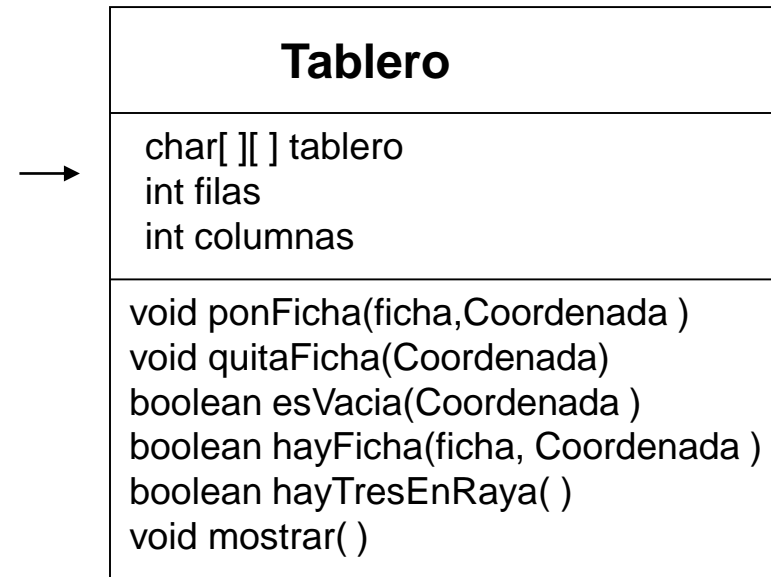
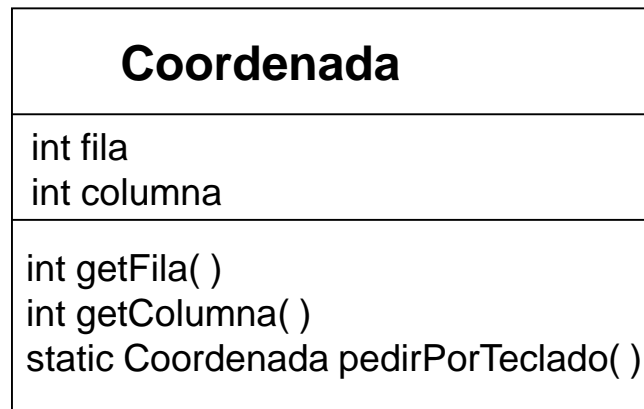
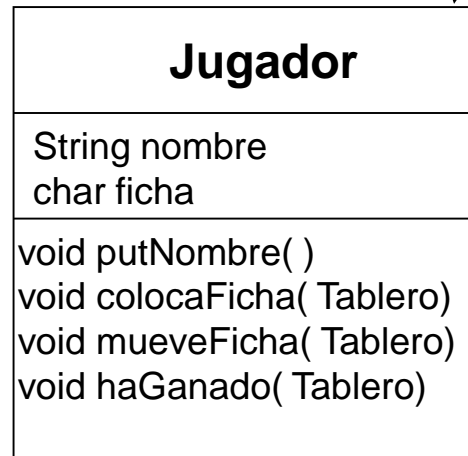
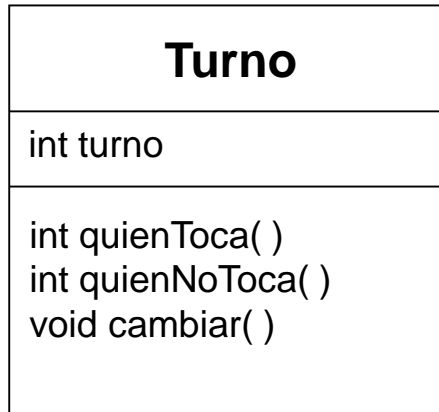
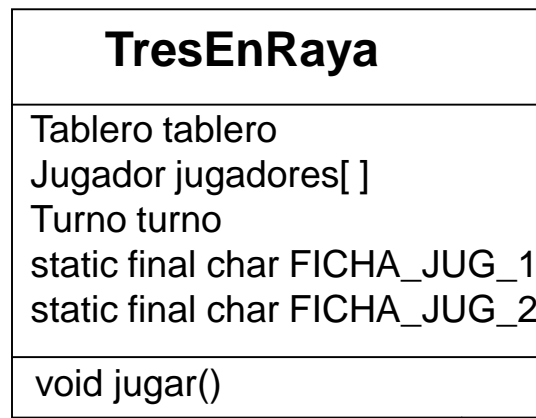


