EXAMEN DWEC 29-11-22 ALUMNE:.....

1. Ejercicio examen anterior (3p)

Implementa el siguiente juego:

Cada vez que se empieza el juego, el programa propone cuatro números (del 0 al 9, sin repetirse), que será el código que el jugador debe adivinar en la menor cantidad de intentos posibles. Ej: '1234' (2234 no válido)

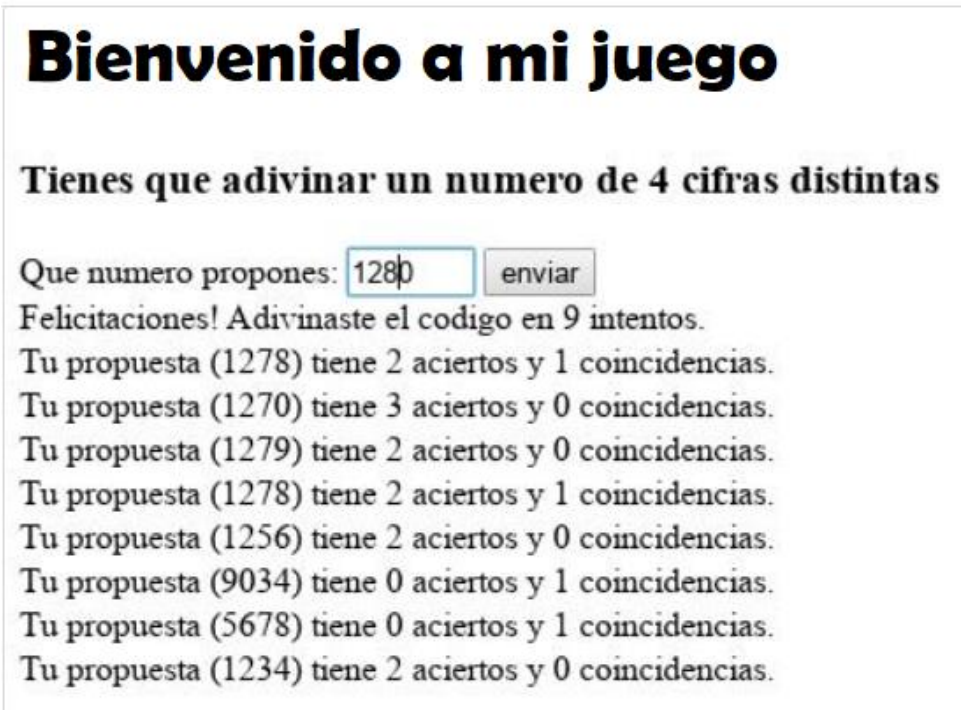
Cada intento consiste en una propuesta de un código posible que escribe el jugador, y una respuesta del programa (se podrá utilizar el método prompt o mediante formulario para recoger las propuestas).

Las respuestas le darán pistas al jugador para que pueda deducir el código. Estas respuestas incluirán:

- El código propuesto.
  - **La cantidad de aciertos**: Es la cantidad de dígitos que propuso el jugador que también están en el código en la misma posición.
  - **La cantidad de coincidencias**: Es la cantidad de dígitos que propuso el jugador que también están en el código, pero en una posición distinta.

El usuario podrá cancelar en juego en cualquier momento y se devolverá un mensaje de que el juego se ha cancelado.

Cuando se acierte, se indicará el número de intentos que han sido necesarios y se preguntará si desea seguir jugando.



Se valorará la claridad y presentación del código del ejercicio, comentarios en el código y su indentación.

EXAMEN DWEC 29-11-22 ALUMNE:.....

2. Juego de cartas (10p)

(1p) Claridad y presentación del código del ejercicio, comentarios en el código y su indentación.

1. (1 p) Crea un tipo de objetos que sirva para representar **Cartas**. Estos objetos tendrán las propiedades:

- **palo**. ("Corazones", "Tréboles", "Diamantes", "Picas")
- **valor**. Un número del 1 al 13.
- Los objetos de este tipo se construyen indicando el palo y el valor.
- Las cartas tendrán el método:
  - **mostrarCarta**. Devolverá en forma de texto entendible el valor de la carta. Por ejemplo: **1 de Picas**

2. (2p) Además, habrá otro tipo de objeto: **Baraja**. La idea es que represente una baraja de cartas . Tendrá los siguientes detallles:

- La baraja la forman 52 cartas. Para ello tendrá la propiedad **cartas** que será un array de 52 cartas.
- Al construir la Baraja se rellenan las cartas en el siguiente orden: por palos y cada palo con las cartas del 1 al 13. No se podrá acceder directamente al array fuera de la función constructora.
- El método **barajar** permite barajar las cartas. Es decir, desordenarlas de forma aleatoria.
- El método **toString** permite obtener la baraja en forma de texto para saber cómo están ordenadas las cartas.
- El método **repartirCarta**, devolviendo una carta y eliminándola de la baraja.

3. (3p) Crea también una clase **Jugador**, con la siguiente funcionalidad:

- El jugador debe tener un **nombre** y **fecha de nacimiento.** Si no es mayor de edad no se le permitirá jugar (La creación del objeto devolverá null)
- El jugador tendrá una **dirección de correo electrónico** que deberá comprobarse sea correcta mediante una expresión regular.
- El jugador debe tener una **mano** (una serie de cartas tomadas de una baraja)
- Deberá tener un método **quienSoy** que devuelva su nombre, edad y dirección de correo electrónico.
- El jugador debe ser capaz de **añadir una carta** a su **mano** (use el método baraja.repartirCarta)
- El jugador debe poder **descartar** una carta de su **mano**, es decir, quitarla de su mano mostrando su valor (usa el método carta.mostrarCarta).

4. (1 p) Añade al tipo de objeto Baraja, un nuevo método que permita resetear la baraja sin cambiar la función constructora. El método **resetearBaraja** dejará la baraja en la situación inicial (52 cartas ordenadas por palo y valor)

5. (2p) Añade el código necesario para comprobar el funcionamiento de las clases creadas y todos sus métodos. Podría ser algo como lo siguiente:

- Crea una baraja y "baraja" las cartas.
- Crea un jugador, muestra quién es y añade cartas a su "mano" mostrando cuáles son (usando el método mostrarCarta). Descarta algunas de sus cartas.
- Muestra la baraja completa en la situación actual.
- Resetea la baraja y muestrala de nuevo