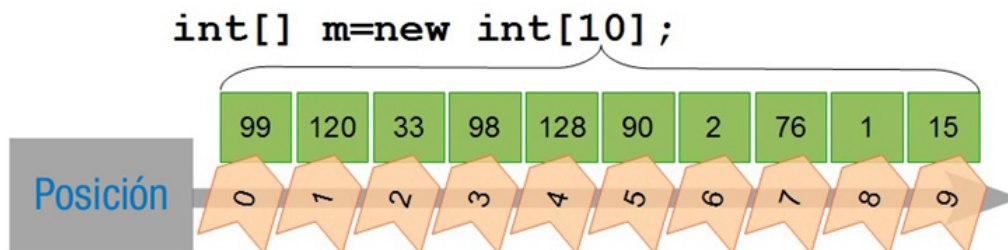


## 6.D. Arrays unidimensionales.

### 1. Arrays unidimensionales.

Todos los lenguajes de programación permiten el uso de arrays, veamos como son en Java.



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[dimensión]", 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 él 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.

#### Autoevaluación

¿Cuáles de las siguientes opciones permitiría almacenar más de 50 números decimales?

- ☐ `int[] numeros; numeros=new int[51];`
- ☐ `int[] numeros; numeros=new float[52];`
- ☐ `double[] numeros; numeros=new double[53];`
- ☐ `float[] numeros=new float[54];`

Resolver