VECTORES (ARREGLOS)

Introducción

Una array es un conjunto de variables indexadas y que forman en una sola y sencilla súper variable que ofrece una manera fácil de pasar varios valores entre las líneas de código, funciones, e incluso páginas. Durante gran parte de este capítulo, vamos a ver el funcionamiento interno de los arreglos.

Java posee la capacidad de definir un conjunto de variables del mismo tipo, dichas variables están agrupadas bajo un mismo nombre, y se las distingue por un número (índice). El tamaño del array se establece cuando se crea.

A los elementos del array se accederá a través de la posición que ocupan dentro del conjunto de elementos del array.

Este concepto es comúnmente conocido como Vector. En Java para utilizar los vectores se utilizan los Arrays para definir un array en java es cómo definir una variable o atributo, pero al especificar el tipo lo que hacemos es colocar un par de corchetes [] para indicar que lo que estamos definiendo es un array.

Por ejemplo:

tipo identificador [];

O bien

tipo [] identificador;

donde:

- tipo es el tipo de dato de los elementos del vector
- identificador es el nombre de la variable.

Creación

Los arrays se crean con el operador new, de la siguiente manera:

[code]

vector= new tipo[elementos];

[/code]

Entre corchetes se indica el tamaño del vector, donde tipo debe coincidir con el tipo con el que se haya declarado el vector.

vector debe ser una variable declarada como tipo[]

Ejemplos

[code]

float[] notas= new float[4];

int[] temperaturas= new int[7];

[/code]

Usos

Para acceder a los elementos de un array, utilizamos índices (para indicar la posición del elemento dentro del array)

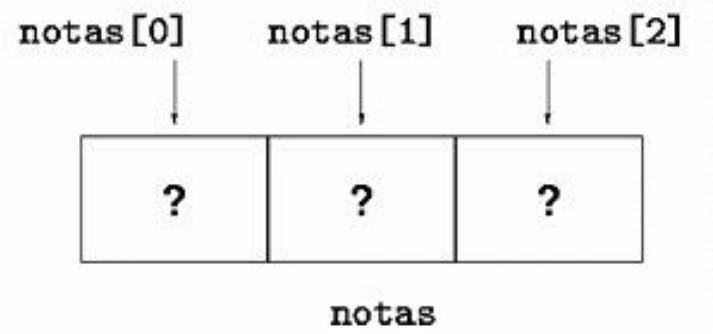
vector[índice]

En Java, el índice de la primera componente de un vector es siempre 0.

El tamaño del array puede obtenerse utilizando la propiedad vector.length. Por lo tanto, el índice del último componente es vector.length-1.

Por Ejemplo:

float[] notas = new float[3]



Inicialización en la declaración

Podemos asignarle un valor inicial a los elementos de un array en la propia declaración.

Si se conocen los valores de antemano se podría crear de la siguiente forma:

int[] vec = {150,500,3,4,5,6};

El compilador deduce automáticamente las dimensiones del array.

Ejemplo definimos un array de enteros llamado losValores que tendrá 10 elementos.

```
[code]
public static void main(String[] args) {
      int [] losValores = new int[10];
      losValores[0]= 1;
      losValores[1]= 34;
      losValores[2]= 191;
      losValores[3]= 9878;
//….continua hasta 9
      losValores[9]= 232;

      System.out.print("El valor de la primera posición es: "+ losValores[0]);
   }
[/code]
```

Recorrido del array

Para recorrer el array sería por ejemplo:

```
[code]
   for(i=0; i<losValores.length; i++)
         {System.out.println(losValores[i]);}
[/code]
```

Si quisiéramos recorrer el array de atrás hacia adelante seria de la siguiente manera:

[code]

```
 for(i=losValores.length-1; i>=0; i--)
        {System.out.println(losValores[i]);}
```

No es necesario utilizar todos los elementos de un vector, por lo que, al trabajar con ellos, se puede utilizar una variable entera adicional que nos indique el número de datos que realmente estamos utilizando.

El tamaño del vector nos dice cuanta memoria se ha reservado para almacenar datos del mismo tipo, no cuántos datos del mismo tipo tenemos realmente en el vector.

Ejemplo: Suma de los n primeros elementos de un vector

```
     int suma=0;
    int n=losValores.length;
    for (i=0; i<n; i++)
       {
       suma= suma+ losValores[i];
       }
    System.out.println(suma);
```

Copia de arrays

```
int pares[]={2,4,6,8,10};
```

Para copiar los elementos de un array, podemos crear un nuevo array y copiar los elementos uno a uno.

```
    int []datos= new int[pares.length];
     for (i=0; i<pares.length;i++)
        {datos[i]=pares[i];}
```

También podemos utilizar una función predefinida en la biblioteca de estándar de Java:

```
System.arraycopy(from, fromIndex, to, toIndex, n);
int []datos= new int[pares.length];
System.arraycopy(pares, 0, datos, 0, pares.length);
```

La biblioteca de clases de Java incluye una clase auxiliar llamada java.util.Arrays que incluye como métodos algunas de las tareas que se realizan más a menudo con vectores:

nota: es necesario importar la biblioteca, import java.util.Arrays;
Arrays.sort(v) ordena los elementos del vector.
Arrays.equals(v1,v2) comprueba si dos vectores son iguales.

Arrays.toString(v) devuelve una cadena que representa el contenido del vector.

Por Ejemplo:

Arrays.sort(losValores);

Búsqueda del número máximo y mínimo de un vector

[code]

```
 int maximo=vec[0],minimo=vec[0];
// si solo voy calcular el maximo y minimo puedo recorrer desde 1
for (int a=1;a<vec.length;a++){
   if (vec[a]>maximo)
      maximo=vec[a];
   if (vec[a]<minimo)
      minimo=vec[a];
}
System.out.println("El maximo es : "+maximo);
System.out.println("El minimo es : "+minimo);
```

[/code]

Contar cuantas veces aparece el número 10

[code]

```
int contador =0;
for(int a=0;a<vec.length;a++){
   if (vec[a]==10)
   contador ++;
}
System.out.println("El numero 10 aparece :"+contador+" veces")
```

[/code]

Totalizar un vector y calcular el promedio

[code]

```
int total=0;
for(int a=0;a<vec.length;a++){
   total+=vec[a];
}
System.out.println("La suma total es:"+total);
System.out.println("El promedio es:"+total/vec.length);
```

[/code]