6.D. Arrays unidimensionales.

1. Arrays unidimensionales.

1.1. Uso de arrays unidimensionales.

Ya sabes declarar y crear de arrays, pero, ¿cómo y cuando se usan? Pues existen tres ámbitos principalmente donde se pueden usar, y los tres son muy importantes: modificación de una posición del array, acceso a una posición del array y paso de parámetros.

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. Veamoslo 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, como se dijo antes, es en el paso de parámetros. 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. Veamos 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, fijate 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);
```

En Java las variables se pasan por copia a los métodos, esto quiere decir que cuando se pasa una variable a un método, y se realiza una modificación de su valor en dicho método, el valor de la variable en el método desde el que se ha realizado la invocación no se modifica. Pero cuidado, eso no pasa con los arrays. Cuando dicha modificación se realiza en un array, es decir, se cambia el valor de uno de los elementos del array, si que cambia su valor de forma definitiva. Veamos un ejemplo que ilustra ambas cosas:

```
public static void main(String[] args) {
    int j=0; int[] i=new int(1);  i[0]=0;
    modificaArray(j,i);
    System.out.println(j+"-"+i[0]);  /* Mostrará por pantalla "0-1", puesto que el contenido del array es
```

```
        modificado en la función, y aunque la variable j también se modifica, se modifica una copia de la misma,
        dejando el original intacto */
}

int modificaArray(int j, int[] i); {
    j++; int[0]++;  /* Modificación de los valores de la variable, solo afectará al array, no a j */
}
```