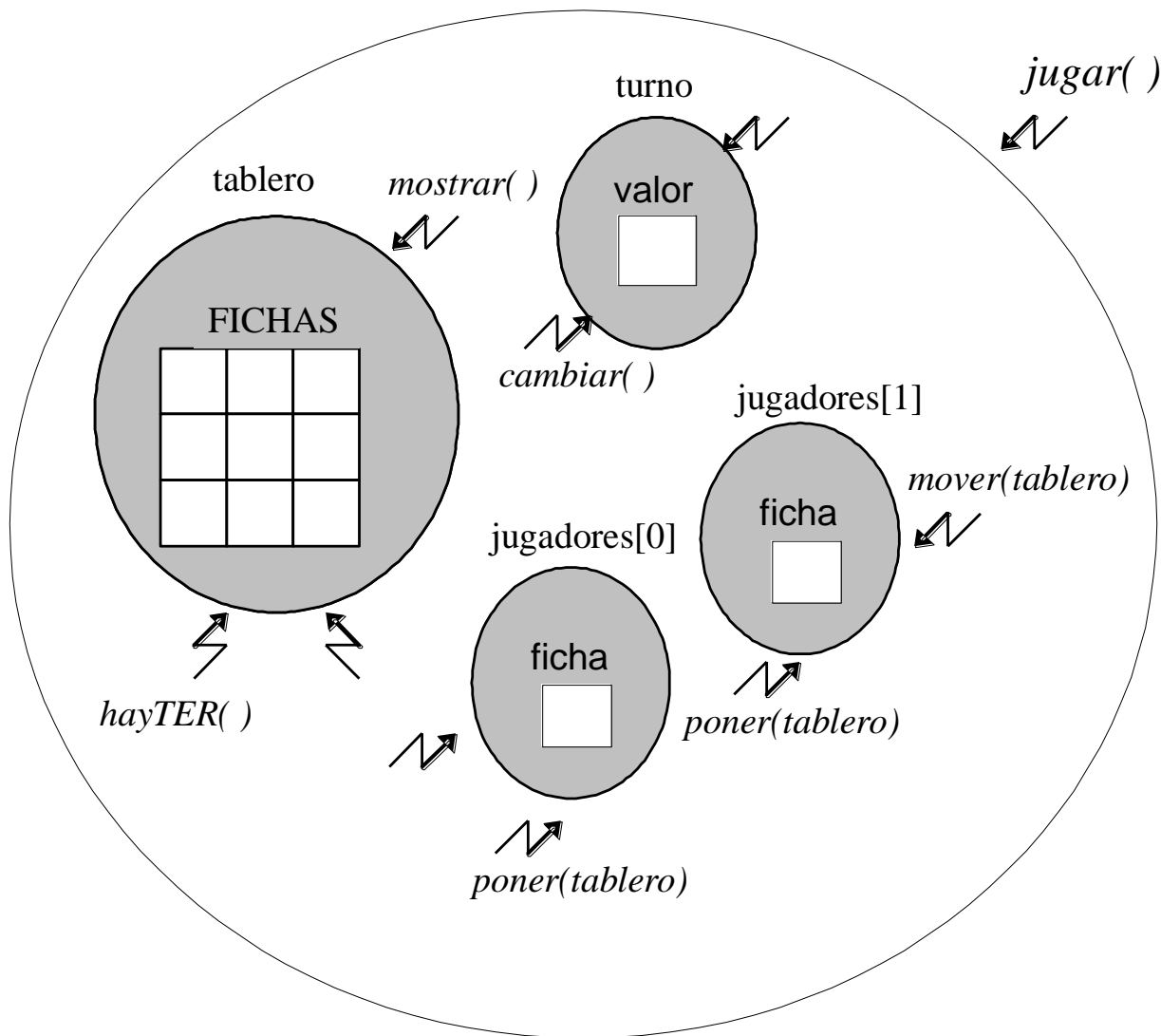
# Tres en Raya (P.O.O. en JAVA)

