

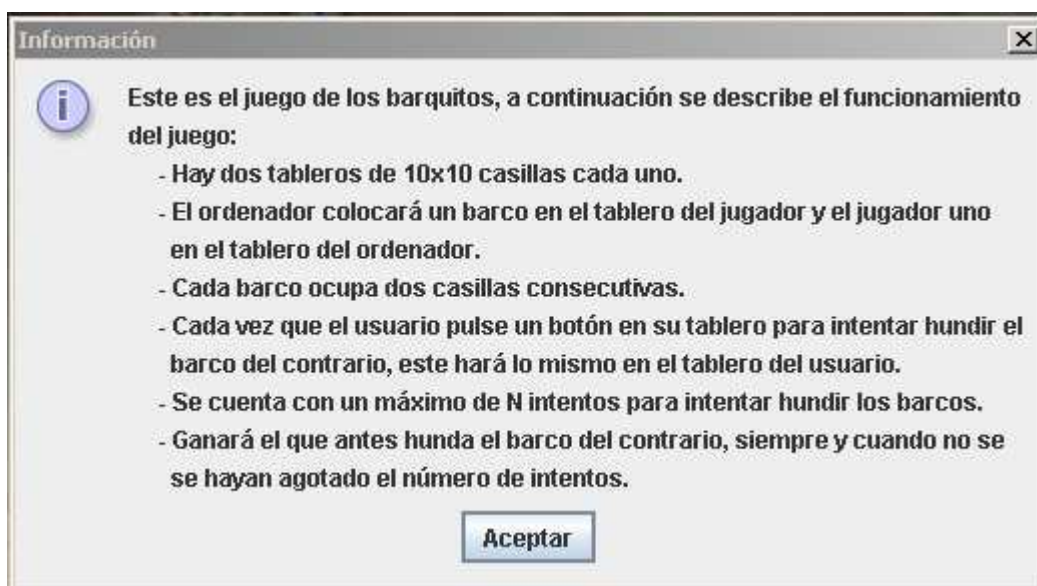
PRÁCTICA 15 DE JAVA

Se pide diseñar una aplicación para jugar a los barquitos con el ordenador. El juego consiste en lo siguiente:

- El ordenador situará un barco de dos cuadrículas de forma aleatoria en un tablero de 10x10 elementos.
- El usuario situará otro barco de dos cuadrículas en otro tablero de 10x10 elementos.
- El ordenador tratará de hundir el barco del usuario, y el usuario el del ordenador
- Se dispone de un máximo de N intentos por jugador para hundir el barco del contrincante.
- Ganará el que antes hunda el barco contrario.
- En el caso de que ninguno de los dos jugadores consiga hundir el barco del contrincante se considerará empate.
- Cuando se consigue localizar sólo una de las cuadrículas que forman el barco, se considera que el barco está TOCADO.
- Cuando se consiguen localizar las dos cuadrículas que forman el barco, se considera que el barco está HUNDIDO..

Se propone el siguiente funcionamiento para la aplicación:

1. Inicialmente se presentará una pantalla con las instrucciones del juego, por ejemplo:



Nota: Se recomienda hacer esta pantalla en una clase aparte, **Clase DialogoInstrucciones** y con la ayuda del método `JOptionPane.showMessageDialog(...)`

2. Una vez que se cierra la ventana con las instrucciones aparecerá otra en la que se pida el número de intentos para hundir los barcos, así como las coordenadas del barco que el usuario va a colocar al ordenador. Se deberá controlar que sea correcto el número de intentos, es decir, un valor numérico entre 1 y 100, así como que las coordenadas del barco sean correctas, para ello deben ser dos casillas consecutivas horizontal o verticalmente. La ventana podría ser como la que se muestra a continuación en la parte izquierda de la página, así mismo también se muestran dos ventanas con mensajes de error que aparecerán bien cuando se introduzca un valor incorrecto en el número de intentos, o bien, cuando las coordenadas no sean consecutivas. Estos errores los estamos controlando mediante **Excepciones**.

El funcionamiento de los botones de la ventana que solicita los parámetros del juego, será el siguiente:

- Botón “Comenzar el juego”: validará el número de intentos y las coordenadas, y si todo es correcto dará paso a la pantalla principal del juego, en caso de que algo sea incorrecto lanzará la correspondiente excepción.
- Botón “Salir del juego”: cerrará la aplicación.

Se recomienda hacer todo esto en una clase aparte, por ejemplo **Clase DialogoParametrosJuego**, cuya cabecera podría ser similar a:

```
public class DialogoParametrosJuego extends JDialog implements ActionListener
```

3. Por último cuando se cierre esta ventana aparecerá la ventana principal del juego, cuyo interfaz podría ser similar al siguiente:

El funcionamiento del juego podrá ser el siguiente, cada vez que el usuario pulse un botón en el “Panel del Usuario”, de forma automática, aparecerá un botón pulsado en el “Panel del Ordenador”, y se irá decrementando el número de intentos. Los botones irán tomando los colores celeste (agua), naranja (tocado) o rojo (hundido). Al finalizar se deberá mostrar una ventana informativa indicando quien gana, o si hay empate, al cerrar dicha ventana se saldrá de la aplicación.



