

Pareja: Ignacio Rabuñal y Victoria Pelayo

GENERO Y DIA

Hemos creado dos enumeraciones.

La primera es género, para los distintos géneros que puede tener una película.

La segunda es día, que son los distintos días de la semana, más uno de carácter general para los festivos. Esta enumeración la hemos creado para la entrada del día del espectador que tendrá cierto descuento el día del espectador que decida el cine y otro para los días festivos.

PELÍCULA

Hemos creado una clase película. La película tiene un director, un género, un título, una sinopsis y un año.

En esta clase hemos metido los correspondientes “geters” y “seters” para ver y modificar sus atributos.

SESIÓN

La clase sesión consta de los siguientes atributos: fecha, el número de butacas disponibles y vendidas y una película.

En esta clase como métodos tenemos los correspondientes “geters” y “seters” para fecha y película.

En cuando a las butacas, como decisión en el diseño, hemos creado las funciones addButaca() y delButaca() que añade y elimina, respectivamente, una butaca disponible. Las butacas que ya estén vendidas no se pueden modificar, es decir, una vez que vendes una butaca no se puede “devolver”.

También tenemos la función venderButaca() que se encarga de vender una butaca, es decir, quitar una de las butacas disponibles y añadirla en vendidas.

SALA

Las salas tendrán tres atributos, un id, como identificador de la sala, un número de butacas y una lista de sesiones que se proyectan en esa sala.

Como particular, esta clase tiene el id, que no es modificable.

El número de butacas de la sala solo se puede ampliar, ya que estamos haciendo un cine y no vamos a eliminar butacas de una sala que ya tenía, se ha decidido que solo se puedan añadir.

Se podrán añadir y eliminar sesiones. Cuando se añade una sesión a una sala, desde un control de errores externo, la sesión está “nueva”, es decir, no hay ninguna butaca vendida. Se

establece que la sesión tenga el mismo número de butacas disponibles que butacas tiene la sala.

Se proponía una modificación de diseño dónde las sesiones tendrían una Sala, en nuestro caso hemos decidido que la Sala pueda acceder a sus sesiones pero las sesiones no a la sala, porque era innecesario. Hemos podido realizar las funcionalidades necesarias para la administración del cine sin necesidad de ello. Esto ha sido una decisión del diseño personal.

ENTRADA

Las entradas tendrá un único atributo el precio. Como funciones tendrán los correspondientes “getters” y “setters”.

ENTRADA ESPECTADOR

Es una entrada especial, ya que tendrá un descuento asociado, es la que utilizaremos cuando una persona compre una entrada el día del espectador, por ejemplo.

Esta clase es una clase hija de entrada, y constará de otro atributo que es el de descuento.

La función getPrecio() en este caso devolverá el precio calculado con el descuento correspondiente.

CINE

Constará de dos enteros estáticos, un máximo de películas y otro de salas.

Además, tendrá los siguientes atributos: nombre, dirección, una lista de películas, una lista de salas y una lista de entradas.

Tendrá “getters” de todos sus atributos y “setters” de nombre y dirección.

Tendrá tres métodos que modifican películas:

- addPelícula(p:Película). Que añade una película a la lista de películas

- removePelícula(p:Película): Elimina una película de la lista de películas. Es una función privada ya que nunca se llamará desde fuera

- removePelículaCartelera(p:Película). Que llamará a la función anterior y además eliminará todas las sesiones en las que se proyectara dicha película.

Tendrá dos métodos que modifican salas:

- addSala(s:Sala) y removeSala(s:Sala), que añade o elimina, respectivamente, una sala del cine.

También tendrá un método para añadir una sesión, se debe especificar en que sala se quiere añadir esa sesión, y un método para eliminar una sesión. En este, si hubiera dos sesiones iguales se eliminaría solo la primera que se encuentre cuando se recorra la lista de salas.

Hay un método llamado comprar entrada que recibirá como argumentos el día que es, el precio de la entrada y el título de la película que quiere ver. Esta función devolverá la sesión de la entrada que ha comprado.

El método getBeneficios() devolverá la suma de todos los precios de las entradas, con descuentos aplicados, que el cine ha vendido.

Por último, hay tres métodos: borrarEntradas(), borrarSalas(), borrarPelículas(). Que borran todos los datos de estos atributos. Sirve en el caso de que se quisieran eliminar todos los datos al finalizar una temporada y así poder empezar desde cero, sin necesidad de ir borrando uno a uno.

