

Laboratorio S12

Objetivos

- Trabajar los conceptos de herencia y polimorfismo en Java

Herramientas a utilizar

- *Eclipse, JUnit y Javadoc*

Entregable

Hay que entregar en un fichero la exportación del proyecto de Eclipse (código + documentación) que contenga todo lo que se solicita en este laboratorio. Nomenclatura para los diferentes elementos.

- Nombre del proyecto Java: *apellido_nombre_S12*
- Nombre del fichero a entregar: *apellido_nombre.zip*
- La entrega es individual y se entregará a través de eGela
- El último día para realizar la entrega es el 23 de abril antes de las 23:55

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, actuar interpretando cada componente su fragmento de una pieza, y conocer 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 marcados tal y como se describirá más adelante,

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

Antes de empezar a realizar las tareas lee detenidamente todo el enunciado del laboratorio 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 de las que constará la aplicación.

C1 La clase `Musician` representa a los músicos en general. Los músicos se caracterizan por su nombre, salario (500 euros por defecto) y si están o no contratados en alguna orquesta. Además, debe proporcionar los siguientes métodos.

- a. Una constructora que reciba por parámetro el nombre
- b. Getters y setters para los atributos
- c. `hire`: método que simula la contratación de un músico. Si el músico no está contratado, el atributo `contratado` del músico pasará a ser `true`. Si el músico ya está contratado entonces se elevará la excepción `AlreadyHiredMusician`. Esta excepción se declarará en el paquete `packmusician`.

- d. `perform`: método que devuelve un `String` que representa la actuación del músico. Por defecto el método `perform` hará que el músico cante “ssssss” es decir, se oirá el silencio.
 - e. Se debe sobrescribir el método `equals`.
 - f. Se debe sobrescribir el método `toString` que mostrará la clase¹ a la que pertenece el músico actual junto con la información de todos los atributos de la clase. Este método se debe sobrescribir de manera incremental en todas las subclases en las que sea necesario.
- C2 Director:** Para las personas que dirigen la orquesta queremos guardar también la cantidad de años que lleva dirigiéndola. El salario de los directores será de 300 euros superior al salario por defecto más un cinco por ciento por cada año de antigüedad. Deberá incluir el `getter` y `setter` de su antigüedad. Su actuación consistirá en: “Tok tok tok: (silence)”
- C3 Singer:** Esta clase representará a los cantantes (son un tipo de `Musician`). De manera general, los cantantes cobrarán 200 euros más que el sueldo por defecto de los músicos. Sin embargo, vamos a distinguir sopranos y tenores. Los Sopranos cobrarán un 33% más si son de talla internacional². Los tenores cantan “La-la-la-laaaaaaa” mientras que todos los sopranos cantan “Li-li-li-liiiiiiii”.
- C4 Instrumentalist:** Esta clase, hija de `Musician`, representa a todos los músicos que tocan algún instrumento en la orquesta. Por lo tanto, se debe añadir información sobre el instrumento que toca. Se añadirá también el `getter` adecuado para saber cuál es el instrumento que toca.
- C5 Pianist:** Esta clase representa a los pianistas. Su sueldo es de 1500 euros y al interpretar produce el sonido “Ding-ting-dang-ding-tang-ting”.
- C6 Trumpeter:** Esta clase 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:** Esta clase representa la orquesta y se caracteriza por un nombre y un `ArrayList` de músicos. Esta clase debe además proporcionar la siguiente funcionalidad:
- a. `getName`: Devuelve un `String` con el nombre de la orquesta
 - b. `hire`: Dado un músico, 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 obtener un `String` indicando cuánto cobrará el músico. Si el músico ya estaba contratado con anterioridad, se tratará la excepción de tipo `AlreadyHiredMusician` mostrando un mensaje adecuado en pantalla.
 - 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 utilizando 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`: Obtiene un `String` con la información de todos los componentes de la orquesta

¹ `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. Implementar las clases de la aplicación, considerando las especificaciones dadas anteriormente. Recordad:

- La clase Orchestra se debe definir dentro del paquete packorchestra
- La clase Musician y las clases que se deriven de la misma se definirán en el paquete packmusician

Tarea 2. Una vez realizadas las tareas anteriores, crea una clase DemoOrchestra en el paquete packdemo. En el método main de esta clase se debe realizar lo siguiente:

- Crear al menos 7 personas músicas incluyendo: director o directora, pianista, dos sopranos (una de talla internacional y otra no), tenor, dos trompetistas.
- Crea la orquesta y contrata a los y las músicas que has creado. A dos hay que intentar contratarlos dos veces.
- Muestra por pantalla la información de todos los componentes de la orquesta
- Haz que actúe la orquesta.
- Muestra por pantalla el presupuesto de la orquesta

A continuación, se muestra una posible 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 already hired.
Chris Black tenor is already 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 on Jazzban Orkestra participants' salaries: 6431.0 euro.
```

Tarea 3. Se debe documentar todo el código de manera adecuada utilizando Javadoc y se debe generar la documentación

Tareas optativas

Tarea 4. Verifica con JUnit los siguientes métodos

- Clase Musician: métodos perform y hire
- Clase Orchestra: método getFee