Laboratorio L05. Herencia

Objetivos

- Poner en práctica conceptos avanzados de PMOO en Java: herencia y polimorfismo.

Herramientas a utilizar

- *Eclipse* y *Javadoc*

Entrega

Fichero **.zip** con todo lo solicitado en el laboratorio L05: exportación del proyecto eclipse (código y documentación).

Nomenclatura de los diferentes contenidos:

- Nombre del proyecto Java: *apellidoNombre_L05*
- Nombre de archivo a entregar: apellidoNombre.zip
- La entrega debe ser individual y a través de eGela.

Contexto del Laboratorio

Queremos realizar una aplicación que gestione los músicos y actuaciones de una orquesta. El objetivo inicial del desarrollo es contratar a músicos, cada uno de los cuales actuará interpretando su fragmento de una pieza, y calcular el coste total de las contrataciones. Para ello, crearemos la clase `Orchestra` dentro del paquete `packOrchestra`. En esta clase, se definirán los atributos y operaciones adecuados para alcanzar los objetivos perseguidos, tal y como se describe más adelante.

Los integrantes de la orquesta serán músicos. La clase `Musician` y las clases que se deriven de ella se definirán en el paquete `packMusician`. Aunque los músicos compartan muchas características, también tienen muchas diferencias. La clase `Musician` reunirá las características que tienen en común todos los músicos y, a partir de ella, se derivan otras clases, tales como `Director` o `Instrumentalist`.

Antes de empezar a realizar las tareas de este laboratorio, lee detenidamente todo el enunciado para que te ayude en todo el proceso de diseño e implementación.

Descripción de las clases de la aplicación

A continuación, se describen todas las clases que utilizará la aplicación:

C1. La clase `Musician` representa un músico en general. Un músico se caracteriza por su nombre, salario (500 euros por defecto) y si está o no contratado por alguna orquesta. Además, debe proporcionar los siguientes métodos:

a. Una constructora con el *nombre* como parámetro
b. Los métodos `get` y `set` de todos sus atributos
c. `hire`: método que realiza la contratación de un músico. Si el músico no estaba contratado, asignará el valor **true** a su atributo *contratado* y el método finalizará devolviendo **true**. Si el músico ya estaba contratado no modificará ningún atributo del músico y finalizará devolviendo el valor **false**
d. `perform`: método que devuelve un String que representa la actuación del músico. Por defecto el método perform hará que el músico cante "sssss", es decir, que esté en silencio.

e. `equals` es necesario sobrescribir el método `equals` por defecto.

f. `toString` también hay que sobreescribir este método. Debe mostrar cuál es la clase[1] del músico, junto con la información de todos las características (atributos) de la clase. Este método se sobreescribirá de forma incremental en todas las clases que sea necesario.

C2. `Director`: (Es un tipo de `Musician`) las personas que dirigen la orquesta incluyen también la cantidad de años que llevan dirigiéndola. El salario de los directores será 300 euros superior al salario por defecto, más un cinco por ciento por cada año de antigüedad. Debe incluir los métodos `get` y `set` de su *antigüedad*. Su actuación consistirá en: "Tok tok tok: (silence)".

C3. `Singer`: clase que representa los cantantes (es un tipo de `Musician`). De manera general, los cantantes cobrarán 200 euros más que el sueldo por defecto de los músicos. Sin embargo, se distinguen `Sopranos` y `Tenores`. Los `Sopranos` cobrarán un 33% más si son de talla **internacional**[2]. Los `Tenores` cantan "La-la-la-laaaaaaa" mientras que los `Sopranos` cantan "Li-li-li-liiiiiiiii".

C4. `Instrumentalist`: con una clase de `Musician` que representa a todos los músicos que tocan algún instrumento en la orquesta. Por lo tanto, esta clase debe añadir información sobre el instrumento que toca. Tendrá también el métido `get` adecuado para obtener qué instrumento toca.

C5. `Pianist`: (es un `Instrumentalist`) clase que representa a los pianistas. Su sueldo es de 1500 euros y al interpretar produce el sonido "Ding-ting-dang-ding-tang-ting".

C6. `Trumpeter`: (es un `Instrumentalist`) representa a los trompetistas. Su sueldo es de 100 euros más que el de por defecto de los músicos y al interpretar produce el sonido "Tu-ru-ru-tu-ru-ru".

