2.F. Anexo I.- Introducción a los arrays.

1. Introducción a los arrays.

El explicar el concepto de array es algo prematuro a estas alturas de curso, pero de cara a poder realizar ciertos algoritmos y programas es necesario disponer de ciertas nociones básicas, que por otro lado, al nivel que contaremos, no serán complejas. La idea es en sí sencilla, pero debemos disponer de algunos conocimientos básicos para poder representarlos. Más adelante, durante el curso procederemos a entrar en más detalle en su descripción. De momento ahí van estas "pinceladas", que te permitirán poder hacer algunos de los ejercicios propuestos en esta unidad:

Los arrays permiten almacenar una colección de objetos o datos del mismo tipo. Son muy útiles y su utilización es muy simple:

- **Declaración del array**. La declaración de un array consiste en decir "esto es un array" y sigue la siguiente estructura: "tipo[] nombre;". El tipo será un tipo de variable o una clase ya existente, de la cual se quieran almacenar varias unidades.
- **Creación del array**. La creación de un array consiste en decir el tamaño que tendrá el array , es decir, el número de elementos que contendrá, y se pone de la siguiente forma: "nombre=new tipo[dimension]", donde dimensión es un número entero positivo que indicará el tamaño del array . Una vez creado el array este no podrá cambiar de tamaño.

Veamos un ejemplo de su uso:

int[] n; // Declaración del array.

n = new int[10]; //Creación del array reservando para el un espacio en memoria.

int[] m=new int[10]; // Declaración y creación en un mismo lugar.

Una vez hecho esto, ya podemos almacenar valores en cada una de las posiciones del array, usando corchetes e indicando en su interior la posición en la que queremos leer o escribir, teniendo en cuenta que la primera posición es la cero y la última el tamaño del array menos uno. En el ejemplo anterior, la primera posición sería la 0 y la última sería la 9.

La modificación de una posición del array se realiza con una simple asignación. Simplemente se especifica entre corchetes la posición a modificar después del nombre del array. Veámoslo con un simple ejemplo:

int[] Numeros=new int[3]; // Array de 3 números (posiciones del 0 al 2).

Numeros[0]=99; // Primera posición del array.

Numeros[1]=120; // Segunda posición del array.

Numeros[2]=33; // Tercera y última posición del array.

El acceso a un valor ya existente dentro de una posición del array se consigue de forma similar, simplemente poniendo el nombre del array y la posición a la cual se quiere acceder entre corchetes:

int suma=Numeros[0] + Numeros[1] + Numeros[2];

Para nuestra comodidad, los arrays, como objetos que son en Java, disponen de una propiedad pública muy útil. La propiedad length nos permite saber el tamaño de cualquier array, lo cual es especialmente útil en métodos que tienen como argumento un array.

System.out.println("Longitud del array: "+Numeros.length);

El tercer uso principal de los arrays es en el paso de parámetros. De nuevo, se trata de un concepto algo avanzado y sobre el que trataremos más adelante, por lo que sólo debes entender, de momento, el concepto. Para pasar como argumento un array a una función o método, esta debe tener en su definición un parámetro declarado como array. Esto es simplemente que uno de los parámetros de la función sea un array. Vamos a ver un ejemplo:

int sumaarray (int[] j) {

int suma=0;

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

suma=suma+j[i];

return suma;

}

En el método anterior se pasa como argumento un array numérico, sobre el cual se calcula la suma de todos los números que contiene. Es un

uso típico de los arrays, fíjate que especificar que un argumento es un array es igual que declarar un array, sin la creación del mismo. Para pasar como argumento un array a una función, simplemente se pone el nombre del array:

```
int suma=sumaarray (Numeros);
```