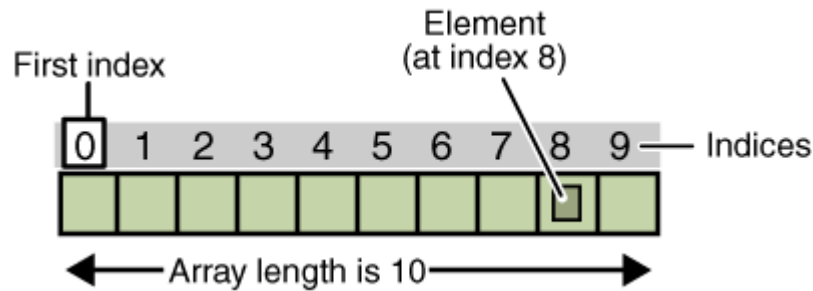


ARRAYS

Un *array* es un contenedor de objetos que mantiene un número fijo de valores de un mismo tipo. La longitud de un array se establece cuando este es creado. Después de su creación, su tamaño no puede variar.



Un array de 10 elementos

Para referirse a un elemento dentro de un array se utiliza su índice numérico. La numeración siempre comienza en 0. Si queremos referirnos al noveno elemento, por ejemplo, utilizaremos el índice 8.

Declaración de Arrays

Para declarar una variable que contenga un array, se indica el tipo base (el tipo de cada elemento del array) seguido por los caracteres [] y a continuación el identificador que asociaremos a esa variable:

```
double[] arrayDeDoubles;
```

Esta declaración no crea el array, solo le indica al compilador que esa variable contendrá un array del tipo especificado.

Otra forma de declararlos es poniendo los corchetes después del identificado, en lugar de a continuación del tipo:

```
double arrayDeDoubles[];
```

Esta segunda forma (la utilizada en C) sin embargo, se recomienda no utilizarla.

Creación, Inicialización y Acceso

Una manera de crear un array es mediante el operador `new`, indicando a continuación del operador el tipo base y entre corchetes el tamaño (número de elementos) del array:

```
arrayDeDoubles = new double[4];
```

Se puede juntar la declaración con la creación:

```
double arrayDeDoubles[] = new double[4];
```

Cada elemento del array puede ser accedido indicando después del nombre del array, entre corchetes, su posición dentro del mismo (recordar que se empieza a numerar en 0).

Para inicializar el array se pueden asignar valores elemento a elemento:

```
arrayDeDoubles[0] = 1.0;    // inicializa el primer elemento  
arrayDeDoubles[1] = 3.4;    // inicializa el segundo elemento  
arrayDeDoubles[2] = 4.56;  
arrayDeDoubles[3] = 0.0;
```

De modo alternativo se puede utilizar una sintaxis abreviada para declarar, crear e inicializar el array:

```
double arrayDeDoubles[] = { 1.0, 3.4, 4.56, 0.0 };
```

Here the length of the array is determined by the number of values provided between `{` and `}`.

Arrays Multidimensionales

Se pueden declarar arrays de arrays, conocidos como arrays multidimensionales usando dos o más conjuntos de corchetes (uno por cada dimensión del array). Para acceder a un elemento se utilizará un índice por cada dimensión.

Por ejemplo, la siguiente matriz:

$$\begin{bmatrix} 7 & 4 & 9 \\ 8 & 1 & 5 \end{bmatrix}$$

La podríamos implementar en Java del siguiente modo:

```
int[][] matriz = { {7, 4, 9}, {8, 1, 5}};
```

O bien declararla y crearla de siguiente modo:

```
int[][] matriz = new int[2][3]; // 2 filas x 3 columnas
```

Y a continuación inicializar cada componente.

Si quisiésemos imprimir el valor de la tercera columna de la primera fila (el 9):

```
System.out.println(matriz[0][2]);
```

En Java, a diferencia de C, un array multidimensional es simplemente un array cuyos components son a su vez arrays. Una consecuencia de ello es que las filas pueden variar en longitud.

```
String[][] nombres = {  
    {"Mr. ", "Mrs. ", "Ms. "},  
    {"Smith", "Jones"}  
};  
System.out.println(nombres[0][0] + nombres[1][0]); //Mr. Smith  
System.out.println(nombres[0][2] + nombres[1][1]); //Ms. Jones
```

Para determinar el tamaño de un array se puede utilizar el método `length`. Aplicado al ejemplo anterior:

```
System.out.println(nombres.length);  
System.out.println(nombres[0].length);  
System.out.println(nombres[1].length);
```

Imprimiría 2 3 2, que son respectivamente, el número de filas, el número de columnas de la primera fila, y el número de columnas de la segunda fila.

Copia de Arrays

Para copiar un array podemos hacer una copia elemento a elemento, pero disponemos de un método de la clase `System` que simplifica la tarea, realizando una copia eficiente de los datos desde un array a otro. Es el método `arraycopy`. Su prototipo es:

```
public static void arraycopy(Object src,
                             int srcPos,
                             Object dest,
                             int destPos,
                             int length)
```

Los dos argumentos `Object` los arrays origen y destino respectivamente. Los tres argumentos `int` especifican la posición de comienzo en el array origen, la posición de destino en el array destino, y el número de elementos a copiar.

El siguiente ejemplo declara un array de caracteres cuyos componentes son las letras de la palabra 'programador'. Utiliza `arraycopy` para rellenar un Segundo array con los caracteres situados a partir de la sexta posición (recordar que se comienza a numerar en 0):

```
char[] origen = {'p', 'r', 'o', 'g', 'r', 'a', 'm',
                 'a', 'd', 'o', 'r'};
char[] destino = new char[6];

System.arraycopy(origen, 5, destino, 0, 6);
System.out.println(new String(destino));
```

Imprimirá:

```
amador
```