

## Arrays y Métodos. Ejercicios 4

### ARRAYS UNIIMENSIONALES

Escribe una clase de nombre **PruebaMetodos1** con los métodos estáticos que se proponen a continuación. Incluir en el método main() de la clase las sentencias necesarias para comprobar el correcto funcionamiento de dichos métodos.

1. Escribe un método, de nombre mostrarArrayPantalla2, que reciba por parámetro un array de enteros y muestre sus valores por pantalla separados por comas.
2. Escribe un método, de nombre obtenerArrayComoString, que reciba un array de enteros por parámetro y devuelva una cadena de caracteres con su contenido.
3. Escribe un método, de nombre completarArray3, que reciba un array de enteros por parámetro y lo rellene de forma que contenga tantos números pares, a partir del cero, como permita su capacidad.
4. Escribe un método, de nombre obtenerSumaArray, que reciba por parámetro un array de enteros y devuelva la suma de sus elementos.
5. Escribe un método, de nombre arrayPotencias2, que cree un array y lo rellene con potencias de 2. Las potencias de 2 comenzarán en 2<sup>0</sup> y el número total de ellas se recibirá por parámetro. El método devolverá el array creado.
6. Escribe un método que reciba como parámetro un array de cadenas y devuelva la cadena resultante de concatenar todas las contenidas en el array.
7. Escribe un método, de nombre obtenerSumaLongCadArray, que reciba por parámetro un array de cadenas y devuelva el número total de caracteres de todas las cadenas del array.
8. Escribe un método, de nombre obtenerLongCadenas, que reciba por parámetro un array de cadenas y devuelva un array de enteros con los tamaños de las cadenas contenidas en el array.
9. Escribe un método, de nombre obtenerArrCad5Vocales, que reciba por parámetro un array de cadenas y devuelva un array con las que contengan las 5 vocales. Para la consideración de un carácter como vocal no se tendrá en cuenta si está en mayúsculas o minúsculas.

## ARRAYS MULTIDIMENSIONALES

Escribe una clase de nombre **PruebaMetodos2** con los métodos estáticos que se proponen a continuación. Incluir en el método main() de la clase las sentencias necesarias para comprobar el correcto funcionamiento de dichos métodos.

10. Escribe un método, de nombre `rellenarMatrizSecuencia2D`, que reciba una matriz de enteros por parámetro y la rellene para que sus posiciones almacenen un valor que se irá incrementando en una unidad por columnas. La matriz se rellenará de manera que dos elementos consecutivos según la primera dimensión almacenen dos valores también consecutivos. Una matriz de 5 elementos en la primera dimensión y 5 en la segunda quedaría como sigue:

0	5	10	15	20
1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24

11. Escribe un método, de nombre `mostrarMatrizID`, que reciba por parámetro un array bidimensional (matriz) de enteros y muestre sus elementos por pantalla de forma que la primera dimensión de la matriz se corresponda con las filas y la segunda con las columnas.
12. Escribe un método, de nombre `matrizIntComoString`, que reciba por parámetro un array bidimensional (matriz) de enteros y devuelva una cadena con la representación textual de la matriz recibida. La cadena dispondrá los elementos de la matriz de forma que la primera dimensión se corresponda con las filas y la segunda con las columnas.
13. Escribe un método, de nombre `obtenerSumaElementosMatriz`, que reciba por parámetro un array bidimensional de números enteros y devuelva la suma de todos sus elementos.
14. Escribe un método, de nombre `rellenarMatrizAsteriscos`, que reciba por parámetro un valor entero que especificará el número de filas de asteriscos que albergará la matriz. La primera fila contendrá un solo asterisco situado en la posición central según la segunda dimensión de la matriz. Cada nueva fila contendrá dos asteriscos más y también se encontrarán centrados según la segunda dimensión de la matriz. El aspecto final que debe presentar la matriz si se recibe un número de filas de asteriscos sería el siguiente:

				*				
			*	*	*			
		*	*	*	*	*		
	*	*	*	*	*	*	*	
*	*	*	*	*	*	*	*	*

15. Escribe un método, de nombre `obtenerLaterales`, que reciba por parámetro una matriz de valores enteros y devuelva una matriz con los valores de los cuatro laterales -superior, izquierdo, derecho e inferior-, de la matriz recibida

## API DE MANEJO DE ARRAYS

16. Escribe una clase, de nombre `APIManejoArrays1`, cuyo método `main()` implemente un programa que lleve a cabo las siguientes acciones:

- a) Declarar y construir un array de 30 enteros, de nombre `arrInt1`. Hacer que el array almacene en las 10 primeras posiciones 10 ceros, en las 10 siguientes 10 unos y en las 10 últimas de nuevo 10 ceros. Mostrar el contenido del array.
- b) Declarar y construir un array de 6 caracteres, de nombre `arrChar`. El array almacenará en la primera mitad de las posiciones el carácter `a` y en segunda mitad el carácter `b`. Mostrar el contenido del array.
- c) Declarar y construir un array de 10 enteros, de nombre `arrInt2`. Hacer que el array almacene en sus posiciones los dígitos del 9 al 0 para mostrar después su contenido. Ordenar el array ascendentemente y mostrar de nuevo su contenido.

17. Escribe una clase, de nombre `APIManejoArrays1`, cuyo método `main()` implemente un programa que lleve a cabo las siguientes acciones:

- a) Declarar y construir un array de cadenas de objetos `String` de nombre `arrStr` que contenga las siguientes cadenas: `"impresora"`, `"peto"`, `"mar"`, `"orilla"` y `"Orihuela"`
- b) Ordenar el array `arrStr` alfabéticamente teniendo en cuenta que las letras mayúsculas figuran en el código Unicode antes que las minúsculas. Mostrar por pantalla el resultado de la ordenación.
- c) Ordenar el array `arrStr` alfabéticamente sin distinguir entre letras mayúsculas y minúsculas. Mostrar por pantalla el resultado de la ordenación.

```
Arrays.sort( arrStr , String.CASE_INSENSITIVE_ORDER );
```