

Taller de Herencia

Proyecto 1

- A. Escribe una clase `Multimedia` para almacenar información de los objetos de tipo multimedia (películas, discos, mp3, mp4...). Esta clase contiene título, autor, formato, y duración como atributos. El formato puede ser uno de los siguientes: wav, mp3, midi, avi, mov, mpg, cdAudio y dvd. El valor de todos los atributos se pasa por parámetro en el momento de crear el objeto. Esta clase tiene además, un método para devolver cada uno de los atributos y un método `toString()` que devuelve la información del objeto. Por último, un método `equals()` que recibe un objeto de tipo `Multimedia` y devuelve `true` en caso de que el título y el autor sean iguales y `false` en caso contrario.
- B. Escribe una clase `Película` que herede de la clase `Multimedia` anterior. La clase `Película` tiene, además de los atributos heredados, un actor principal y una actriz principal. Se permite que uno de los dos sea nulo, pero no los dos. La clase debe tener dos métodos para obtener los nuevos atributos y debe sobrescribir el método `toString()` para reflejar la nueva información.
- C. Escribe una clase `ListaMultimedia` para almacenar objetos de tipo multimedia. La clase debe tener un atributo, que sea un array de objetos `Multimedia` y un entero para contar cuantos objetos hay almacenados. Además, tiene un constructor y los siguientes métodos:
1. **el constructor** recibe un entero por parámetro indicando el número máximo de objetos que va a almacenar.
 2. **`int size()`**: devuelve el número de objetos que hay en la lista.
 3. **`boolean add(Multimedia m)`**: añade el objeto al final de la lista, y devuelve *true*, en caso de que la lista esté llena, devuelve *false*.
 4. **`Multimedia get(int position)`**: devuelve el objeto situado en la posición especificada.
 5. **`String toString()`**: devuelve la información de los objetos que están en la lista.

D. Escribe una aplicación dónde:

- Se cree un objeto de tipo ListaMultimedia de tamaño 10.
- Se creen tres películas y se añadan a la lista.
- Se muestre la lista por pantalla.

E. Escribe una clase Disco, que herede de la clase Multimedia ya realizada. La clase Disco tiene, aparte de los elementos heredados, un atributo para almacenar el género al que pertenece (rock, pop, ópera...). La clase debe tener un método para obtener el nuevo atributo y debe sobrescribir el método toString() para que devuelva toda la información.

F. Escribe una aplicación donde:

1. Se cree un objeto de tipo ListaMultimedia de tamaño máximo 20.
2. Se creen tres discos y se añadan a la lista.
3. Se muestre la lista por pantalla.
4. Se cree un objeto de tipo Disco, introduciendo el título y el autor de uno de los elementos de la lista. Para el resto de los argumentos se utilizan valores no significativos.

Proyecto 2

A. Escribe una interfaz, llamada *ColeccionInterfaz*, que declare los siguientes métodos:

1. **estaVacía()**: devuelve true si la colección está vacía y false en caso contrario.
2. **extraer()**: devuelve y elimina el primer elemento de la colección.
3. **primero()**: devuelve el primer elemento de la colección.
4. **añadir()**: añade un objeto por el extremo que corresponda, y devuelve true si se ha añadido y false en caso contrario.

B. A continuación, escribe una clase Pila, que implemente esta interfaz, utilizando para ello un array de *Object* y un contador de objetos.

C. Escribe una clase, de nombre PruebaPila, en la que se implementen dos métodos:

1. **rellenar()**: recibe por parámetro un objeto de tipo *ColeccionInterfaz*, y añade los números del 1 al 10.
2. **imprimirYVaciar()**: recibe por parámetro un objeto de tipo *ColeccionInterfaz* y va extrayendo e imprimiendo los datos de la colección hasta que se quede vacía.
3. Crea el objeto de tipo Pila y probar.

D. Escribe una clase Cola que implemente la interfaz *ColeccionInterfaz*, usando un objeto de la clase *LinkedList*.

Proyecto 3

A. Escribe un programa para una biblioteca que contenga libros y revistas.

- Las características comunes que se almacenan tanto para las revistas como para los libros son el código, el título, y el año de publicación. Estas tres características se pasan por parámetro en el momento de crear los objetos.
- Los libros tienen además un atributo prestado. Los libros, cuando se crean, no están prestados.
- Las revistas tienen un número. En el momento de crear. En el momento de crear las revistas se pasa el número por parámetro.
- Tanto las revistas como los libros deben tener (aparte de los constructores) un método toString() que devuelve el valor de todos los atributos en una cadena de caracteres. También tienen un método que devuelve el año de publicación, y otro el código.
- Para prevenir posibles cambios en el programa **se tiene que implementar una interfaz Prestable con los métodos prestar(), devolver() y prestado.** La clase Libro implementa esta interfaz

B. Escribe una aplicación en la que se implementen dos métodos:

1. **cuentaPrestados():** recibe por parámetro un array de objetos, y devuelve cuántos de ellos están prestados.
2. **publicacionesAnterioresA():** recibe por parámetro un array de Publicaciones y un año, y devuelve cuántas publicaciones tienen fecha anterior al año recibido por parámetro.
3. **En el método main(),** crear un array de Publicaciones, con 2 libros y 2 revistas, prestar uno de los libros, mostrar por pantalla los datos almacenados en el array y mostrar por pantalla cuántas hay prestadas y cuantas hay anteriores a 1990.