## Enunciado:

- Implementar el juego de las tres en raya, utilizando programación orientada a objetos.
- En el juego de las tres en raya hay un tablero de 3x3 y dos jugadores, donde cada uno juega con 3 fichas. Para que un jugador gane tiene que conseguir que sus tres fichas formen una línea a lo largo del tablero, bien en una fila, en una columna o en una de las dos diagonales
- Al inicio del juego el tablero se encuentra vacío, sin ninguna ficha. Mediante un turno alternativo, los jugadores van poniendo fichas, evitando aquellas posiciones que ya estén ocupadas por otra ficha
- Cuando cada jugador tenga sus respectivas 3 fichas en el tablero, alternativamente, cada uno irá realizando movimientos (en cada movimiento una sola ficha) para conseguir tres en raya.
- En el momento que un jugador consiga tres en raya se mostrará un mensaje por pantalla indicando quién ha ganado y finaliza la partida.



## partida



Visión  
dinámica

## **Clase de Turno**

### Atributos:

- turno es un número entero que cambiará entre 0 y 1.

### Constructor:

- crea un objeto turno, que empiece en 0;

### Métodos:

- int quienToca( ) devuelve el valor actual del atributo “turno”
- int quienNoToca( ): devuelve el valor opuesto al que contiene el atributo “turno”
- void cambiar( ) : cambia el valor del atributo “turno”.

## **Clase Coordenada:**

### Atributos:

- fila: contiene el numero que identifica la fila
- columna: contiene el numero que identifica la columna

### Constructor

- construye una coordenada, a partir de una fila y una columna

### Métodos:

- int getFila( ): devuelve el valor de la fila
- int getColumna( ): devuelve el valor de la columna
- static Coordenada pedirPorTeclado( ): deberá pedir por teclado los valores de la fila y de la columna, y devolver un objeto Coordenada.

## Clase Tablero

### Atributos:

- tablero: es un array bidimensional de caracteres del tamaño del tablero (3x3)
- int filas: es el número de filas del tablero (3)
- int columnas: es el número de columna del tablero (3)

### Constructor:

- El constructor no recibe parámetros. Inicializa tablero, filas y columnas. El tablero lo llena del valor '\_'

### Métodos:

- void ponFicha(char ficha, Coordenada): poner la ficha, en la coordenada, que se pasan como parámetro. No comprueba nada.
- void quitaFicha(Coordenada): poner en la coordenada que se pasa como parámetro, el carácter vacío '\_'. No comprueba nada.
- boolean esVacía(Coordenada): devuelve si la coordenada está vacía o no.
- boolean hayFicha(char ficha, Coordenada): devuelve si la ficha del parámetro, está en la coordenada del parámetro.
- boolean hayTresEnRaya( ): devuelve si en el tablero hay tres en raya o no (revisar todas las filas, las columnas y las diagonales).
- void mostrar( ): muestra el tablero por pantalla.

