

Ejercicio

Contextualización

Se está desarrollando versión digital del juego de mesa “La batalla naval”. El desarrollo se lleva a cabo utilizando el paradigma de Programación Orientada a Objetos. Hasta el momento, la estructura del proyecto es la siguiente:

```
src
├── app
│   └── Main.java
├── entidades
│   ├── Barco.java
│   └── Tablero.java
└── enums
    └── Orientacion.java
```

Todas las clases del proyecto **ya se encuentran implementadas** a excepción de la clase `Tablero`.

Consigna

Implemente la clase `Tablero` considerando las siguientes especificaciones:

Atributos

- **TAMAÑO**: una constante estática entera que representa el tamaño del tablero y debe estar inicializada con el valor 10.
- **Cuadrícula**: una matriz bidimensional de enteros de tamaño `TAMAÑO x TAMAÑO` que representa el estado del tablero.
- **barcos**: un `ArrayList` de objetos de tipo `Barco` que representa los barcos que se hayan colocado en el tablero.

El atributo `Cuadrícula`: Como se mencionó anteriormente, este atributo es una matriz bidimensional de enteros de 10×10 que representa el estado del tablero. Cada celda de la matriz puede tener uno de los siguientes valores:

- 0: indica que la celda está vacía y que nunca fue atacada.
- 1: indica que la celda está ocupada por una parte de un barco.
- 2: indica que la celda ha sido atacada y había un barco (impacto).
- 3: indica que la celda ha sido atacada y no había ningún barco (agua).

Métodos

- `Tablero()`: constructor que inicializa el atributo `Cuadrícula` con ceros y el atributo `barcos` como una lista vacía.

- `colocarBarco(Barco barco, int fila, int columna)`: método que coloca un barco en la cuadrícula del tablero en la posición especificada por `fila` y `columna`. Si la colocación es válida¹, actualiza la cuadrícula y agrega el barco a la lista de barcos. Al agregar el barco al tablero, la `columna` y la `fila` representan la posición de la parte superior izquierda del barco. Por último, el posicionamiento debe respetar la orientación del barco (horizontal o vertical) definida en el objeto `Barco`.
- `recibirAtaque(int fila, int columna)`: método que recibe un ataque en la posición especificada por `fila` y `columna`. Si la celda contiene un barco (valor 1), actualiza la celda a 2 (impacto). Si la celda está vacía (valor 0), actualiza la celda a 3 (agua) y devuelve `false`.
- `mostrarTablero()`: método que imprime en consola el estado actual del tablero.

Documentación

Clase enumerada Orientacion

Una `Orientacion` representa las posibles orientaciones que puede tener un barco en el juego de batalla naval.

Resumen de constantes

Constante	Descripción
<code>HORIZONTAL</code>	Indica que el barco está orientado horizontalmente.
<code>VERTICAL</code>	Indica que el barco está orientado verticalmente.

Clase Barco

La clase `Barco` representa un barco en el juego de batalla naval. Los barcos son elementos que ocupan varias celdas contiguas en el tablero y pueden tener diferentes tamaños y dos orientaciones.

Constructores

Constructor	Descripción
<code>Barco()</code>	Crea un barco sin tamaño ni orientación definidos.
<code>Barco(int tamaño, Orientacion orientacion)</code>	Crea un barco con el tamaño y la orientación especificados.

Resumen de métodos públicos

Retorno	Método	Descripción
<code>Orientacion</code>	<code>getOrientacion()</code>	Devuelve la orientación del barco.
<code>int</code>	<code>tamaño()</code>	Devuelve el tamaño del barco.
<code>boolean</code>	<code>estaOrientado(Orientacion orientacion)</code>	Devuelve <code>true</code> si el barco está orientado según la orientación especificada; de lo contrario, devuelve <code>false</code> .
<code>void</code>	<code>Rotar()</code>	Cambia la orientación del barco de horizontal a vertical o viceversa.

¹ Idealmente, el método debe verificar que el barco no se salga de los límites del tablero ni se superponga con otros barcos ya colocados.