

Relación de Ejercicios de Clases y Objetos (1): Carta y Baraja.

Vamos a crear dos clases y un pequeño programita para probarlas.

La primera clase es **Carta**, en la que guardaremos la información relativa a una carta de la baraja española. Para ello, implementaremos los siguientes atributos, métodos y propiedades:

- Atributos (privados):
 - `numero`: un *entero* (1..12) que nos representa el número de la carta, siendo el 10 la sota, el 11 el caballo y el 12 el rey. En el tipo de baraja más normal no existirán los números 8 y 9.
 - `palo`: un *enum* **Palo**, que podrá tener 4 valores distintos: Oros, Copas, Espadas y Bastos.
- Constructores:
 - `Carta(numero, palo)`: nos crea una carta pasándole el palo y el número.
 - `Carta(id)`: a la que le pasamos un número entre 1 y 40 que representa la carta, siendo 1 el as de oros y 40 el rey de bastos. A partir del id calcularemos el número y el palo y almacenaremos estos datos en los atributos.
- Propiedades:
 - `NumeroCarta`: de sólo lectura, que nos devuelve el número.
 - `PaloCarta`: de sólo lectura, que nos devuelve el palo (le cambiamos el nombre porque si no tendríamos 2 Palos, el *enum* y la propiedad).
 - `NombreNumero`: de sólo lectura, que nos devuelve el número de la carta como un string y con letras (1=as, 2=dos, ..., 12=rey).
 - `NombrePalo`: de sólo lectura, que nos devuelve el palo de la carta, en letras (Oros, Copas, etc.).
 - `NombreCarta`: de sólo lectura, nos devuelve el nombre completo de la carta (ej.: as de oros).
 - `ValorTute`: de sólo lectura, nos devuelve el valor de la carta en el juego del tute (1 = 11, 3 = 10, sota = 2, caballo = 3, rey = 4).
 - `ValorMus`: de sólo lectura, nos devuelve el valor de la carta en el juego del mus (1, 2 = 1; 3, sota, caballo y rey = 10, el resto su valor).
 - `Valor7ymedia`: de sólo lectura, nos devuelve el valor de la carta en el juego de las 7 y media (figuras: 0.5, el resto su valor). Tendrá que devolver un decimal para poder devolver el medio punto.
- Métodos:
 - `ToString()`: que devuelve lo mismo que `NombreCarta`.

La otra clase que tendremos que crear será **Baraja**, que la definiremos como sigue:

- Atributos (privados):
 - `listaCartas`: será una lista de cartas.
- Constructores:
 - `Baraja()`: nos crea una baraja vacía.
 - `Baraja(bool barajar)`: nos crea una baraja española de 40 cartas, creando cada carta e insertándolas en la lista. Se recomienda usar el constructor de carta que recibe un *id*. Si el *bool* que le pasamos es *true*, habrá que barajar la baraja después de crearla.
- Métodos:
 - `Barajar()`: mezcla aleatoriamente las cartas de la baraja. Hay varias formas de hacerlo, yo os propongo crear otra lista de cartas; mientras queden cartas en nuestra baraja, elegimos una al azar (con un `Random`), la sacaremos y la meteremos en la siguiente baraja. Al terminar, copiamos todas las cartas a nuestra baraja y listos.
 - `Cortar(int posición)`: corta la baraja. Consiste en pasar tantas cartas como nos digan desde la primera posición a la última.
 - `Robar()`: roba una carta. Nos devuelve un objeto de tipo carta que se corresponde al primer elemento de la lista que eliminaremos de la misma.
 - `InsertaCartaFinal(int idCarta)`: meteremos una carta nueva al final de la baraja correspondiente al *id*.
 - `InsertaCartaPrincipio(int idCarta)`: meteremos una carta nueva al principio de la baraja correspondiente al *id*.
 - `InsertaCartaFinal(Carta c)`: igual que la anterior, pero le pasamos un objeto de tipo carta.
 - `InsertaCartaPrincipio(Carta c)`: igual que la anterior, pero le pasamos un objeto de tipo carta.
- Propiedades:
 - `NumeroCartas`: de sólo lectura, nos dice cuántas cartas quedan en la baraja.
 - `Vacia`: de sólo lectura, será un booleano que nos dice si la baraja está vacía o aún quedan cartas.

Para terminar, haremos un programita para jugar a las 7 y media. Será muy sencillo: creamos una baraja, la barajamos y le vamos ofreciendo al usuario si quiere más cartas o quiere plantarse. Con cada carta que robemos nos irá diciendo nuestra puntuación total o si ya nos hemos pasado.

Como parte adicional para los que acaban prontito, hacer que el ordenador juegue también (lo menos que dan en Inteligencia Artificial: el programa que juega a las 7 y media).