

EXAMEN PARCIAL 2

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reusable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación conceptual de las relaciones entre clases.

Enunciado

Nos han encargado el diseño e implementación de la primera versión de CineORT, una aplicación que permite la compra en línea de entradas de cine.

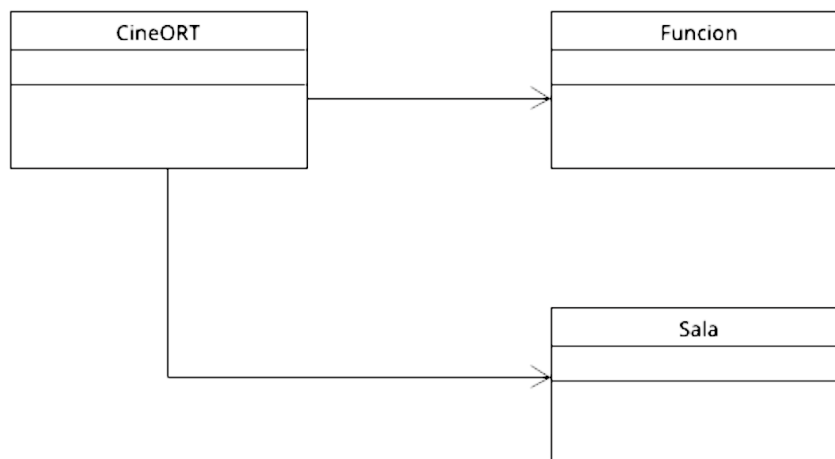
La aplicación mantiene una lista de las funciones en cartelera. Para cada función, se conoce su código identificador (alfanumérico), título, duración (expresada en minutos como un valor entero) y precio por butaca.

Cada función está asociada a una sala, de la cual sólo se conoce un número entero que la identifica y la capacidad total de butacas. El número de sala debe asignarse automáticamente de forma secuencial (1, 2, 3, etc.) en el momento en que la sala es creada. Es importante señalar que diferentes funciones pueden estar asignadas a la misma sala, dado que pueden proyectarse en diferentes días y horarios.

Además, la aplicación mantiene una colección con todos sus clientes registrados. Por el momento, los únicos datos a almacenar de los clientes son el DNI y su género, pudiendo elegir entre FEMENINO, MASCULINO y OTRO.

Cada cliente tiene una colección de entradas, donde cada una está relacionada con una función, y almacena la cantidad de butacas compradas y el precio total. En esta versión inicial, los clientes solo pueden elegir la cantidad de butacas y no las butacas específicas que desean.

Afortunadamente, un compañero de nuestro equipo ha comenzado a diseñar y nos ha dejado una base para que podamos continuar evolucionándola:



Se requiere:

1. Implementar los constructores de **CineORT** y **Cliente**
2. Implementar el método **comprarEntrada(...)** que reciba un DNI, un código identificador de la función y una cantidad de butacas, y realice la compra. Las siguientes posibilidades de retorno deben ser consideradas:
 - **CLIENTE_INEXISTENTE**: el cliente no existe en la colección de clientes registrados.
 - **FUNCION_INEXISTENTE**: la función no existe entre las funciones en cartelera.
 - **FUNCION_SIN_CAPACIDAD**: la cantidad de butacas solicitadas es superior a la disponible.

EXAMEN PARCIAL 2

- OK: la operación se realizó correctamente.
2. Implementar el método **cambiarFuncionDeSala(...)** que, recibiendo un código identificador de la función y un número de sala, cambie la función de la sala actual a la sala especificada. Es importante tener en cuenta que la función podría tener entradas vendidas y la nueva sala debe tener una capacidad mayor o igual a la cantidad de butacas vendidas. Las siguientes posibilidades de retorno deben ser consideradas:
- FUNCION_INEXISTENTE: la función no existe en la lista de funciones en cartelera.
 - SALA_INEXISTENTE: la sala no existe entre las existentes.
 - CAPACIDAD_SALA_INSUFICIENTE: la sala no tiene la capacidad necesaria para alojar la cantidad de butacas vendidas.
 - OK: la operación se realizó correctamente.
3. Implementar el método **obtenerMejoresClientes()** que devuelva una colección con los clientes que, sumando todas sus entradas, tengan el monto total mayor. Puede haber ninguno, uno o varios clientes en la colección resultante.
Ejemplo: Si un cliente compró 3 entradas con un precio total de \$2.000 y en otra oportunidad compró 5 entradas por un precio total de \$8.000, su monto total será de \$10.000.
4. Implementar el método **obtenerMejoresClientes(...)** que reciba un monto total mínimo y devuelva una colección con los clientes que, sumando todas sus entradas, tengan un monto total superior al monto indicado. Puede haber ninguno, uno o varios clientes en la colección resultante.