C7. `Orchestra`: representa la orquesta y se caracteriza por un nombre y un `ArrayList` de músicos. Esta clase, además, debe proporcionar las siguientes funcionalidades:

a. `getName`: devuelve un String con el nombre de la orquesta.

b. `hire`: dado un `Musician`, se realiza el proceso de su contratación en la orquesta. Si el músico no está ya contratado, se contratará y se añadirá a la lista de músicos de la orquesta; además, debe crear un String indicando cuánto cobrará el músico. Si el músico ya estaba contratado con anterioridad, devolverá el mensaje **"`The musician was already hired`"**.

c. `act`: escribe un mensaje indicando el nombre de la orquesta que actúa; a continuación, hace que actúe el `Director` y posteriormente hace que cada componente de la orquesta realice su interpretación invocando el método polimórfico `perform` de la clase `Musician`.

d. `getFee`: Calcula el precio o presupuesto para una representación de la orquesta, calculando la suma de los sueldos de todos los componentes de la orquesta.

e. `toString`: construye y devuelve un String con la información de todos los miembros de la orquesta

[1] Utiliza la siguiente expresión para lograrlo: `this.getClass().getSimpleName()`

[2] Podéis añadir un atributo en Soprano que os permita saber si es o no de talla internacional. No es necesario crear una clase para los sopranos internacionales y otra para los que no lo son.

Tareas a realizar en el laboratorio

Tarea 1. Realiza el **diagrama de clases UML** (a mano, en papel o como te resulte más cómodo) e implementa las clases de la aplicación teniendo en cuenta las especificaciones que aparecen en el enunciado. Recuerda:

1.1. La clase `Orchestra` debe definirse dentro del paquete `packOrchestra`.

1.2. La clase `Musician` y todas las clases que deriven de ella deben definirse en el paquete `packMusician`.

Tarea 2. Una vez realizadas las tareas anteriores, crea una clase `DemoOrchestra` que contenga el método **`main`** del proyecto. Define esta clase dentro del paquete `packDemo`. Su método **`main`** debe realizar las siguientes operaciones:

2.1. Crear al menos 7 personas que sean músicos, de los cuales: uno será director o directora, otro pianista, dos sopranos (uno de talla internacional y otro no), un tenor y dos trompetistas.

2.2. Crea la orquesta y contrata a las personas músicas que has creado. Intenta contratar a dos personas en dos ocasiones.

2.3. Muestra en pantalla la información de todos los miembros de la orquesta.

2.4. Haz que la orquesta dé una función.

2.5. Imprime en pantalla el presupuesto de la orquesta.

A continuación, puedes ver un ejemplo de ejecución:

```
Monica Segovia pianist, hired for 1500.0 euros.
Chris Black tenor, hired for 700.0 euros.
Edurne Berasaluze soprano, hired for 700.0 euros.
Ines Barrutieta trumpeter, hired for 600.0 euros.
Monica Segovia pianist is alredy hired.
Chris Black tenor is alredy hired.
Rafael Zurbano director, hired for 1400.0 euros.
Jon Kaperotxipi trumpeter, hired for 600.0 euros.
Caroline Linecarol soprano, hired for 931.0 euros.

These are the Jazzban Orkestra participants
Director name=Rafael Zurbano, salary=1400.0, hired=true , antiquity=15
Pianist name=Monica Segovia, salary=1500.0, hired=true ,instrument=Piano
Tenor name=Chris Black, salary=700.0, hired=true
Soprano name=Edurne Berasaluze, salary=700.0, hired=true ,international=false
Trumpeter name=Ines Barrutieta, salary=600.0, hired=true ,instrument=Trumpet
Trumpeter name=Jon Kaperotxipi, salary=600.0, hired=true ,instrument=Trumpet
Soprano name=Caroline Linecarol, salary=931.0, hired=true ,international=true

This is the Jazzban Orkestra performance:
  [Rafael Zurbano]: Tok tok tok: (silence)
  [Monica Segovia, Piano]: Ding-ting-dang-ding-tang-ting
  [Chris Black]: La-la-la-laaaaaa
  [Edurne Berasaluze]: Li-li-li-liiiiii
  [Ines Barrutieta, Trumpet]: Tu-ru-ru-tu-ru-ru
  [Jon Kaperotxipi, Trumpet]: Tu-ru-ru-tu-ru-ru
  [Caroline Linecarol]: Li-li-li-liiiiii

Spent onJazzban Orkestra participants' salaries: 6431.0 euro.
```

Tarea 3. Todo el código debe documentarse correctamente utilizando JavaDoc.