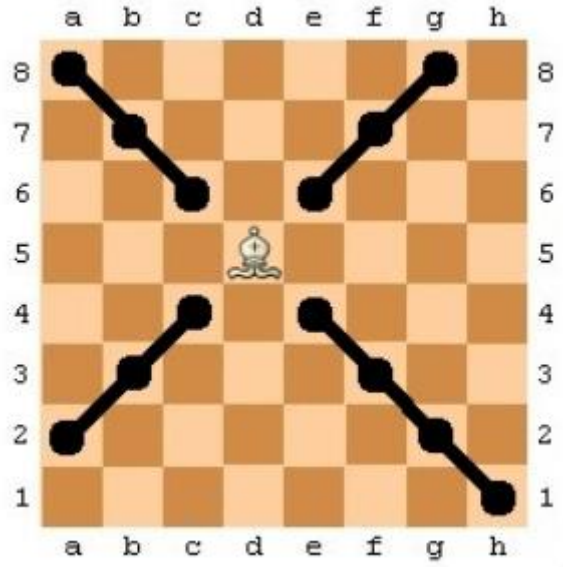
EJERCICIO 1. ARRAY 2D: MOVIMIENTOS DE AJEDREZ.



Dado un tablero de 8x8 casillas, donde las columnas se indican con letras de "a" a "h" y las filas del 1 al 8 como se ve en la imagen anterior.

Fase1. En esta ocasión nuestro programa debe solicitar al usuario que figura vamos a mover en todo momento (solicitando un número entre 1 y 5):

- 1: Peón
- 2: Alfil
- 3: Caballo
- 4: Reina
- 5: Rey

Si el usuario selecciona un número no correcto se le volverá a pedir un número, mostrando el siguiente mensaje: "Figura incorrecta seleccione un número válido entre 1 y 5. Inserte una figura correcta".

Fase2. Nos pedirá una posición de partida de la figura en el tablero, pidiendo tanto una letra [a … h] y un número [1 … 8]

Nuevamente, si el usuario pone una posición incorrecta (fuera de los límites del tablero) se le solicitará nuevamente una posición en el tablero válida.

Fase3. El programa dado un tipo de figura y una posición, mostrará los movimientos válidos permitidos: https://docs.kde.org/trunk5/es/kdegames/knights/piece-movement.html]

```
Ejemplo:
Introduzca la posición del alfil: d5
El álfil puede moverse a las siguientes posiciones:
h1 a2 g2 b3 e3 c4 e4 c6 e6 b7 f7 a8 g8
```

Fase4. El programa a continuación pedirá al usuario un movimiento:

- Si el usuario indica un movimiento inválido no permitido se le notificará al usuario con el siguiente mensaje “Movimiento inválido” y no se moverá la figura. Volver Fase4.
- Si el usuario indica un movimiento válido, moveremos la figura hasta esa posición y proporcionaremos nuevamente todos los movimientos permitidos volviendo a Fase3.
- Si el usuario introduce -1 terminará el programa. Es decir, mientras el usuario no termina el programa va movimiento la figura

PREMISAS:

1. Uso una matriz de 2D de 8x8 = 64 casillas.
2. El ejercicio debe ser completamente modular. Una posibilidad es que cada fase mostrando se una función.
3. El programa debe guiar en todo momento al usuario tanto en que posición se encuentra, que movimientos puede hacer y si se indica un movimiento válido o inválido.
4. Los movimientos válidos, contemplará tanto la característica de la figura como las limitaciones del tablero.

EJERCICIO 2. POO: MOVIMIENTOS DE AJEDREZ.

Aplicar el paradigma de programación orientada a objetos al ejercicio anterior.

Para ello, creamos una clase llamada “Figura” con los siguientes atributos:

- Tipo. // Se aconseja utilizar constantes publicas para se elección de acuerdo al ejercicio anterior.
- Posición. Puede utilizarse un atributo o dos.

Luego crearemos los métodos que consideremos oportunos, pero entre ellos debe estar los siguientes:

- getPosicionDentroTablero(): String. Proporciona la posición de la figura dentro del tablero, es decir, nos devuelve “d5” si está en la columna ‘d’ y la fila 5.
- movimiento (fila, columna): boolean. Permite al usuario mover la figura dentro del tablero. Nos devolverá true si el movimiento se ha realizado de forma satisfactoria o false si no.

PREMISAS:

- Al menos 2 clases: Figura.java y MainFigura.java
- Crear diagrama UML
- Documentar con Javadoc la clase Figura.java