



APELLIDOS, Nombre

TITULACIÓN Y GRUPO

MÁQUINA

NOTAS PARA LA REALIZACIÓN DEL EJERCICIO:

- El ejercicio se almacenará en el directorio **Mis Documentos\POO**. En caso de que no exista deberá crearse, y si ya existiese, deberá borrarse todo su contenido antes de comenzar.
- **Cada fichero deberá identificarse con el nombre del alumno, titulación, grupo y número de equipo.**
- La evaluación tendrá en cuenta la claridad de los algoritmos, del código y la correcta elección de las estructuras de datos, así como los criterios de diseño que favorezcan la reutilización.
- **Está permitido:**
 - Consultar el API.
- **No está permitido:**
 - Utilizar otra documentación electrónica o impresa.
 - Intercambiar documentación con otros compañeros.
 - Utilizar soportes de almacenamiento.
- Una vez terminado el ejercicio subir un fichero comprimido (.jar, .zip o .rar) sólo con los **fuentes** que hayáis realizado a la tarea creada en el campus virtual para ello.

Se desea crear una aplicación que permita gestionar el préstamo de películas de un videoclub. En concreto la clase `Videoclub` permitirá hacer distintos tipos de consultas y operaciones de préstamo y devolución de películas, representadas dentro de nuestra aplicación por una clase `Pelicula`. La clase `VideoclubPorGeneros` nos permitirá especializar nuestro videoclub, de forma que sea posible asociar a cada género cinematográfico una colección de películas que representen sus títulos más significativos.

(1,5 puntos) **1.** Defínase la clase `Pelicula`, cuyas instancias dispondrán de un título (de tipo `String`), un conjunto de actores (de tipo `Set<String>`), un código (de tipo `String`) y un año de estreno (de tipo `int`). Se pide:

a) (0.5 ptos.) Un constructor que inicialice apropiadamente las instancias de la clase y métodos de consulta para las distintas variables de instancia. Implementar los métodos de consulta `título`, `actores`, `codigo` y `año`. Además, obtener una representación textual de los objetos de la forma: [título , actores , codigo , año]

b) (0.5 ptos.) Redefínanse los métodos necesarios para que dos películas sean iguales cuando lo sean sus títulos (sin distinguir mayúsculas y minúsculas) y el año de estreno.

c) (0.5 ptos.) La manera natural de comparar dos películas es tal que sea menor la que tenga el título anterior en orden lexicográfico (sin distinguir mayúsculas y minúsculas) y a igualdad de título, la de menor año de estreno.

(5 puntos) **2.** Defínase la clase `Videoclub`. Un videoclub dispone de un catálogo con todas las películas del videoclub representado como dos aplicaciones: una que asocia a cada película el número de copias disponibles de la misma (el número de copias nunca será negativo, aunque puede ser cero si todas las copias están prestadas), y otra que mantiene contadores con las películas prestadas (si no hay copias prestadas de una película determinada esta no aparecerá en la aplicación; por supuesto, un contador nunca podrá ser negativo). Se pide:

a) (1.5 ptos.) Un constructor que tome como argumento el nombre de un fichero de texto con el que se inicializará el videoclub. Se supondrá que cada una de las líneas de dicho fichero tiene el formato:

título de la película%actores%código%año de estreno%número de copias

donde supondremos que ninguno de los campos contiene el carácter separador '%'. Los campos '*año de estreno*' y '*número de copias*' son números, mientras que '*actores*' es una lista de actores separados por comas. Véase el fichero `peliculas.txt` para familiarizarse con el formato. Si hubiera películas repetidas se incrementará el número de copias. Cualquier error en el formato del fichero deberá contemplarse con el lanzamiento de una excepción no comprobada `VideoclubException`, incluyendo un mensaje descriptivo del error.

b) (0.5 ptos.) Defínase el método `void añadir(Pelicula p, int copias)` para añadir un número de copias de una película `p` dada al videoclub.

c) (0.5 ptos.) Defínase el método `int disponibles(Pelicula p)` para comprobar cuántas copias hay disponibles de la película `p`.

d) (0.5 ptos.) El método `Pelicula pelicula(String titulo)` devuelve la película del videoclub con el título dado, o `null` si no encuentra una película con dicho título.

e) (1 pto.) El método `Set<Película> peliculasDeActor(String actor)` devuelve las películas en las que el actor dado ha intervenido (solo o en colaboración con otros actores). Devuelve un conjunto vacío si no encuentra ninguna.

f) (1 pto.) Implementar los métodos `void prestar(Pelicula p)` y `void devolver(Pelicula p)` que, respectivamente, simulen el préstamo y la devolución de películas del videoclub. En particular, `prestar` debe decrementar el número de copias disponibles e incrementar el número de copias prestadas de la película en cuestión, y `devolver` decrementará el número de copias prestadas (eliminando la entrada correspondiente si queda a 0) e incrementará el número de copias disponibles de la película especificada. Si no es posible llevar a cabo alguna de las operaciones se lanzará una excepción `VideoclubException`.

