

PRÁCTICA 1 TEMA 6 - BLACKJACK

Vamos a realizar el juego del **BlackJack** (nuestro 21). Se juega con una baraja inglesa de 52 cartas sin comodines y consiste en sumar lo más próximo a 21 sin pasarse. Realizaremos dos jugadores: tú mismo contra el crupier (ordenador). Reglas: (no apuestas)

- El crupier debe pedir carta siempre que sume 16 o menos.
- El crupier se plantará si suma 17 o más.
- Las cartas numéricas suman su valor. Las figuras valen 10 y el As vale 11 o 1 según interese (podéis simplificarlo a que siempre valga 11).
- Gana el que más se acerque a 21 sin pasarse. La mejor jugada es 21 con dos cartas (Blackjack). Esa jugada gana a cualquier otro 21.

Clases a implementar

- Clase **Carta**
 - o Atributo "palo".
 - o Atributo "figura".
 - o Constructor, se le pasa un palo y una figura y los asigna a los atributos.
 - o `__toString()`, devuelve la carta como una cadena.
 - o `getPalo()`, `getFigura()`, `setPalo(unPalo)`, `setFigura(unaFigura)`.
 - o `getValor()`, devuelve el valor de la carta calculado según las reglas.
- Clase **abstracta Baraja**
 - o Atributo "mazo" que será un array de 52 cartas de la clase Carta.
 - o Método abstracto `repartirCarta()`.
 - o Constructor, crea un array vacío para el mazo.
 - o Método `barajar()`, el array del mazo lo desordena.
 - o Método `listar()`, muestra por pantalla el mazo de cartas.
- Clase **BarajaInglesa**, hereda de la anterior.
 - o Atributo estático "palos" array con corazones, diamantes, picas y tréboles.
 - o Atributo estático "figuras" array con As, 2, 3, 4, 5, 6, 7, 8, 9, jota, reina y rey.
 - o Constructor, llama al constructor del padre. Después llama a un método privado llamado `generarMazo()`.
 - o Método `repartirCarta()`, devolverá una carta del mazo, de la parte superior, y la elimina de la baraja.
 - o Método `generarMazo()`. Va a generar las 52 cartas. Se recorre los palos, para cada palo se recorre las figuras, genera una carta nueva con ese palo y figura, y la mete en el array del mazo.
- Clase **Jugador**
 - o Atributo "mano", la mano del jugador, las cartas que haya ido robando.
 - o El constructor creará un array vacío para el atributo mano.
 - o Método "nuevaCarta(\$Carta)", añade una Carta pasada como parámetro a la mano del jugador.
 - o Método "`__toString()`", que muestre la mano del jugador.
 - o Método "`valorMano()`", que calcule el valor de la mano del jugador, la suma del valor de sus cartas.

- Si necesitas más atributos o métodos, puedes añadirlos.
- Clase **Partida**, usa Jugador y BarajaInglesa. Llevará los dos jugadores y la baraja. Debes decidir tú qué implementar de esta clase y cómo.

FUNCIONAMIENTO

El juego podéis organizarlo de varias formas, o van sacando carta una a una, crupier y jugador alternativamente, o primero juega uno sus cartas y luego el otro. Un **posible** funcionamiento sería:

- Crupier: saca cartas de la baraja de la sesión y las mete en la mano del crupier. Comprueba que no tenga BlackJack, en cuyo caso gana. Comprueba que no se haya pasado en cuyo caso pierde. Comprueba que no tenga que plantarse (17 o más). Puede jugar toda su mano automáticamente hasta el final.
- Jugador: saca de la baraja de la sesión cartas, una a una, y las mete en la mano del jugador. Comprueba que no tenga BlackJack, en cuyo caso gana. Comprueba que no se haya pasado en cuyo caso pierde. Con un botón o similar tienes que hacer que vas pidiendo nueva carta.
- Ganador: si ninguno se ha pasado, ni tiene Blackjack, comprobamos si alguno suma 21. Gana el que tenga 21 con menos cartas. Si los dos están plantados gana el que más se acerque a 21, o en caso de empate el que tenga menos cartas.

IMPLEMENTACIÓN

- En la parte de arriba se irán mostrando las cartas del crupier.
- Un poco más abajo, separado, se mostrarán las cartas del jugador.
- Aparecerán una serie de botones:
 - **Nueva partida.** Borra la variable de sesión, “partida”. Crea nueva Partida y la mete en la sesión. **El objeto partida estará en la sesión serializado**, ojo con esto.
 - **Plantarse.** No se reparten más cartas para el jugador.
 - **Pedir carta.** El jugador recibe una nueva carta del mazo de cartas a su mano.
- Cuando el jugador o crupier ganen o pierdan, se mostrará un mensaje con el ganador y no se podrá pulsar el botón “Me planto” ni “Robar Carta”.
- Se valorará muy positivamente que las cartas aparezcan en pantalla como imágenes, así como el diseño general de la aplicación.
- Se valorará la modularidad de la aplicación (nuevas clases si son necesarias, nuevos métodos, etc.).
- Se valorará el uso del patrón MVC
- Se valorará el uso de Ajax o SPA.