

Hoja de ejercicios 55_57:

- **Atributos y métodos estáticos.**

55.- Crea una clase llamada *Empleado*:

- Con los atributos privados *nombre*, y *teléfono*, de tipo *String*.
- El constructor necesitará el *nombre* y el *teléfono*.
- Implementa los métodos necesarios para acceder a estos dos atributos de manera pública (*getters* y *setters*).
- Añade el atributo estático y privado *numeroEmpleados*. Este atributo almacenará el número de instancias que se han creado de la clase *Empleado*.
- Implementa un método público para conocer el valor de *numeroEmpleados*. Llámalo *getNumeroEmpleados*.
- Crea una clase aparte con el método *main*:
 - Crea un *array* de 4 *empleados*. Los datos de cada *empleado* será leídos por consola.
 - Muestra los datos de cada *empleado* del *array*.
 - Llama al método *getNumeroEmpleados* para mostrar el número de *empleados* instanciados.

56.- Crea una clase llamada *MiString*, sin atributos y con los siguientes métodos públicos y estáticos:

- *String alReves(String)*: Devuelve la cadena pasada como parámetro pero al revés.
- *String limpiaCaracteres(String, String)*: Devuelve la cadena pasada como primer parámetro pero eliminando los caracteres pasados en la cadena como segundo parámetro.
- *String sustituye(String, char, char)*: Devuelve la cadena pasada como parámetro pero sustituyendo todas las apariciones del primer *char* que pudiera haber por el segundo.
- *boolean todosIguales(String)*: Devuelve *true* o *false*, dependiendo de si todos los caracteres de la cadena son iguales o no.
- *String quitaEspacios(String)*: Devuelve una cadena con los espacios en blanco que pudiera tener eliminados.
- *boolean esNumeroEntero(String)*: Devuelve *true* o *false*, dependiendo de si todos los caracteres pasados como parámetro son numéricos.

Crea una clase aparte con el método *main* para probar todas las funcionalidades de la clase *MiString*.

57.- Crea una clase llamada *Número*, sin atributos y con los siguientes métodos públicos y estáticos:

- *boolean esPrimo(int)*
- *boolean esPerfecto(int)*
- *boolean esCapicua(int)*
- *boolean sonAmigos(int, int)*
- *int[] extraeDivisores(int)*: Devuelve un *array* con los divisores del número pasado como parámetro, sin incluir el propio número.
- *void duplica(int[])*: Duplica el contenido de cada uno de los números de un *array* de enteros pasado como parámetro.
- *void ordena(int[] ,int)*: Ordena el contenido de un *array* de enteros pasado como primer parámetro. El segundo parámetro indica el tipo de orden, siendo los valores posibles constantes de tipo *int* llamadas ASCENDENTE y DESCENDENTE.
- *boolean eliminaPrimos(int[])*: Elimina los posibles números primos que pudiera tener el *array*, y lo redimensiona a su nuevo tamaño si fuera necesario. Devuelve *true* si había algún número primo, y *false* en caso contrario.

Crea una clase aparte con el método *main* para probar todas las funcionalidades de la clase *Numero*.