row*,*rank*);}//判断胜利否

**if**(res)

{

System.***out***.println("玩家"+player+"获胜");

**break**;

}

**else** **if**(i==9)

{

System.***out***.println("平局");

}

}

}

}