Práctica 3. Interfaces y Menús de interacción con el usuario

En esta práctica se hará uso del concepto de interfaz y se creará un menú de interacción con el usuario que permitirá utilizar objetos de clases diferentes de una forma homogénea mediante un array.

3.1. Primera parte. Interfaces

En esta primera parte del ejercicio se va a implementar una interfaz **Jugable** que implementarán los juegos desarrollados hasta ahora, y los que se desarrollen en un futuro.

Modificación de la Clase JuegoConVidas

- Se eliminará su método abstracto **juega**, pero la clase se seguirá manteniendo como abstracta ya que no interesa que se creen instancias de esta clase.

Creación del Interfaz Jugable

- Dispondrá de un método **juega** con las mismas características que el método juega que se eliminó de la clase JuegoConVidas.
- Se incorporará el método **reiniciaPartida** con la misma cabecera que tiene el método en la clase JuegoConVidas.
- Se incorporará un método **muestraNombre** que no tome ningún parámetro y que obligue a las clases que implementen la interfaz a mostrar un mensaje por pantalla con el nombre del juego.
- Se incorporará un método **muestraInfo** que no tome ningún parámetro y que obligue a las clases que implementen la interfaz a mostrar un mensaje por pantalla con una descripción de cómo jugar al juego.

Modificación de la Clase JuegoAdivinaNumero

- Debe implementar la interfaz Jugable.
- Implementar el método **muestraNombre** que visualizará por pantalla el texto “Adivina un número”.
- Implementar el método **muestraInfo** que visualizará por pantalla una descripción de cómo se juega al juego, informando del número de intentos que se le dan al jugador.

Modificación de la Clase JuegoAdivinaNumeroPar

- Reescribir el método **muestraNombre** para que visualice por pantalla el texto “Adivina un número par”.
- Reescribir el método **muestraInfo** para que visualice la información de este nuevo juego.

Modificación de la Clase JuegoAdivinaNumeroImpar

- Reescribir el método **muestraNombre** para que visualice por pantalla el texto Adivina un número impar
- Reescribir el método **muestraInfo** para que visualice la información de este nuevo juego.

Modificar la clase principal

- El método Jugar se modificará para que reciba un parámetro de tipo Jugable.

Además, se añadirán las invocaciones a los métodos **muestraNombre** y **muestraInfo** antes de la invocación del método **reinciaPartida**.

- Por último, el método **main** creará de nuevo una instancia de cada uno de los tres juegos implementados, pero el tipo de referencia que señalará a dichos objetos será de tipo Jugable. Como en el ejercicio anterior, el número de vidas de cada juego será 5 y como número a adivinar un número cualquiera, otro par y otro impar respectivamente, todos comprendidos entre 0 y 10.

3.2. Segunda parte. Menú de usuario

En esta segunda parte se introduce el concepto de menú de usuario como la clase, o colección de clases, donde el sistema interacciona con el usuario a la hora de intercambiar información. Este diseño persigue la independencia del modelo (conjunto de clases donde se opera con los datos para transformarlos en información) con respecto del interfaz de interacción con el usuario, lo que permitirá mejorar la extensión y mantenimiento de la aplicación.

La clase menú hará uso de una clase, denominada JuegosReunidos, donde se centralizarán todos los recursos del programa (los distintos juegos). En esta clase se hará uso de un array de objetos de tipo Jugable, lo que permitirá acceder a cualquiera de los tres juegos a través de dicha estructura de datos.

Crear la clase Juegos reunidos

La clase JuegosReunidos representa la centralización de todos los recursos del modelo y, por tanto, será el lugar donde se encuentren todos los juegos que se creen y estén disponibles para su selección. Esta clase contendrá como atributo un array de tipo Jugable con tres posiciones donde se almacenará un objeto de cada juego. La inicialización del array se llevará a cabo en el constructor. Los juegos así creados y almacenados tendrán 5 vidas. Además del constructor, se añadirá un método denominado **recuperarJuego** que recibe como parámetro un número entero que representa el índice del array donde se almacena el juego que se quiere seleccionar y que retorna un objeto de tipo Jugable que se corresponde con dicho juego.

Crear la clase Menu

La clase “Menu” representa la entidad que se encarga de gestionar la interacción con el usuario. Esta clase contendrá un atributo de la clase JuegosReunidos de tal forma que le permita acceder a los distintos recursos que centraliza. El constructor de esta clase deberá tener como parámetro una referencia de tipo JuegosReunidos que utilizará para inicializar su atributo.

Además del constructor, esta clase tendrá los siguientes métodos:

- Método **jugar**: es el mismo método que se implementó en la clase principal en el apartado anterior, pero en este caso es un método no estático.
- Método **mostrarOpciones**: Este método muestra por consola el siguiente mensaje:

  ¿A que quieres jugar?.
  Introduce un 1 si quieres jugar a Adivinar un número.
  Introduce un 2 si quieres jugar a Adivinar un número par.
  Introduce un 3 si quieres jugar a Adivinar un número impar.

- Método **chequeoOpcion:** Es un método que tiene como parámetro una variable de tipo entero y que retorna un booleano. Este método retorna true si el número recibido es 1, 2 o 3 y false en caso contrario.
- Método **eligeOpcion:** es un método que permite al usuario llevar a cabo la selección de una de las posibles opciones, validando que la opción introducida se corresponda con las que se muestran. Este método no tiene parámetros y retorna un entero. Para ello, lleva a cabo las siguientes acciones dentro de un bucle: muestra las opciones que tiene el usuario invocando el método **mostrarOpciones**, solicita que el usuario introduzca por teclado la opción que desea en forma de un entero. A continuación, se invoca el método **chequeoOpción** al que se le pasa el número introducido por el usuario. Si el número es correcto no se hace nada, se sale del bucle y se retorna dicho valor. En caso contrario, se muestra por consola un mensaje que indica que la opción no es correcta y que se debe volver a introducir la opción. En este último caso, se deberá controlar la salida del bucle para que se vuelva a entrar y repetir la operación.
- Método **ejecuta**: Este método no tiene parámetros de entrada ni valor de retorno. Su misión es ejecutar el juego elegido y dar la opción al usuario de volver a jugar una vez haya terminado con el juego elegido. Para ello, y dentro de un bucle, se invocará el método **jugar**, al que se le pasará por parámetro el resultado de invocar, a través del atributo de la clase, el método **recuperarJuego**. Para seleccionar el juego concreto se le pasará a este método el resultado de invocar el método **eligeOpción**. Una vez se haya finalizado de jugar con ese objeto de tipo Jugable, se le dará la opción al usuario de elegir si quiere seguir jugando o no. Para ello, se le solicitará que introduzca una “s” en caso afirmativo. Se deberá disponer la salida del bucle de tal forma que si el usuario introduce una “s” se repitan todos los pasos anteriormente descritos. Cualquier otro carácter deberá dar como resultado la salida del bucle y la finalización del método.

Modificación de la clase principal

Por último, se deberá modificar la clase Aplicación para que desde el método principal:

1) Se creen un objeto de tipo JuegosReunidos y un objeto de tipo Menu al que se le pasará por parámetro la referencia que apunta al objeto de tipo JuegosReunidos.
2) Se invoque, a través de la referencia de tipo Menu, el método ejecuta.