Asignatura

Programación

UNIVERSAE
Instituto Superior de FP

Asignatura

Programación

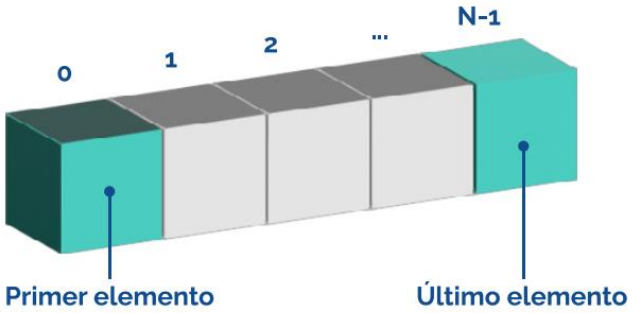
UNIDAD 8

Estructura de almacenamiento Arrays y cadena de caracteres

UNIVERSAE
Instituto Superior de FP

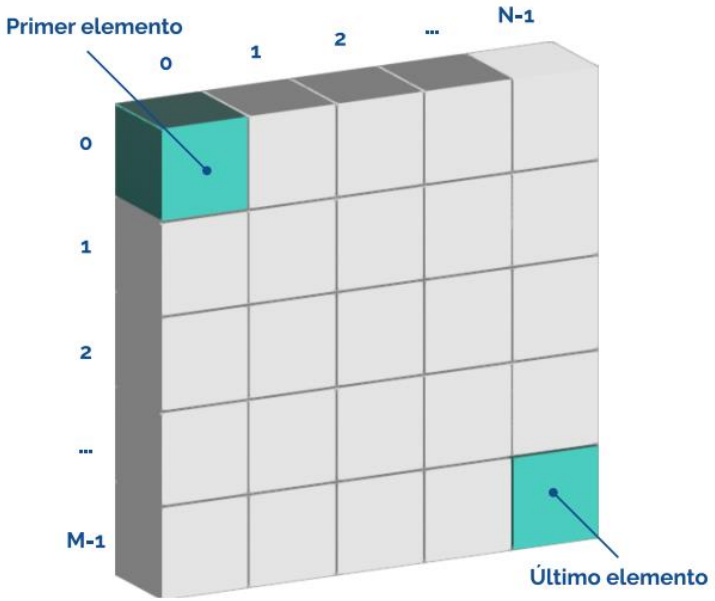
Arrays

- Objeto estático
- Contiene elementos de forma secuencial
- Número finito de elementos
- Todos los elementos son del mismo tipo



Tipos de datos

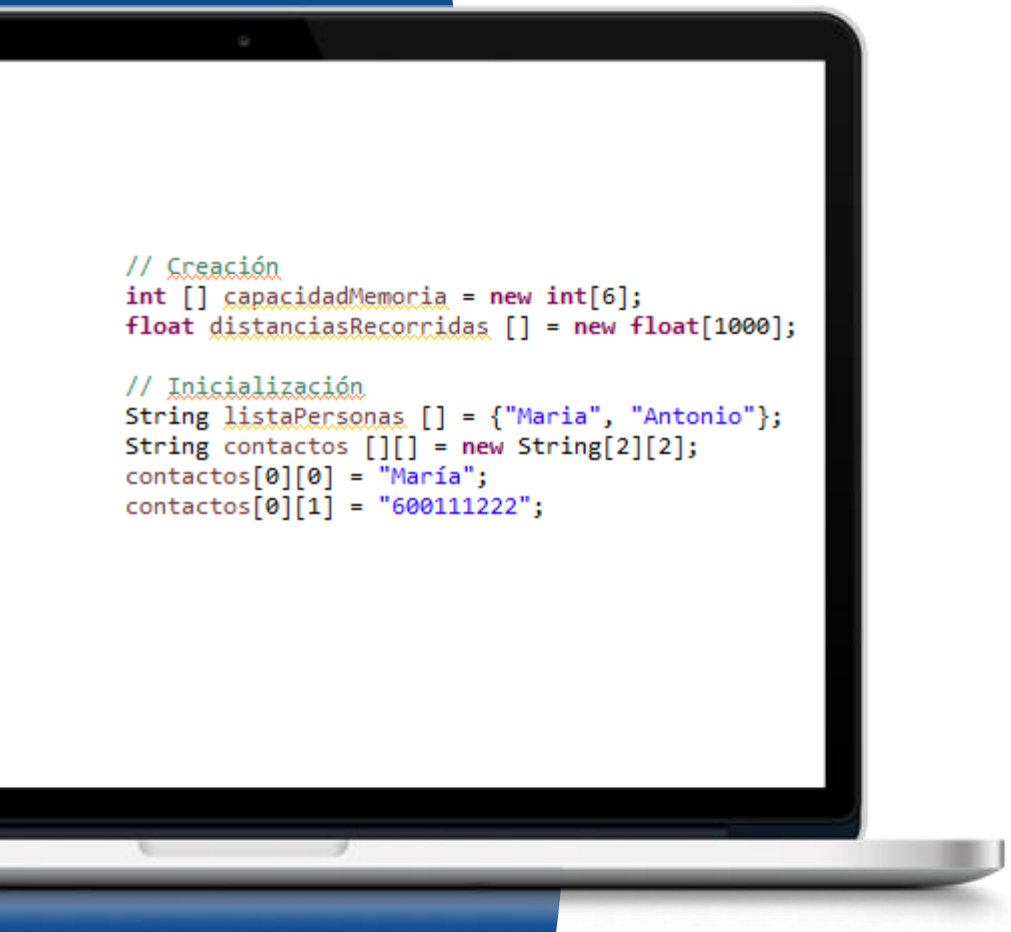
- **Primitivo**: Int, long, double, boolean, etc..
- **Abstracto o referencia.** Contiene objetos, en su defecto, null



Dimensiones

- **Unidimensional.** Representa una fila de forma lineal
- **Multidimensional o Matriz.** Representa una tabla, con columnas y filas.

