

## Ejercicios Extra II

**Ejercicio 1.** Se debe programar el juego del MultiPóker, que es un programa que simula la extracción aleatoria de un conjunto de cartas de tamaño N y descubre las combinaciones de cartas que son útiles para ganar el juego. Las reglas básicas son:

- las cartas se numeran del 1 al 10 (sota, caballo y rey se numeran como 8, 9 y 10). No se considera el palo de las cartas.
- En todas las rondas del juego se debe jugar con el mismo número de cartas.
- Las combinaciones posibles son: pareja (2 cartas repetidas), trío (tres cartas repetidas) y escalera (del grupo de cartas al menos 4 deben ser de números consecutivos) . Ej. en el juego de seis cartas {5 2 3 3 8 4} hay una pareja y una escalera (2 al 5).
- Cuando no hay ninguna combinación útil se dice que es una “mala ronda”.

Hacer un programa Java en una única clase utilizando los métodos static que se consideren necesarios para reproducir el siguiente comportamiento para un solo jugador.

1. Pedir al usuario el número de cartas utilizado para las rondas.
2. Pedir al usuario el número de malas rondas que pueden pasar antes de terminar la partida.
3. Para cada ronda:
  - generar el conjunto N de cartas de forma aleatoria
  - imprimir las cartas de la ronda
  - imprimir el número de parejas, de tríos y si hay o no escalera en la ronda. Si no hay nada imprimir “mala ronda”.
  - Terminar la partida si han pasado el número de malas rondas establecido.

**Ejercicio 2.** Programar un sistema hipotético de control de efectivo en bancos y cajeros automáticos que tenga las siguientes características.

- Una cuenta corriente abierta en un banco pertenece a un usuario, tiene un número de cuenta, un número de tarjeta y mantiene el balance de dinero.
- Un usuario tiene un nombre, DNI y una dirección. Además puede tener cuentas bancarias, cómo máximo 5.
- Un cajero automático pertenece a un banco, tiene asociada una dirección y mantiene el sado del efectivo disponible.

Se pide lo siguiente:

1. Diseñar las clases Java necesarias para representar los elementos descritos anteriormente.
2. Programar los constructores completos, los métodos get y set de cada clase.
3. Implementar en la clase más adecuada el método de `sacarDineroCajero` que dada un usuario y una cantidad a retirar, haga lo siguiente.
  - Determine si el usuario tiene una cuenta en el banco al que pertenece el cajero. De no tener se debe avisar al usuario para no continuar el proceso.
  - Determine si la cuenta tiene saldo disponible para la cantidad solicitada. Si no tiene saldo se debe avisar al usuario para no continuar con el proceso.
  - Determine si el cajero tiene efectivo disponible para satisfacer la operación.
  - Reste la cantidad solicitada de la cuenta del usuario y del saldo disponible del cajero.
4. Implementar en la clase más adecuada el método de `reponerCajero` que dado una cantidad de efectivo actualice el saldo disponible del cajero.
5. En una clase nueva, hacer un método `main` que haga lo siguiente.
  - Cree 3 cajeros, cada uno de bancos distintos
  - Cree 2 usuarios. Para el usuario 1 se deben crear 2 cuentas de bancos distintos, y para el usuario 2 una cuenta del banco restante.
  - Simular que un usuario intenta sacar dinero de un cajero que no es de su banco, y luego que saca dinero correctamente.
  - Imprimir el saldo final de los cajeros.