## **Clase Jugador:**

### Atributos

- nombre: almacena el nombre de este jugador
- ficha: almacena la ficha de este jugador

### Constructores

- Jugador (char ficha, String nombre)
- Jugador (char ficha)

### Métodos:

- void putNombre( ): pide por pantalla el nombre del jugador.
- void colocarFicha (Tablero) es cuando todavía no están introducidas las 3 fichas. En este método se deberá:
  - Crear un objeto Coordenada y pedir sus datos por teclado (usar el método estático)
  - Comprobar que esa coordenada del tablero está vacía. Si es así, llamar al método del tablero que coloca la ficha del jugador en una coordenada; y en caso contrario, volver a pedir la coordenada por teclado.
- void mueveFicha (Tablero) es cuando ya están las 3 fichas introducidas. Hay que borrar una ficha y ponerla en otra posición. Para ello se deberá:
  - Crear una coordenada para quitar una ficha. Para ello, se pedirán los datos de la coordenada por teclado y se comprobará si hay una ficha del jugador en esa coordenada, para eliminarla. Si no fuera así, se volvería a pedir la coordenada.
  - Después, se pedirá otra coordenada para colocar la ficha. Luego sería como el método “colocarFicha”
- void haGanado (Tablero) es mostrar un mensaje por pantalla diciendo que ya ha ganado el jugador que está en ese objeto y a continuación muestra el tablero por pantalla

## **Clase de TresEnRaya**

### Atributos:

- tablero: es el tablero de las tres en raya.
- jugadores[]: es un array para guardar los dos jugadores
- turno: me dirá si es el turno del jugador 0 o del jugador 1.
- static final char FICHA\_JUG\_1 : guarda la ficha del jugador 1(ej: X)
- static final char FICHA\_JUG\_2 = 'O' : : guarda la ficha del jugador 2(ej: O)

### Constructor:

- crea un objeto turno.
- crea un objeto Tablero
- crea un array con dos jugadores y les asigna su ficha, con los atributos estáticos

### Métodos:

- void jugar( ) : empieza el juego.
  - Llama a la función putNombre( ), del jugador, para pedir los nombres de los jugadores.
  - Los jugadores colocarán sus 3 fichas alternativamente según su turno (usar el método del jugador)
  - Si hay 3 en raya, se felicita el jugador ganador
  - Si no hay 3 en raya, el jugador que tenga el turno, mueve una ficha (usar el método del jugador), y se vuelve a preguntar si hay 3 en raya.
  - El juego termina cuando un jugador, consigue 3 en raya.

¡Ojo!, el turno puede tomar los valores 0 y 1, que son las posiciones del array jugadores. Utilizar el turno para indicar qué posición del array estamos utilizando en cada momento. Recuerda cambiar el turno, después de cada uso (turno.cambiar()).

## Ejemplos de partes del código:

```
//Método de la clase Main
public static void main(String[] args) {
    TresEnRaya partida = new TresEnRaya();
    partida.jugar();
}
```

//Extracto del método jugar, de la clase TresEnRaya

```
while (!tablero.hayTresEnRaya()) {
    jugadores[turno.quienToca()].mueveFicha(tablero);
    turno.cambiar();
    tablero.mostrar();
}
//El jugador al que NO le toca, ha ganado, pues ha cambiado el turno
jugadores[turno.quienNoToca()].haGanado(tablero);
```



//Método hayTresEnRaya de la clase Tablero

public boolean hayTresEnRaya( ) //devuelve si en el tablero hay tres en raya o no.

```
{
    for (int i=0;i<filas;i++){ //Revisamos las filas
        if (tablero[i][0]==tablero[i][1] && tablero[i][0]==tablero[i][2] && tablero[i][0]!='_'){
            return true;
        }
    }

    for (int i=0;i<columnas;i++){ //Revisamos las columnas
        if (tablero[0][i]==tablero[1][i] && tablero[0][i]==tablero[2][i] && tablero[0][i]!='_'){
            return true;
        }
    }

    //Revisamos las diagonales
    return (tablero[0][0] != '_' && tablero[0][0] == tablero[1][1] && tablero[1][1] == tablero[2][2]) ||
        (tablero[0][2] != '_' && tablero[0][2] == tablero[1][1] && tablero[1][1] == tablero[2][0]);
}
```

### **Posibles mejoras a este programa:**

- Cuando se muestra el nombre del jugador, mostrar también la ficha que utiliza
- Controlar errores al introducir las coordenadas
- Mostrar las coordenadas, al mostrar el tablero
- ....