DESCUENTO PERSONA

Para este apartado hemos creado una nueva clase, EntradaReducida.

Esta clase es una clase hija de EntradaEspectador. Tendrá dos descuentos, uno, por el día de la semana que sea, 0 si no es festivo ni el día del espectador, y otro por la persona, si es que tiene.

El único cambio que hemos tenido que hacer en otras clases es añadir un nuevo parámetro de entrada en la función venderEntrada en Cine. También hemos creado una nueva enumeración Edad.

CARTELERA

Hemos creado una nueva clase, cartelera. Ahora, el cine no tendrá una lista de películas, si no que tendrá una cartelera.

La cartelera está formada por una lista de películas.

La manera de imprimir una cartelera es más “estética” que la manera en como imprimíamos antes las películas de un cine. Esto aporta simplicidad a la hora de ver que películas están.

También aporta simplicidad en el código a la hora de añadir/eliminar una película, ya que las funciones de cine quedan más simples, simplemente se llama a una función de cartelera, en vez de tener que recorrer la lista de películas.

TESTERS

Hemos creado distintos main para probar las clases Película, Sesión, Sala y Cine y su correcto funcionamiento.

Hemos ido comentando los test paso a paso para que se entienda su correcto funcionamiento y las funcionalidades que se están probando en cada momento.

DIAGRAMAS DE CLASES

No se han especificado las funciones en el diagrama de clases, para mejorar la comprensión, ya que todos los métodos son explicados en la memoria anteriormente. Acerca de las clases que no se mencione lo contrario, tendrán getters y setters de sus atributos.

DIAGRAMA DE CLASES 1.

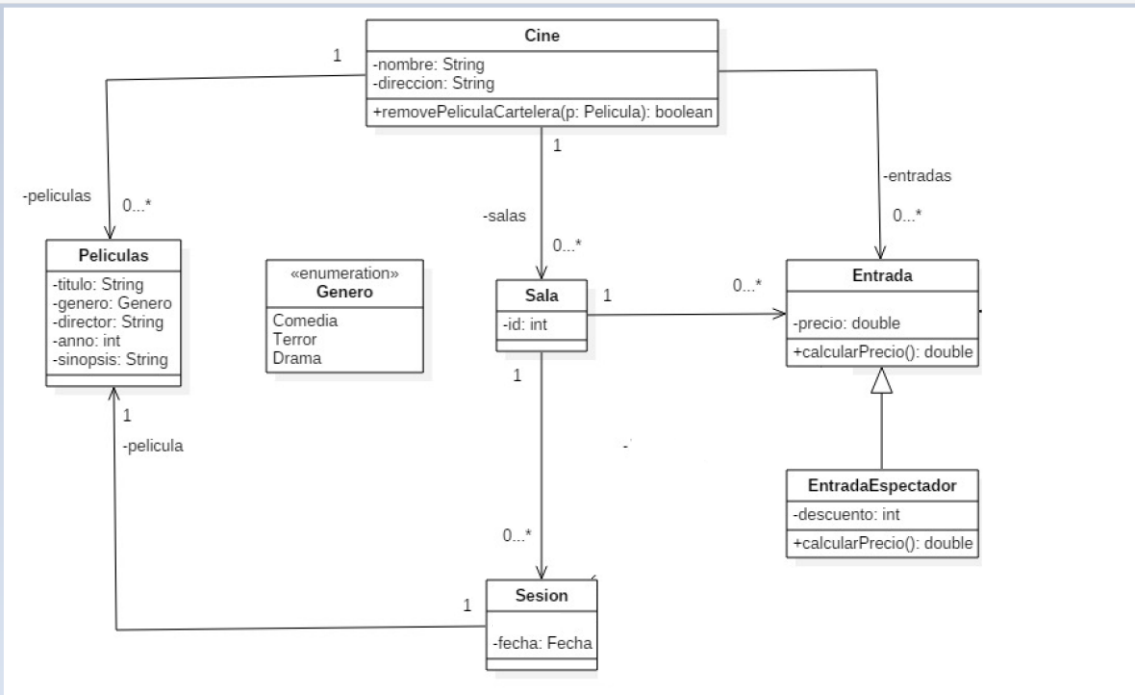


DIAGRAMA DE CLASES(APARTADO OPCIONAL)

