

Clase 11 - Arreglos Multidimensionales

Taller de preparación para la OCI - PUC

Ya conocemos los arreglos, que nos sirve para coleccionar elementos

```
public class Main {  
    public static void main(String[] args) {  
        String[] nombres = {"Pedro", "Juan", "Diego"};  
    }  
}
```

Arreglos Multidimensionales

- Un arreglo en Java puede coleccionar elementos de distintos tipos, como Strings, números y valores booleanos
- Un arreglo en Java incluso puede coleccionar arreglos
- A los arreglos que coleccionan arreglos les llamamos arreglos multidimensionales

Pero profesor, para qué quiero un arreglo de arreglos?

Arreglos Multidimensionales

- Tablas
- Matrices
- Tableros
- Grafos
- ...

Arreglos Multidimensionales

En este caso `numeros` es un arreglo de arreglos

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int[][] numeros = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
  
    }  
  
}
```

Arreglos Multidimensionales

- En la posición 0 está el arreglo {1, 2, 3}
- En la posición 1 está el arreglo {4, 5, 6}
- En la posición 2 está el arreglo {7, 8, 9}

Accediendo a un elemento en particular

Vamos a acceder a un elemento en particular y lo vamos a imprimir

```
public class Main {  
    public static void main(String[] args) {  
        int[][] numeros = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
        int elemento = numeros[1][2];  
        System.out.println(elemento);  
    }  
}
```

En este caso accedemos al arreglo en la posición 1 (el {4, 5, 6}) y luego a la posición 2 en ese arreglo, que es sería el número 6

Editando un elemento en particular

Vamos a editar un elemento en particular

```
public class Main {  
    public static void main(String[] args) {  
        int[][] numeros = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
        numeros[2][0] = 10;  
    }  
}
```

¿Cómo se ve nuestro arreglo ahora?

Creando un arreglo

```
public class Main {  
    public static void main(String[] args) {  
        String[][] instructores = new String[4][3];  
        instructores[0][0] = "Andrés";  
        instructores[0][1] = "Fernández";  
        instructores[0][2] = "@andresfdezc";  
        instructores[1][0] = "Nebil";  
        instructores[1][1] = "Kawas";  
        instructores[1][2] = "@nebil";  
        instructores[2][0] = "Florencia";  
        instructores[2][1] = "Valladares";  
        instructores[2][2] = "@fvr1";  
        instructores[3][0] = "Adrián";  
        instructores[3][1] = "Soto";  
        instructores[3][2] = "@alanezz";  
    }  
}
```

Recorriendo un arreglo bidimensional

Necesitamos dos for, para recorrer las dos dimensiones del arreglo

```
public static void imprimir(String[][] matriz) {  
    for (int i = 0; i < matriz.length; i++) {  
        for (int j = 0; j < matriz[0].length; j++) {  
            System.out.println(matriz[i][j]);  
        }  
        System.out.println("###");  
    }  
}
```

Otro ejemplo para imprimir un arreglo

```
public static void imprimir(String[][] matriz) {  
    for (int i = 0; i < matriz.length; i++) {  
        String fila = "- ";  
        for (int j = 0; j < matriz[0].length; j++) {  
            fila = fila + matriz[i][j] + " - ";  
        }  
        System.out.println(fila);  
        System.out.println("###");  
    }  
}
```

Comparando arreglos

Esta función retorna true si dos arreglos son iguales

```
public static boolean comparar(String[][] matriz1, String[][] matriz2) {  
    if (matriz1.length == matriz2.length &&  
        matriz1[0].length == matriz2[0].length) {  
        for (int i = 0; i < matriz1.length; i++) {  
            for (int j = 0; j < matriz1[0].length; j++) {  
                if (matriz1[i][j] != matriz2[i][j]) {  
                    return false;  
                }  
            }  
        }  
        return true;  
    }  
    return false;  
}
```

Comparando arreglos

- Los arreglos NO se pueden comparar con ==
- Debemos comprobar que los largos sean los mismos para que no tire error

Clase 10 - Arreglos Multidimensionales

Taller de preparación para la OCI - PUC