

Repaso de arrays

1. Crea una clase llamada **Doscientos**.java con un array de enteros con 200 elementos (aleatorios entre 1 y 25) como atributo de la clase. Crea los siguientes métodos:
 - a. Buscar e imprime por pantalla el mayor de ellos.
 - b. Calcular el sumatorio de sus elementos.
 - c. Calcular la media de sus elementos
 - d. ¿Sabría recorrer un array de forma circular? Es decir, ¿cuándo llegue al último elemento que pase al primero?
 - e. Mostrar por pantalla los pares. ¿Sabría qué código debe escribir para mostrar el primer elemento del array que sea par y salir del bucle que lo recorre?
 - f. Pide al usuario un número entre 1 y 25 (validar que es un número y que está entre 1 y 25) y el programa mostrará las veces que está este número en los 200 y en qué posiciones aparece.
2. Realizar un programa que pide al usuario la **temperatura** de los 7 días de la semana y después muestra la temperatura media de la semana y los días en los que la temperatura ha estado por encima de la media y los días que ha estado por debajo de ella.
El resultado debe ser similar al siguiente ejemplo:

```

Introduce la temperatura del lunes: 24
Introduce la temperatura del martes: 31
Introduce la temperatura del miércoles: 26
Introduce la temperatura del jueves: 29
Introduce la temperatura del viernes: 34
Introduce la temperatura del sábado: 39
Introduce la temperatura del domingo: 42
*****
La temperatura media es 32.142857
*****
La temperatura del lunes (24.0) está por debajo de la media
La temperatura del martes (31.0) está por debajo de la media
La temperatura del miércoles (26.0) está por debajo de la media
La temperatura del jueves (29.0) está por debajo de la media
La temperatura del viernes (34.0) está por encima de la media
La temperatura del sábado (39.0) está por encima de la media
La temperatura del domingo (42.0) está por encima de la media
*****
  
```

- Una pila sigue una política LIFO (Last In First Out), es decir, el último en entrar es el primero en salir.

Creo una clase llamada **PilaLIFOMia**.java que tenga las siguientes características:

- La pila siempre se creará vacía.
 - Se podrán insertar elementos.
 - Se podrán sacar elementos.
 - Se podrá preguntar si la pila está vacía.
 - Se podrá preguntar si la pila está llena.
 - Se podrá saber el número de elementos actual de la pila.
 - Implemente un método imprimir que muestre los elementos de la pila.
- Programe un método, que utilizando un objeto de la clase Pila, y que reciba un array de enteros, almacene en la pila los enteros del array. Imprima la pila por pantalla para comprobar.
 - Continuando con el ejercicio anterior, una vez almacenados los elementos, sáquelos y almacénalos en otro array y recorra este array para imprimirlo por pantalla.
 - Si ahora insertamos de nuevo los elementos en la pila ¿quedan en el mismo orden que inicialmente? ¿Por qué razón?

Matrices

Una matriz o tabla es un array de dos dimensiones. Para declarar una tabla utilizaremos:

```
Tipo_de_dato [ ] [ ] nombre_tabla;
```

Una vez declarado, debemos reservar el espacio con new. Por ejemplo:

```
int [ ][ ] numeros;

numeros = new int[4][5];
```

Hemos asignado a la tabla 20 posiciones para almacenar enteros organizado en 4 filas y 5 columnas.

Para asignar un valor o para acceder al valor almacenado en una posición de la tabla será necesario indicar dos índices, uno para la fila y otro para la columna. Por ejemplo:

```
numeros[2][3] = 5;
```

Hemos colocado el valor 5 en la posición 2, 3 (recuerda que tanto las filas como las columnas comienzan en el índice 0). Podemos conocer el número de filas con el método `length`, por ejemplo `numeros.length` y también podemos conocer el número de columnas con el mismo método de la siguiente forma `numeros[0].length`

Ejercicios:

1. Realizar un programa en Java que rellene una matriz de 3 x 4 por filas colocando números consecutivos en cada posición de la tabla. Una vez rellena la tabla la tiene que mostrar en pantalla por filas (con forma de matriz).
2. Realizar un programa que pida al usuario las filas (mínimo 2 y máximo 10) y las columnas (mínimo 2 y máximo 10) que tendrá una tabla de valores enteros (suponemos que el usuario tecleará correctamente los datos). Después se rellena esa tabla con números aleatorios del 1 al 50 y por último la mostrará en pantalla. Para leer el número de filas y de columnas utilizaremos la clase Scanner. La salida será similar a (recuerda que son números aleatorios, tus valores serán diferentes):

Introduce Nº de filas (2-10)

7

Introduce Nº de columnas (2-10)

6

50	18	20	39	50	1
42	20	11	33	3	39
27	28	39	42	23	34
46	13	28	29	32	4
7	36	47	45	36	41
2	45	33	34	13	49
27	41	5	36	7	7

3. Realizar un programa en Java que rellene dos matrices de números enteros de 3 x 4 con números aleatorios entre 1 y 30. Después calculará la matriz suma (también de 3 x 4) que será la suma elemento a elemento de las dos anteriores y por último mostrará las tres matrices según la salida adjunta. La salida será similar a (recuerda que son números aleatorios y los valores pueden cambiar):

MATRIZ A

12	9	29	1
14	24	20	1
22	6	25	23

+ MATRIZ B

29	20	18	3
27	13	12	14
30	28	21	16

= MATRIZ C

41	29	47	4
41	37	32	15
52	34	46	39

4. Realizar un programa en Java que rellene dos matrices de números enteros de 3 x 4 con números aleatorios entre 1 y 30. Después calculará la matriz suma (también de 3 x4) que será la suma elemento a elemento de las dos anteriores y por último mostrará las tres matrices según la salida adjunta.

La salida será similar a (recuerda que son números aleatorios y los valores pueden cambiar):

	MATRIZ A					MATRIZ B					MATRIZ C			
30	25	14	12		14	12	13	16		44	37	27	28	
1	12	12	17	+	15	25	16	9	=	16	37	28	26	
27	24	28	3		13	13	17	22		40	37	45	25	

5. Muestre por pantalla una simulación de un tablero de ajedrez. Las casillas blancas las representaremos con el carácter B y las negras con el carácter N. Utilice arrays de dos dimensiones.

La salida será similar al siguiente ejemplo:

```

B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B

```