

# PRACTICA 1

## Clarificación de conceptos de POO

### Arrays, ordenación e interfaces

---

Realizar un programa en Java que implemente distintos algoritmos de ordenación, tanto iterativos como recursivos, sobre arrays. Con este propósito se diseñará una clase *ClaseVector* que incorpora dentro un array de objetos de tipo *ObjetoOrdenado*, así como todos los métodos necesarios para su manipulación. El tamaño del este array se determinará en tiempo de ejecución y en un determinado momento puede no estar completamente lleno.

La clase *Persona* debe almacenar toda la información necesaria sobre un alumno, formada por nombre, apellidos y nota. Se proporcionarán los métodos de acceso y modificación correspondientes, así como un método que simplifique su impresión directa por pantalla. Además esta clase debe implementar el interfaz *ObjetoOrdenado*, teniendo en cuenta que dos alumnos son diferentes si tanto sus nombres como sus apellidos son diferentes, y que el criterio de ordenación será el alfabético según apellidos y teniendo en cuenta el nombre si estos son iguales.

```
/** interfaz que declara el tipo de objetos ordenados */
public interface ObjetoOrdenado {

    /** devuelve <0, >0 o igual a 0 segun el objeto sea menor, mayor o igual que el
     * argumento */
    public int compareTo(ObjetoOrdenado obj);
}
```

En *ClaseVector* se deben incluir los siguientes constructores y métodos públicos:

- *public ClaseVector(int tamaño)* - Construye un vector vacío de tamaño máximo "tamaño".
- *public ClaseVector()* - Construye un vector vacío de tamaño máximo TAMMAX (constante definida en la propia clase).
- *public anadirElemento(ObjetoOrdenado elemento)* – Si hay espacio disponible añade el elemento en la siguiente posición libre del array. En caso contrario muestra un error por pantalla.
- *public ObjetoOrdenado accederElemento(int posicion)* – Si *posicion* es correcta devuelve el elemento situado en dicha posición del array. En caso contrario muestra un mensaje de error por pantalla.
- *public void mostrar()* - Muestra por pantalla el array contenido en *ClaseVector*.
- *public void ordenarPorInsercion()*– Ordena el array utilizando el método de inserción directa
- *public void ordenarRapidamente()*– Ordena el array utilizando el método *quicksort*
- *public ClaseVector devuelveVector()* - Devuelve una  **copia**  del objeto *ClaseVector*

Finalmente, el funcionamiento de estas clases se ejemplificará mediante una aplicación interactiva, sencilla de manejar mediante menús, que permita el mantenimiento de una lista de clase. Todas las lecturas se realizarán utilizando el paquete *esi.jar* que se ha proporcionado en el laboratorio.

Cada clase debe incluir en su método principal código que pruebe la correcta implementación de los métodos que incluye.

#### **Parte opcional**

Mejorar las opciones proporcionadas en el menú de interacción.

Incluir otros métodos de ordenación (p.e. selección directa, ordenación por mezclas *-mergesort*).

Incluir un método de búsqueda de un elemento en la lista utilizando búsqueda binaria.

#### **Memoria de la práctica**

El programa debe estar escrito de forma clara y legible. Con este propósito, se deben utilizar identificadores significativos e incluir comentarios que faciliten la comprensión. Se entregará disquete con el código correctamente comentado y formateado, así como la documentación generada con  *javadoc* .

Cada grupo entregará la memoria exclusivamente en el laboratorio y en el turno correspondiente. La fecha de entrega de la práctica es dos semanas después de la primera sesión.