Arrays. Declaración e inicialización



Creación

- **Sintaxis array unidimensional**
  - *tipo [ ] NOMBRE = new tipo [NUMERO DE ELEMENTOS]*
  - *tipo NOMBRE [ ] = new tipo [NUMERO DE ELEMENTOS]*
- **Sintaxis array multimedimensional**
  - *tipo [ ][ ] NOMBRE = new tipo [NUMERO DE ELEMENTOS] [NUMERO DE ELEMENTOS]*
  - *tipo NOMBRE [ ][ ] = new tipo [NUMERO DE ELEMENTOS] [NUMERO DE ELEMENTOS]*

Inicialización

- **En la misma declaración**
  - *tipo [ ] NOMBRE = { valor1, valor2 }*
  - *tipo [ ] [ ] NOMBRE = {[valor1, valor2],[valor3,valor4]}*
- **Elemento a elemento.**
  - NOMBRE[posición] = valor1;
  - NOMBRE[posición][posición] = valor1;

Arrays. Acceso y recorrido

Acceso

- El primer elemento tiene la posición 0
- El último elemento será el número total de elementos -1
- Ejemplo, acceder al elemento 5

  array[4];
- Para conocer la posición del último elemento se usa la propiedad **.length** del array

Recorrido

- Array unidimensional. Solo un bucle

```
String listaPersonas [] = {"Maria", "Antonio"};

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

}
```

- Array multidimensional. Un bucle dentro de otro

```
String contactos [][] = new String[2][2];
contactos[0][0] = "María";
contactos[0][1] = "600111222";

for (int i = 0; i < contactos.length; i++) {
    for (int j = 0; j < contactos[i].length; j++) {

    }
}
```

Arrays. Operaciones

Operaciones varias

- Funciones propias del array
- Uso de la clase ***java.util.Arrays***
- O funciones propias del sistema. ***System***

Función	Descripción
binarySearch(Array, valor)	Busca el valor y devuelve la posición
toString(Array)	Devuelve un conjunto de cadenas de todo el array
sort(Array)	Ordena los elementos del array
equals(Array1, Array2)	Compara si tienen los mismos valores
fill(Array, valor)	Inicializa toda la array con el valor indicado
Array.clone()	Realiza una copia del array
copyOf(Array, posición)	Realiza una copia del array hasta el elemento de la posición indicada
System.arraycopy(Array, posición inicio, Array Copia, posición inicio, longitud)	Copia el array de origen hacia el array copia, se especifica la posición de inicio y la longitud total a copiar.

- **Copia Deep o profunda**
- **Copia shallow o poco profunda**

Operaciones de inserción y eliminación

- Costoso. Las arrays son estáticas
- Solución. Crear un nuevo array
  - Para inserción, aumentando los elementos totales
  - Para eliminación, reduciendo los elementos totales
- Uso de funciones como **arraycopy**

Cadena de caracteres

Las cadenas de caracteres

- Conjunto de caracteres que forman una palabra, frase, oración, etc.
- Array de tipo char[]
- Tipo String

Declaración

- char [ ] array = { 'H', 'o', 'l', 'a' };
- String saludo = "Hola";
- String saludo = String.copyValueOf(array)

Cadena de caracteres. Operaciones

Función	Descripción
charAt(posición)	Obtiene el caracteres de la posición dada
compareTo(Cadena)	Obtiene 0 si son iguales, negativo si la cadena precede a la anterior o positivo en caso contrario. (Mira si alfabéticamente es mayor o menor)
contains(cadena)	Indica si la cadena contiene el valor del parámetro
toCharArray()	Devuelve un array de tipo char de la cadena
concat(cadena)	Concatena la cadena. Igual a usar el operador +
toLowerCase()	Pasa a minúscula
toUpperCase()	Pasa a mayúsculas
length()	Obtiene el tamaño de la cadena
substring(posición inicial, posición final)	Extrae una parte de la cadena según la posición inicial y final
replace(cadena a buscar, cadena remplazo)	Busca la cadena y la remplaza por la indicada.
indexOf(Cadena)	Busca la primera ocurrencia y devuelve la posición de comienzo
lastIndexOf(Cadena)	Busca la última ocurrencia y devuelve la posición

Conversiones

- Es posible transformar un tipo de dato a cadena

De String a tipo primitivo

- Byte.parseByte(cadena);
- Short.parseShort(cadena);
- Integer.parseInteger(cadena);
- Long.parseLong(cadena);
- Float.parseFloat(cadena);
- Double.parseDouble(cadena);

De String a clase envoltorio

- Byte.valueOf(cadena);
- Short.valueOf(cadena);
- Integer.valueOf(cadena);
- Long.valueOf(cadena);
- Float.valueOf(cadena);
- Double.valueOf(cadena);

De tipo número a String

- String.valueOf(numero)

Resumen

1. Arrays
2. Arrays. Declaración e inicialización
3. Arrays. Acceso y recorrido
4. Arrays. Operaciones
5. Cadena de caracteres
6. Cadena de caracteres. Operaciones
7. Conversiones

UNIVERSAE

CHANGE YOUR WAY