Problema de Programación (Ing. Telecomunicaciones)

Enunciado

Queremos modelar el juego Conecta 4 y desarrollar un bucle que permita competir a dos humanos. Para ello, se pide desarrollar una clase Java Connect con los siguientes métodos:

- Método main, que tendrá el bucle del juego.
- Método actualizaEstadoPartida, actualiza un enum con los siguientes valores (enJuego, ganadorO, ganadorX, empate).
- Método mueve, que valida si la entrada y realiza el movimiento usando un input.
- Método pintaTablero, que pintará el tablero tras cada turno.
- Método mueve, que realiza el movimiento por parte del PC.

Nota: el tablero tiene 7 posiciones horizontales y 6 verticales.

Nota 2: hay dos jugadores, con fichas O y X respectivamente. El jugador con O empieza la partida.

Solución

Por simplicidad lo haremos todo en el mismo fichero Conecta.java. Comencemos con los imports:

```
import java.util.Scanner;
import java.util.ArrayList;
```

En primer lugar, dentro de la clase Conecta definiremos el enum con los valores posibles del Estado (este enum lo podríamos haber definido en un fichero auxiliar o fuera de la clase principal).

```
public class Conecta{

    enum Estado {
        EN_JUEGO,
        EMPATE,
        GANAO,
        GANAX
    }
```

A continuación, definimos los atributos de la clase que almacenarán el estado del juego (incluyendo el tablero):

```
    private final char[] jugadores = new char[] { 'O', 'X' };
    private final int ancho = 7;
    private final int alto = 6;
    private final char[][] tablero;
```

```
    private int turno = 0;
    private Estado estado = Estado.EN_JUEGO;
```

Como todo salvo el tablero lo hemos inicializado con valores por defecto, en el constructor tan sólo tenemos que rellenar el tablero (usamos un espacio para las posiciones no ocupadas):

```
    public Conecta() {
        this.tablero = new char[this.ancho][this.alto];
        for (int x = 0; x < this.ancho; x++) {
            for (int y = 0; y < this.alto; y++) {
                this.tablero[x][y] = ' ';
            }
        }
    }
```

Pasamos ya a los métodos principales:

```
    public String pintaTablero() {
        String salida = "1234567\n";
        for (int y = 0; y < this.alto; y++) {
            for (int x = 0; x < this.ancho; x++) {
                salida += this.tablero[x][this.alto - y - 1];
            }
            salida += '\n';
        }
        return salida;
    }

    public void mueve(Scanner input) {
        System.out.print("\nTurno del jugador " + this.jugadores[this.turno]);
        System.out.println("\nIntroduce un número de columna");
        do {
            int col = input.nextInt() - 1;

            if (! (0 <= col && col < this.ancho)) {
                System.out.println("La columna debe estar entre 1 y 7");
                continue;
            }
            for (int y = 0; y < this.alto; y++) {
                if (this.tablero[col][y] == ' ') {
                    this.tablero[col][y] = this.jugadores[this.turno];
                    this.turno = (this.turno - 1) * (this.turno - 1);
                    return;
                }
            }
            System.out.println("La columna " + (col + 1) + " está llena.
Selecciona otra");
        } while (true);
    }
```

```
public char ganadorLinea(char[] linea) {
    char actual = linea[0];
    int longitudLinea = 1;
    for (int j = 1; j < linea.length; j++) {
        char nuevo = linea[j];
        if (nuevo == actual) {
            longitudLinea++;
        } else {
            actual = nuevo;
            longitudLinea = 1;
        }
        if (longitudLinea == 4 && actual != ' ') {
            return actual;
        }
    }
    return ' ';
}

public ArrayList<char[]> obtenerLineas() {
    ArrayList<char[]> lineas = new ArrayList<char[]>();

    // Busca lineas horizontales
    for (int y = 0; y < this.alto; y++) {
        char[] nuevaLinea = new char[this.ancho];
        for (int x = 0; x < this.ancho; x++) {
            nuevaLinea[x] = this.tablero[x][y];
        }
        lineas.add(nuevaLinea);
    }

    // Busca lineas verticales
    for (int x = 0; x < this.ancho; x++) {
        char[] nuevaLinea = new char[this.alto];
        for (int y = 0; y < this.alto; y++) {
            nuevaLinea[y] = this.tablero[x][y];
        }
        lineas.add(nuevaLinea);
    }

    // Busca lineas diagonales /
    for (int x = -this.alto + 1; x < this.ancho; x++) {
        char[] nuevaLinea = new char[this.alto];
        for (int y = 0; y < this.alto; y++) {
            int posx = x + y;
            if ( (posx < 0) || (posx >= this.ancho))  {
                nuevaLinea[y] = ' ';
            } else {
                nuevaLinea[y] = this.tablero[posx][y];
            }
        }
        lineas.add(nuevaLinea);
    }
```

```
        // Busca lineas antidiagonales \
        for (int x = 0; x < this.ancho + this.alto - 1; x++) {
            char[] nuevaLinea = new char[this.alto];
            for (int y = 0; y < this.alto; y++) {
                int posx = x - y;
                if ( (posx < 0) || (posx >= this.ancho))  {
                    nuevaLinea[y] = ' ';
                } else {
                    nuevaLinea[y] = this.tablero[posx][y];
                }
            }
            lineas.add(nuevaLinea);
        }
        return lineas;
    }

    public void actualizaEstadoPartida() {
        boolean tableroLleno = true;
        for (int x = 0; x < this.ancho; x++) {
            if(this.tablero[x][this.alto - 1] == ' ') {
                tableroLleno = false;
            }
        }

        ArrayList<char[]> lineasTablero = this.obtenerLineas();
        for (char[] linea : lineasTablero) {
            char ganador = this.ganadorLinea(linea);
            if (ganador == 'X') {
                this.estado = Estado.GANAX;
                return;
            }
            if (ganador == 'O') {
                this.estado = Estado.GANAO;
                return;
            }
        }
        if (tableroLleno) {
            this.estado = Estado.EMPATE;
            return;
        }
        this.estado = Estado.EN_JUEGO;
        return;
    }
```

Nótese que la lógica de comprobar si hay una línea se ha dividido en varios métodos: por un lado, tenemos un método que devuelve un ArrayList con todas las líneas del tablero (horizontales, verticales y ambas diagonales). Luego, hay un método que comprueba si en una determinada línea hay un ganador. Y por último tenemos el método principal de actualizar el estado que hace uso de los dos anteriores.

Por último, el método main con el bucle principal del juego:

```
    public static void main(String[] args) {
        try (Scanner input = new Scanner(System.in)) {
            Conecta conecta = new Conecta();
            while(conecta.estado == Estado.EN_JUEGO) {
                System.out.println(conecta.pintaTablero());
                conecta.mueve(input);
                conecta.actualizaEstadoPartida();
            }
            System.out.println(conecta.pintaTablero());
            System.out.println("La partida ha terminado");
            System.out.println("El resultado es " + conecta.estado);
        }
    }
```