

Nombre:	<u>Nota</u>
Curso:	

Cosas permitidas y no permitidas

- El código entregado tiene que compilar sin problemas
- Cada error, falta de implementación, situación no controlada o diseño no apropiado restará de 0,25 a 2 puntos dependiendo de la gravedad
- Existen algunos errores (comentados en clase) que hará que la nota máxima de este examen sea de 2
- LA NOTA MÍNIMA PARA APROBAR 4 en este examen sobre 10



Tiempo estimado	Dificultad	Tipo de actividad
1 hora	Baja-Media	Evaluación



Se desea realizar parte de un programa basado en el juego del Ajedrez.

- En el ajedrez hay varios tipos de fichas: peones, alfiles, caballos, torres, reinas y reyes.
- Las fichas pueden ser blancas o negras.
- Cada ficha se pone en una casilla de una rejilla de 8 x 8 casillas.
- Cada ficha tiene un movimiento específico.

Se pide:

- Implementar la clase Position, que tiene un x y un y simplemente (y todo lo que quieras añadir), que son enteros.
- Implementar la interfaz IFigure, que contiene los siguientes métodos
 - GetColor
 - GetType → Devuelve el tipo de ficha que es un enum
 - GetPosition
 - MoveTo (Position)
 - GetAvailablePositions → Devuelve una lista de posiciones
 - HasBeenMoved → Devuelve si la figura se ha movido alguna vez o no
- Implementar la clase Figure que hereda de IFigure que contendrá
 - El color que no se puede cambiar
 - Posición → Sólo las clases hijas pueden acceder a este atributo
 - Un booleano que indica si se ha movido o no
 - No contendrá nada más a no ser que sea totalmente justificado
- Implementar la clase King
 - El rey sólo puede moverse una casilla hacia el lado que quiera. Si hay una casilla de diferente color allá donde se mueva el rey, se eliminará, si hay una casilla de su color, no se puede mover a esa casilla
- Implementar la clase Board
 - Contiene una lista de IFigure
 - En el constructor añade todas las fichas según el dibujo que se muestra al final
 - Tiene un método llamado RemoveFigure, al que se le pasa una x y una y, y eliminará la figura que hay en esa casilla.
 - Tiene un método que, dado un x y un y, te dice la casilla que hay en esa posición
- (1 punto) Implementar la clase Peon
 - Un peón se puede mover de la siguiente manera:
 - Los blancos se mueven hacia arriba, los negros hacia abajo
 - Si no tienen nada delante se pueden mover una casilla hacia adelante
 - Si tienen una figura de diferente color en su diagonal adyacente, se pueden mover en ella y por lo tanto eliminarla
 - Si no se han movido todavía pueden moverse dos casillas hacia adelante, siempre y cuando no haya una figura en ese camino

Especificaciones:

- Se ha de implementar siguiendo las filosofías de la programación orientada a objetos.
- Si surge algún imprevisto resuélvelo

