



Tarea Sesión 4.

1 Preliminares

Esta tarea deberá ser entregada **24 horas antes** de la siguiente clase de laboratorio.

Antes de la entrega, renombra los proyectos dentro de Eclipse (usando Refactor-> Rename) con:

`apellido_1_apellido2_nombre_lab04_task_dome`

`apellido1_apellido2_nombre_lab4_task_post` en minúsculas y sin tildes.

Exporta los proyectos juntos a un fichero comprimido y **añade la checkList** de la tarea. Súbelo al Campus Virtual.

Añade a los proyectos el diagrama de clases UML ampliado con la funcionalidad requerida en esta tarea.

2 Objetivo

Se trata de completar los proyectos DOME y Social Network con algunas operaciones más que implican el uso del polimorfismo.

3 Ampliación del proyecto DOME

Realiza una copia del proyecto dome de la sesión 4 y modifica su nombre a `apellido1_apellido2_lab04_task_dome`. Añade las siguientes operaciones.

3.1 Redefine equals en todas las subclases

Si no lo has hecho en clase, redefine el método `equals` en todas las subclases de forma que sea utilizado en `search` para comprobar que el ítem que recibe como argumento es “igual” a alguno de los que hay en DOME. Recuerda que:

- Dos Cds son iguales, si lo son el título y autor
- Dos Dvds son iguales si lo son el título y el director
- Dos VideoGames son iguales si lo son el título y la plataforma

3.2 Añade un precio a cada ítem

Añade un precio base (`basePrice`, de tipo `double`) a cada ítem. Haz que se reciba como último parámetro con el resto de los datos, en el constructor. Asegúrate de que el precio no supere los 1000€ y no sea negativo. En caso contrario se lanzará `IllegalArgumentException`.



3.3 Añade `double getTotalPrice()` a `MediaLibrary`.

Método que devuelva un valor de tipo `double` con la suma de los precios finales de todos los ítems que tenga la librería. **Comienza realizando las pruebas.** A continuación, implementa el código.

El precio final depende del tipo de ítem y habrá que calcularlo en cada ocasión. Se debe **calcular** de la siguiente forma.

- En caso de los Cds, se les añade al precio base un impuesto TAX de 2€.
- Los dvds tienen el precio base como precio final
- Los videoGames tienen un precio incrementado del 10% del precio base.

3.4 Genera un código para los items

Añade un método a la clase `MediaLibrary` llamado `generateCodes()` que devuelve una cadena con el código de todos los ítems de la mediateca, separados por un guion. El código será una cadena formada por las tres primeras letras del título y un número secuencial, comenzando en 0. Por ejemplo, si el título del primer elemento es “Yesterday, y el del segundo es “All you need is Love”, el código que vuelca es “Yes0-All1”. Si no hay ítems, devuelve “”.

Comienza por implementar las pruebas. Posteriormente implementa el código.

Pista: busca información sobre el método `substring` de la clase `String` en la documentación de Oracle (Oracle help centre).

Añade código a la clase `MediaPlayer` para que imprima también el código generado.

```
-----Código generado por los Items:  
The0-Alc1-Gam2
```

Comprueba la funcionalidad con JUnit.

Finalmente, si por alguna razón no has hecho aún alguna actividad obligatoria de tareas anteriores que afecten al proyecto DOME (añadir `VideoGame`, etc.) complétalo y envíalo.

4 Ampliación del proyecto red social

Importa el proyecto `post` de la tarea de `lab03` y cambia el nombre del proyecto y de sus paquetes para reflejar la nueva sesión de laboratorio:

Nombre proyecto: `apellido1_apellido2_nombre_lab04_task_post`
Nombre paquete: `uo.mp.lab04.post.model` y otros

4.1 Añade el método `toHtmlFormat()` a la clase `SocialNetwork`

Este método permite formatear los posts y devuelve una lista de cadenas con todos los post formateados en html, teniendo en cuenta que:



- a. En caso de que el post sea un mensaje el formato html que devuelve será:
`<p> mensaje </p>`.
- b. En caso de que sea una foto devolverá:
` caption `

Comienza por implementar los test.

4.2 **Añade** código al método `simulateClient()` de `NetworkApp`

Para que, además, muestre por pantalla toda la lista que se genera con los post de la red en formato html, de forma que salga uno en cada línea.

Ejemplo:

```
Post en formato Html
<img src = foto1>Marina</img>
<p> Me voy al cine</p>
<img src = foto2>Montaña</img>
<p> Estoy en Africa</p>
```