Consejos para empezar y organizar todo esto:

La práctica se debe desarrollar siguiendo el Modelo-Vista-Controlador (MVC). Para ello vamos a ir siguiendo los siguientes pasos:

1.- Crear el **MODELO**, que estará formado por las clases **Barco** y **JuegoBarco**, cada una de estas clases llevará el siguiente funcionamiento:

- **Clase Barco**: esta clase tendrá dos variables de instancia, una que será un array (matriz 10x10) de tipo char que almacena un tablero en el que en dos casillas estará el barco, y en el resto habrá agua, y otra variable que almacene el estado en el que en ese momento está el tablero al ir destapando casillas, pudiendo ser este: tocado, hundido, ni tocado ni hundido. Sus métodos podrán ser:
 - Constructor: se encargará de preparar el tablero y de colocar en él, en las coordenadas indicadas el barco.

```
public Barco(int coorX1, int coorY1, int coorX2, int coorY2)
```
 - Método que comprueba las coordenadas dadas, devuelve *true* si se ha conseguido hundir el barco y *false* en caso contrario.
 - Método *boolean* que devuelva si el barco se ha hundido o no.
 - Métodos get de las variables de instancia.

- **Clase JuegoBarco:** esta clase guardará la siguiente información, dos objetos de tipo Barco, uno con el barco del usuario, y otro con el barco del ordenador, así como una variable numérica con el número de intentos máximo para hundir el barco. Sus métodos podrían ser los siguientes:

- Constructor: se encargará de crear los dos tableros para jugar al juego de los barquitos, así como de inicializar la variables con el número de intentos, para el tablero del ordenador, nos llegarán las coordenadas en las que el usuario ha colocado el barco, para el tablero del usuario, se generará de forma aleatoria las coordenadas en las que irá colocado el barco.

```
public JuegoBarco(int coorX1, int coorY1, int coorX2, int coorY2,
                 int intentos)
```

- Métodos get de las variables de instancia.
- Método set de la variable intentos.
- Método en que el usuario marca una casilla en el tablero del ordenador. Devuelve *true* o *false*, en función de ha conseguido hundir el barco o no.

```
public boolean juegaUsuario(int fil, int col)
```

- Método en que el ordenador marca una casilla en el tablero del usuario. Devuelve *true* o *false*, en función de ha conseguido hundir el barco o no.

```
public boolean juegaOrdenador(int fil, int col)
```

- Método que genera de forma aleatoria las coordenadas de una casilla del tablero.

```
public int[] generaCoordenadasAleatorias()
```

2.- Una vez que tenemos el modelo, deberíamos probarlo en modo consola (sin interfaz gráfico todavía), y comprobar que funciona correctamente. Para ello crear la clase MainConsola, cuyo código podría ser similar al siguiente:

```
public static void main(String[] args) {

    int fil, col;
    int [] coorAleatorias;

    JuegoBarco juego = new JuegoBarco(3,3,3,4,5);
    do
    {
        System.out.println("Fila: ");
        fil = leeInt();
        System.out.println("Columna: ");
        col = leeInt();
        juego.setIntentos(juego.getIntentos()-1);
        coorAleatorias=juego.generaCoordenadasAleatorias();

    }
    while ((juego.juegaUsuario(fil,col)!=true) &&
           (juego.juegaOrdenador(coorAleatorias[0],coorAleatorias[1])!=true) &&
           (juego.getIntentos()!=0));

    if (juego.getBarcoOrd().hundido())
        System.out.println("Gana el usuario");
    else
        if (juego.getBarcoUsu().hundido())
            System.out.println("Gana el ordenador");
        else
            System.out.println("Nadie gana, intentos agotados");

}
```

4.- Una vez que tenemos hecho el modelo y hemos comprobado que funciona, vamos a pasar a desarrollar la **VISTA**. Para ello desarrollaremos las siguientes clases:

- Hacer las clases **DialogoInstrucciones** y **DialogoParametrosJuego**, son clases de apoyo a la VISTA.
- **Clase Vista** que como siempre heredar  de JPanel, prepararemos un panel con la pantalla principal del juego, desde ella crearemos objetos de las clases DialogoInstrucciones y DialogoParametrosJuego. No olvid is estructurar correctamente la clase, no vay is a hacerlo todo en el constructor, que la interfaz es compleja y larga. Tampoco olvid is los m todos get de aquellas variables de instancia que os hagan falta, as  como el m todo control, que a ade control a los botones del panel con el que juega el usuario.

4.- Concluida la VISTA, vamos a pasar a desarrollar el **CONTROLADOR**, es decir a programar los eventos que se produzcan en este caso cada vez que el usuario pulsa un bot n en el tablero del ordenador, el bot n cambiar  de color, y simult neamente se pulsar  de forma autom tica otro bot n en el tablero del usuario, generado aleatoriamente por el ordenador. Debemos llevar tambi n un control de si se ha conseguido hundir el barco. Todo esto estar  recogido en la **clase Controlador**.

5.- Para probar el funcionamiento final necesitamos una **clase Main**, que enlaza la Vista y el Controlador y el Modelo, cuyo c digo podr a ser como el siguiente:

```
public static void main(String [] args)
{
    // Crear la vista
    Vista miVista = new Vista();

    // Crear el modelo
    JuegoBarco juego = new JuegoBarco
        (miVista.getCoorX1(), miVista.getCoorY1(),
         miVista.getCoorX2(), miVista.getCoorY2(),
         miVista.getIntentosIni());

    // Crear el controlador
    Controlador ctr = new Controlador(miVista, juego);

    // Enlazar la vista y el modelo
    miVista.control(ctr);

    JFrame ventana = new JFrame("Juego de los Barquitos");
    ventana.setContentPane(miVista);
    ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ventana.pack();
    ventana.setVisible(true);
}
```

6.- Por  ltimo tambi n necesitareis una clase vuestra propia, **clase MiExcepcion** para lanzar excepciones desde la clase DialogoParametrosJuego, cuando las coordenadas que os den no sean consecutivas.