

Ejercicios de colecciones

Índice

1 Implementaciones e interfaces (1 punto).....	2
2 Uso de listas (1 punto).....	2

1. Implementaciones e interfaces (1 punto)

Tenemos una clase como la siguiente para encapsular los datos de una película, en la que se almacena, entre otros campos, la lista de actores:

```
public class PeliculaTO {
    String titulo;
    ArrayList<String> actores;
    ArrayList<String> directores;

    public PeliculaTO() {
        actores = new ArrayList<String>();
        directores = new ArrayList<String>();
    }

    public ArrayList<String> getActores() {
        return actores;
    }

    public void addActor(String actor) {
        actores.add(actor);
    }
}
```

Como segunda opción, tenemos una implementación alternativa como la siguiente:

```
public class PeliculaTO {
    String titulo;
    List<String> actores;
    List<String> directores;

    public PeliculaTO() {
        actores = new ArrayList<String>();
        directores = new ArrayList<String>();
    }

    public List<String> getActores() {
        return actores;
    }

    public void addActor(String actor) {
        actores.add(actor);
    }
}
```

Imaginemos que más adelante comprobamos que se hacen un gran número de operaciones de borrado o inserción en mitad de la lista, y decidimos que sería más eficiente utilizar un `LinkedList`. Si nuestra clase pertenece a una librería que se está utilizando desde múltiples puntos de una gran aplicación, ¿qué cambios implicaría el pasar a utilizar una lista enlazada en cada una de las dos versiones? ¿Cuál de ellas consideras por lo tanto más apropiada?

2. Uso de listas (1 punto)

Vamos a añadir a nuestro proyecto de gestión de filmotecas de la sesión anterior un nuevo

DAO que manejará datos en memoria, en lugar de guardarlos en fichero o base de datos. A este DAO le llamaremos `MemoryPelículaDAO`, y utilizará internamente colecciones para almacenar las películas. Debemos poder añadir películas, eliminarlas, o ver la lista de todas las películas que tengamos. Se pide:

- a) Las operaciones más comunes que haremos en la aplicación van a consistir en añadir una película al final de la lista, eliminar una película dado su identificador (habrá que buscarla en la lista), u obtener el listado de todas las películas y recorrerlo entero para mostrarlo. ¿Qué tipo de colección consideras más apropiada para almacenar esta información?.
- b) Añadir el código necesario a las operaciones para agregar películas y consultar la lista de películas disponibles. Comprobar que la aplicación funciona correctamente.
- c) Consideraremos que dos películas son la misma si su identificador coincide. Añadir el código necesario a la clase `PelículaTO` para que esto sea así. Comprobar que funciona correctamente implementando el método para eliminar películas (si al método `remove` de la colección se le pasa como parámetro un objeto `PelículaTO` con el mismo identificador que una de las películas ya existentes en dicha colección, deberá eliminarla de la lista).
- d) Al obtener la lista de películas almacenadas, mostrarlas ordenadas alfabéticamente. Utilizar para ello los algoritmos que se nos proporcionan en el marco de colecciones.

Nota

Si internamente estamos almacenando las películas en un tipo de colección sin información de orden, para ordenarlas tendríamos que volcar esta colección a otro tipo de colección que si que sea ordenable. Es más, aunque internamente se almacenen en una colección con orden, siempre es recomendable volcar los elementos a otra colección antes de devolverla, para evitar que se pueda manipular directamente desde cualquier lugar del código la estructura interna en la que guardamos la información. Además las películas deberán ser *comparables*, para que los algoritmos sepan en qué orden se deben ordenar.

- e) Utilizar otro tipo de colección para almacenar las películas en memoria. Si se ha hecho un diseño correcto, esto no debería implicar más que el cambio de una línea de código.
- f) Para cada película, en lugar de almacenar únicamente el nombre de los actores, nos interesa almacenar el nombre del actor y el nombre del personaje al que representa en la película. Utiliza el tipo de datos que consideres más adecuado para almacenar esta información.

