

Ejercicio 1

Nos piden hacer un programa orientado a objetos sobre un cine (solo de una sala) tiene un conjunto de asientos (8 filas por 9 columnas, por ejemplo). Del cine nos interesa conocer la película que se está reproduciendo y el precio de la entrada en el cine. De las películas nos interesa saber el título, duración, edad mínima y director. Del espectador, nos interesa saber su nombre, edad y el dinero que tiene. Los asientos son etiquetados por una letra (columna) y un número (fila), la fila 1 empieza al final de la matriz como se muestra en la tabla. También deberemos saber si está ocupado o no el asiento.

8	A	8	B	8	C	8	D	8	E	8	F	8	G	8	H	8	I	7	A	7	B	7	C	7	D	7	E	7	F	7	G	7	H	7	I	6	A	6	B	6	C	6	D	6	E	6	F	6	G	6	H	6	I	5	A	5	B	5	C	5	D	5	E	5	F	5	G	5	H	5	I	4	A	4	B	4	C	4	D	4	E	4	F	4	G	4	H	4	I	3	A	3	B	3	C	3	D	3	E	3	F	3	G	3	H	3	I	2	A	2	B	2	C	2	D	2	E	2	F	2	G	2	H	2	I	1	A	1	B	1	C	1	D	1	E	1	F	1	G	1	H	1	I
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Realizaremos una pequeña simulación, en el que generaremos muchos espectadores y los sentaremos aleatoriamente (no podemos donde ya este ocupado). En esta versión sentaremos a los espectadores de uno en uno. Solo se podrá sentar si tienen el suficiente dinero, hay espacio libre y tiene edad para ver la película, en caso de que el asiento este ocupado le buscamos uno libre. Los datos del espectador y la película pueden ser totalmente aleatorios.

Ejercicio 2

Vamos a hacer una baraja de cartas españolas orientado a objetos.

Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos) y un palo (espadas, bastos, oros y copas)

La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Las operaciones que podrá realizar la baraja son:

barajar: cambia de posición todas las cartas aleatoriamente siguienteCarta: devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas. cartasDisponibles: indica el número de cartas que aún puede repartir darCartas: dado un número de cartas que nos pidan, le devolveremos ese número de cartas (piensa que puedes devolver). En caso de que haya menos cartas que las pedidas, no devolveremos nada pero debemos indicárselo al usuario. cartasMonton: mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario mostrarBaraja: muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta. Spoiler Inside SelectShow 11) Estando en un grupo de amigos, se planea hacer una porra de la liga de fútbol. A nosotros se nos ocurre hacer un programa en POO para simular como podría desarrollarse la porra.

Cada jugador de la porra, pone un 1 euro cada jornada y decide el resultado de los partidos acordados.

Si nadie acierta en una jornada los resultados, el bote se acumula.

En principio, deben acertar el resultado de dos partidos para llevarse el dinero del bote de la porra.

Todos empiezan con un dinero inicial que será decidido por el programador (ya sea como parámetro o como constante). Si el jugador no tiene dinero en una jornada no podrá jugar la porra.

Para esta versión, entre jugadores podrán repetir resultados repetidos, por lo que el jugador que primero diga ese resultado (tal como estén de orden) se llevara primero el bote.

Los resultados de la porra serán generados aleatoriamente, tanto los de jugador como los de los partidos (no es necesario nombre, solo resultados).

Al final del programa, se deberá mostrar el dinero que tienen los jugadores y el número de veces que han ganado.

Para este ejercicio, recomiendo usar interfaces (no hablo de interfaces gráficas) para las constantes y métodos que sean necesarios.

Ejercicio 3

Vamos a hacer el juego de la ruleta rusa en Java.

Como muchos sabéis, se trata de un número de jugadores que con un revolver con un sola bala en el tambor se dispara en la cabeza.

Las clases a hacer son:

Revolver: Atributos: posición actual (posición del tambor donde se dispara, puede que esté la bala o no) posición bala (la posición del tambor donde se encuentra la bala) Estas dos posiciones, se generaran aleatoriamente.

Funciones: disparar(): devuelve true si la bala coincide con la posición actual siguienteBala(): cambia a la siguiente posición del tambor toString(): muestra información del revolver (posición actual y donde está la bala) Jugador: Atributos id (representa el número del jugador, empieza en 1) nombre (Empezara con Jugador más su ID, "Jugador 1" por ejemplo) vivo (indica si está vivo o no el jugador) Funciones: disparar(Revolver r): el jugador se apunta y se dispara, si la bala se dispara, el jugador muere. Juego: Atributos: Jugadores (conjunto de Jugadores) Revolver Funciones: finJuego(): cuando un jugador muere, devuelve true ronda(): cada jugador se apunta y se dispara, se informara del estado de la partida (El jugador se dispara, no ha muerto en esa ronda, etc.) El número de jugadores será decidido por el usuario, pero debe ser entre 1 y 6. Si no está en este rango, por defecto será 6.

En cada turno uno de los jugadores, dispara el revólver, si este tiene la bala el jugador muere y el juego termina.

Aunque no lo haya comentado, recuerda usar una clase ejecutable para probarlo.