(3,5 puntos) **3.** La clase `VideoclubPorGeneros` representa videoclubs que además de la funcionalidad descrita en el apartado anterior para cualquier videoclub, permiten gestionar las películas del videoclub por géneros cinematográficos. Para ello, se dispone de una estructura que permite asociar a cada género cinematográfico un conjunto de películas. Se pide:

a) (2 pto.) Un constructor que tome como argumento los nombres de dos ficheros de texto con los que se inicializará el videoclub, uno con películas como cualquier videoclub, y otro con las películas organizadas por géneros cinematográficos. Véase el fichero `generos.txt` como muestra del formato de este tipo de fichero. El fichero `generos.txt` tiene en cada línea el nombre de una película precedido por su género y separado por ":" (sin espacios). Para añadir una película a la estructura que asocia a los géneros, debe usarse el método del apartado b). De nuevo, cualquier error en el formato del fichero deberá contemplarse con el lanzamiento de una excepción no comprobada `VideoclubException`, incluyendo un mensaje descriptivo del error

b) (1 pto.) Defínase el método `void añadir(String g, String t)` para añadir la película de título `t` al género cinematográfico `g`. Si la película no está en el videoclub (ni disponible ni prestada) se lanzará una excepción `VideoclubException`.

c) (0.5 ptos.) Defínase el método `Set<Película> genero(String g)` para consultar las películas que están asociadas en el videoclub al género cinematográfico `g`.

La siguiente clase principal muestra el funcionamiento de las clases anteriormente descritas. Consúltense los ficheros de prueba de texto proporcionados.

```
import java.util.*;
import java.io.IOException;
public class TestVideoclub {
    public static void main(String[] args){
        try {
            Videoclub b = new Videoclub("peliculas.txt");
            System.out.println("Sobre el titulo:Sentido y sensibilidad");
            Pelicula p = b.pelicula("Sentido y sensibilidad");
            System.out.println(p);
            while (b.disponibles(p) > 0) {
                b.prestar(p);
                System.out.println("Sentido y sensibilidad prestada");
            }
            System.out.println("Sentido y sensibilidad no disponible");
            Pelicula p1 = b.pelicula("Sentido y sensibilidad");
            b.devolver(p1);
            System.out.println("Sentido y sensibilidad devuelta");
            for (Pelicula ppp : b.peliculasDeActor("Julie Andrews")) {
                System.out.println(ppp);
                if (b.disponibles(ppp) > 0) {
                    b.prestar(ppp);
                    System.out.println(ppp.titulo()+" prestada");
                }
            }
            // salida esperada:
            /*Sobre el titulo :Sentido y sensibilidad
            [Sentido y sensibilidad, [Kate Winslet, Emma Thompson, Hugh Grant],
            DR-0098, 1995]
            Sentido y sensibilidad prestada
            Sentido y sensibilidad prestada
            Sentido y sensibilidad prestada
            Sentido y sensibilidad no disponible
            Sentido y sensibilidad devuelta
            [Mary Poppins, [Dyck Van Dyke, Julie Andrews], Inf-0034, 1964]
            Mary Poppins prestada
            [Sonrisas y lagrimas, [Christopher Plummer, Julie Andrews], INF-
            0887, 1965]
            Sonrisas y lagrimas prestada*/
            VideoclubPorGeneros v
                = new VideoclubPorGeneros("peliculas.txt", "generos.txt");
            Set<Pelicula> gen = v.genero("Suspense");
            if (gen == null) {
                System.out.println("No constan peliculas de ese genero");
            } else {
                System.out.println("Suspense: " + gen);
            }
            //salida esperada
            /*Suspense: [[Tesis, [Ana Torrent, Fele Martínez, Eduardo Noriega],
            SUS-0033, 1996],
            [Con la muerte en los talones, [Eva Marie Saint, Cary Grant], SUS-
            0055, 1959],
            [Los otros, [Nicole Kidman, Fionnula Flanagan], SUS-0021, 2001]]*/
        } catch (IOException ioe) {
            System.err.println("Error de E/S: " + ioe.getMessage());
        } catch (VideoclubException ee) {
            System.err.println(ee.getMessage());
        }
    }
}
```