Problema: simulador de casino

Se desea simular los posibles beneficios de diversas estrategias de juego en un casino. La ruleta francesa es un juego en el que hay una ruleta con 37 números (del 0 al 36). Cada 3000 milisegundos los diversos hilos apuestan para ver si ganan y el croupier saca un número al azar. Todos los hilos empiezan con 1.000 euros y la banca (que controla la ruleta) con 50.000. Cuando los jugadores pierden dinero, la banca incrementa su saldo.

- Se puede jugar a un número concreto. Habrá 4 hilos que eligen números al azar del 1 al 36 (no el 0) y restarán 10 euros de su saldo para apostar a ese ese número. Si sale su número su saldo se incrementa en 360 euros (36 veces lo apostado).
- Se puede jugar a par/impar. Habrá 4 hilos que eligen al azar si apuestan a que saldrá un número par o un número impar. Siempre restan 10 euros para apostar y si ganan incrementan su saldo en 20 euros.
- Se puede jugar a la «martingala». Habrá 4 hilos que eligen números al azar. Elegirán un número y empezarán restando 10 euros de su saldo para apostar a ese número. Si ganan incrementan su saldo en 360 euros. Si pierden jugarán el doble de su apuesta anterior (es decir, 20, luego 40, luego 80, y así sucesivamente)
- La banca acepta todas las apuestas pero nunca paga más dinero del que tiene.
- Si sale el 0, todo el mundo pierde y la banca se queda con todo el dinero.

Clases: Jugador, Banca, Ruleta, Main

Problema: barberos

En una peluquería hay barberos y sillas para los clientes. Sin embargo, en esta peluquería no siempre hay trabajo por lo que los barberos duermen cuando no hay clientes a los que afeitar. Un cliente puede llegar a la barbería y encontrar alguna silla libre, en cuyo caso, el cliente se sienta y esperará que algún barbero le afeite. Puede ocurrir que el cliente llegue y no haya sillas libres, en cuyo caso se marcha. Simular el comportamiento de la barbería mediante un programa Java teniendo en cuenta que:

- Se generan clientes continuamente, algunos encuentran silla y otros no. Los que no consigan silla desaparecen (terminan su ejecución)
- Puede haber más sillas que barberos y al revés (poner constantes para poder cambiar fácilmente entre ejecuciones).
- Se recuerda que no debe haber inanición, es decir ningún cliente debería quedarse en una silla esperando un tiempo infinito.

Clases: Cliente, Barbero, GestorSillas, Main

Problema: Cinta transportadora

En una empresa hay procesos que producen números y procesos que leen esos números. Todos los números se introducen en una cinta transportadora (cola o vector) limitada.

Todo el mundo lee y escribe de/en esa cinta. Cuando un productor quiere poner un número tendrá que comprobar si la cinta está llena. Si está llena, espera un tiempo al azar. Si no está llena pone su número en la última posición libre.

Cuando un lector quiere leer, examina si la cinta está vacía. Si lo está espera un tiempo al azar, y sino coge el número que haya al principio de la cinta y ese número ya no está disponible para el siguiente.

Crear un programa que simule el comportamiento de estos procesos evitando problemas de entrelazado e inanición.

Nota, utilizad una LinkedList para simular el comportamiento de la cinta

Clases: Productor, Consumidor, Cola, Main

Problema: Almacenes

En unos grandes almacenes hay 200 clientes agolpados en la puerta para intentar conseguir un producto del cual solo hay 50 unidades.

Por la puerta solo cabe una persona, pero la paciencia de los clientes es limitada por lo que solo se harán un máximo de 10 intentos para entrar por la puerta. Si después de 10 intentos no ha podido entrar, el cliente desiste y se marcha.

Cuando se consigue entrar por la puerta el cliente puede encontrarse con dos situaciones:

- Quedan productos: el cliente cogerá uno y se marchará.
- No quedan productos: el cliente simplemente se marchará.

Simula el comportamiento e indica los clientes que han obtenido el producto.

Clases: Cliente, Almacén, Puerta, Main

Problema: Simulación bancaria

Un banco necesita comprobar su sistema informático sobre el acceso a cuentas bancarias. Para ello desea hacer un programa de prueba en Java que permita lanzar procesos que ingresen y retiren dinero a la vez y comprobar así si el resultado final es el esperado.

Se parte de una cuenta con 100 euros y se pueden tener procesos que ingresen 100 euros, 50 o 20. También se pueden tener procesos que retiran 100, 50 o 20 euros euros. Se desean tener los siguientes procesos:

- 40 procesos que ingresan 100
- 20 procesos que ingresan 50
- 60 que ingresen 20.

De la misma manera se desean lo siguientes procesos que retiran cantidades.

- 40 procesos que retiran 100
- 20 procesos que retiran 50
- 60 que retiran 20.

Se desea comprobar que tras la ejecución la cuenta tiene exactamente 100 euros, que era la cantidad de la que se disponía al principio. Realizar el programa Java que demuestra dicho hecho.
Clases: Cliente, Cuenta, Main