Práctica 9: Colecciones

El objetivo del presente ejercicio es el de desarrollar una aplicación en Java que permita gestionar el funcionamiento de un videoclub, donde tendremos una serie de clientes y una serie de DVDs que estos podrán tomar prestados.

1. Créese una clase **DVD** que represente copias de películas en DVD, con un código (que será un String), el título de la película, el nombre del director (suponemos que una película tiene un único director) y una lista de actores. Defínanse los siguientes constructores y métodos:

   a) Un constructor con un código, un título, un director y una lista de actores como argumento.

   b) Métodos para consultar cada una de las variables de instancia de un objeto de esta clase.

   c) La clase DVD debe permitir la comparación entre objetos de la misma. Las copias se compararán atendiendo al valor numérico de sus códigos. Se debe hacer dos métodos de comparación uno que será método de instancia u objeto y otro que será método de clase. Para ello, redefiniremos, el método *equals* que proporciona Java.

   d) Un método de consulta que muestre la información sobre una película. Por ejemplo, un DVD con código 0005, título "Como Dios", dirigida por Shadyac e interpretada por Carrey, Freeman y Aniston se representará como:

      0005: Como Dios (dirigida por Shadyac e interpretada por [Carrey, Freeman, Aniston])

   e) La clase DVD debe permitir la comparación entre objetos de la misma, así como la inserción de sus instancias dentro de colecciones ordenadas. Las copias se compararán atendiendo al valor numérico de sus códigos. ******

      ****** Para ello debe implementar la interfaz ***Comparable*** que pertenece al paquete *java.lang* y redefinir el único método de esta clase que es ***compareTo*****(***Object ob***)** que devuelve un número negativo, igual a cero o positivo según el objeto *ob* sea mayor, igual o menor que el actual (argumento implícito), respectivamente.

      Este método es necesario implementarlo, ya que lo necesita el método *add* de la clase *TreeSet* (o cualquier otra que necesite mantener un orden entre sus componentes) puesto que necesita comprobar que el nuevo objeto no exista previamente en la colección y además lo ordene según el criterio elegido, aunque esta comprobación será ajena a nosotros. La clase *TreeSet* la usaremos posteriormente en la clase DVDClub para almacenar la lista de películas.

      Podéis comprobar fácilmente la necesidad de este método cuando probéis la clase DVDClub, para ello anular con comentarios el método *compareTo* de la clase DVD e intentar insertar más de un DVD en la lista de películas, veréis que el primero lo inserta sin problemas, pero en cuanto llega al segundo dará un error, ya que no podrá compararlo con el existente.

2. Constrúyase una clase **Cliente** que represente a un cliente con su nombre y el conjunto de películas alquiladas por dicho cliente en un momento determinado. Las películas que tiene prestadas un cliente se almacenarán en una Lista que no permita elementos repetidos (por ejemplo un ***HashSet***). Defínanse los siguientes constructores y métodos:

   a) Un constructor que tome el nombre del cliente como argumento.

   b) Métodos apropiados para devolver el nombre y las películas alquiladas por un cliente.

   c) Un método `boolean alquila(DVD p)` que añada la película p al conjunto de películas alquiladas por el cliente que reciba el mensaje.

   d) Un método `DVD devuelve(String t)` que extraiga la película con título t del conjunto de películas alquiladas por el cliente que recibe el mensaje. Dicha película será devuelta como resultado del método; si la película no estuviera alquilada por dicho cliente debe devolverse `null`.

   e) Añade o más bien redefine el método `equals()`. Dos clientes son iguales si tienen el mismo nombre (sin distinguir mayúsculas y minúsculas).

3. Un club de DVD's vendrá representado por una clase **DVDClub**, la cual mantendrá una lista de los clientes del club y una lista de las películas disponibles. Esta clase deberá incluir:

   a) Un constructor.

   b) Dos métodos para añadir Clientes y Películas al club.

   b) Un método `DVD sacarPelícula(String título)` que dado un título devuelva un objeto de tipo DVD con la película, o bien un null en caso de no estar disponibles esa película. Este método además de devolver la película la extrae de la lista de películas del club.

   c) Un método `Cliente cliente(String nombre)` que devuelva el objeto cliente del club cuyo nombre coincide con la cadena de caracteres dada como argumento. Devolverá `null` en caso de no existir un cliente con dicho nombre.

   d) Un método `void alquila(String titulo, String nombre)` que tenga el efecto esperado de un alquiler (la película con ese título deja de estar disponible y pasa a estar entre las películas alquiladas por el cliente con el nombre indicado).

   e) Un método `void devuelve(String titulo, String nombre)` que tenga el efecto esperado de una devolución de un DVD alquilado previamente.

   f) Método `void disponibles()` muestra un listado con todas las películas disponibles en el club.

   g) Método `void alquiladas(String nombre)` muestra un listado con todas las películas alquiladas por un determinado cliente.

Supongamos ahora que se desea dejar a un lado el espíritu altruista del videoclub y se decide crear un negocio a partir de él, un videoclub donde los clientes tengan un saldo disponible y por cada alquiler de DVD se cobre una cantidad determinada de dinero.

4. La clase **DVDClubRentable** tendrá toda la funcionalidad de Videoclub pero cobrará una cantidad determinada por cada alquiler, que supondremos una cantidad constante (3€, independientemente del tiempo que dure el préstamo). Un cliente no podrá alquilar un DVD si no dispone del saldo suficiente. Los clientes de este tipo de clubs se crearán con un saldo inicial (10€) y podrán incrementarlo en cualquier momento con un método recargar. Se pide realizar, **SOLO** la clase **ClienteDinero** relacionada con el DVDClubRentable, por lo que no se completará el ejercicio del club rentable.

5. Probar el funcionamiento del ejercicio, creando objetos de las distintas clases (DVD's, Clientes, DVDClub, etc) así como probando los distintos métodos existentes en las clases.