

Volem implementar un conjunt de classes per simular part d'una partida del **joc de les dames**. De moment tenim creada una **classe Partida** per guardar informació general de la partida:

Partida
- _jugadors: list<list<Fitxa>> - _torn_actual: int
+ inicialitza(nom_fitxer:string) + bool mou_fitxa(inici: (int, int), final: (int, int)) + bool converteix_dama(posició: (int, int))

Com podem veure aquesta classe té com a atributs:

- **_jugadors**: és una llista de llistes que serveix per **guardar les fitxes que té cada jugador**. Cada element de la llista és una altra llista que guarda les fitxes d'un dels dos jugadors de la partida. Les fitxes es guarden utilitzant objectes d'una **classe Fitxa** que, com expliquem més endavant **haureu de definir i implementar** vosaltres.
- **_torn_actual**: és un enter que guarda quin jugador té el torn actual (pot tenir els valors 0 ó 1).

I també té els mètodes següents:

- **inicialitza**: permet **inicialitzar les fitxes** al principi de la partida, o per recuperar les posicions de les fitxes d'una partida que hem deixat a mitges i volem continuar jugant. Les posicions inicials de les fitxes es llegeixen d'un **fitxer que té aquest format**:

```
fila columna jugador tipus_fitxa  
fila columna jugador tipus_fitxa  
...
```

Cada línia del fitxer conté la posició d'una fitxa. La posició ve determinada per la fila i columna (valors entre 0 i 7) que ocupa dins del tauler. La posició (0,0) correspon a la cantonada superior esquerra del tauler. El valor del jugador pot ser 0 (blanques) ó 1 (negres) i el valor del tipus de fitxa pot ser 0 (fitxa normal) ó 1 (dama).

- **mou_fitxa**: permet **moure una fitxa del jugador que té el torn actual** des de la posició d'inici a la posició final. Les posicions inicial i final estan indicades utilitzant tuples amb el valor de la fila i la columna: (fila, columna). **Retorna un booleà** indicant **si el moviment és vàlid o no**. Un moviment pot ser invàlid perquè el jugador que té el torn actual no tingui cap fitxa a la posició inicial o perquè moure la fitxa a la posició final sigui un moviment no permès (tal com us expliquem més endavant). En aquesta versió del joc de les dames implementarem una **versió simplificada del moviment de les fitxes**, que **no considerarà l'opció de matar**.
- **converteix_dama**: converteix en dama la fitxa del jugador actual que ocupa la posició que es passa com a paràmetre. **Retorna un booleà** indicant **si la conversió s'ha pogut fer correctament**. La conversió no es podrà fer si el jugador actual no té cap fitxa a la posició indicada.

Per implementar tots aquests mètodes **necessitareu utilitzar la classe Fitxa** i els seus mètodes seguint les especificacions que us expliquem a continuació.

La **classe Fitxa** ha de servir per guardar i gestionar la informació de cadascuna de les fitxes dels dos jugadors. Les fitxes poden ser **fitxes normals o dames**, que es diferencien pel tipus de moviments que poden fer. Com hem dit abans, considerarem una **versió simplificada del moviment de les fitxes** en què no tindrem en compte l'opció de matar altres fitxes. D'aquesta forma els moviments vàlids de fitxes normals i dames quedaran definits així:

- **Fitxes normals:** les fitxes normals es poden moure una posició en diagonal. Les fitxes blanques (jugador 0) només es poden moure cap a baix del tauler i les fitxes negres (jugador 1) només es poden moure cap a dalt del tauler. En una partida real hauríem de comprovar que la posició final no està ocupada per una altra fitxa, però en aquesta versió simplificada del joc no ho haurem de comprovar. Només haurem de comprovar que la posició final està dins dels límits del tauler.
- **Dames:** Les dames es poden moure tantes posicions en diagonal com vulguin, en qualsevol direcció, tant cap a baix com cap a dalt del tauler. Per les dames aplicarà el mateix que hem comentat abans per les fitxes normals, respecte a la comprovació de la posició final. Tampoc haurem de comprovar que entre la posició inicial i la posició final no hi hagi cap altra fitxa, comprovació que sí hauríem de fer en una partida real.

A nivell **d'atributs**, aquesta classe **Fitxa** haurà de tenir tots els atributs que facin falta per poder saber la posició de la fitxa i fer el moviment de la fitxa tal com hem explicat abans. A nivell de **mètodes**, apart dels **getters** i **setters** que facin falta, només haurà de tenir algun mètode per permetre **fer el moviment simplificat** de cada tipus de fitxa tal com hem explicat abans.

A partir d'aquestes especificacions **us demanem**:

1. **Completeu el disseny de classes** necessari per implementar aquesta versió simplificada del joc de les dames. Dibuixeu el **diagrama de classes UML** complet afegint a la classe Partida, la classe o classes que facin falta per guardar la informació de les fitxes. Tingueu en compte d'aplicar tots els conceptes que hem anat explicant:
 - Separació privat/públic.
 - Incloure els getters/setters necessaris
 - Utilitzar correctament l'herència per gestionar els diferents tipus de fitxes.
 - Pensar i especificar bé els paràmetres i valor de retorn dels mètodes que faci falta declarar.
2. Feu la **implementació completa de la classe o classes** que afegiu al diagrama de classes per guardar les fitxes.
3. Feu la **implementació** dels mètodes de la classe Partida que hem indicat abans: **inicialitza, mou_fitxa, converteix_dama**.

Exercicis de Numpy (2 PUNTS)

4. Feu la **implementació** d'un nou mètode de la classe `Partida` que es digui **genera_tauler** i que generi l'estat actual de la partida en forma d'array- de *numpy*. Aquest array ha de guardar les posicions buides com a 0 i les posicions ocupades seguint la següent codificació: 1: fitxa normal blanca, -1: dama blanca, 2: fitxa normal negra, -2: dama negra. Per exemple, a l'inici d'una partida, l'array hauria de tenir aquests valors:

```
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [2 0 2 0 2 0 2 0]
 [0 2 0 2 0 2 0 2]
 [2 0 2 0 2 0 2 0]]
```

5. **Implementeu** una funció **comprova_moviment** que rebi com a paràmetres un array amb l'estat del tauler tal com es genera a la funció de l'apartat anterior i dues tuples que contenen la posició inicial i la posició final d'un possible moviment d'una dama. La funció ha de verificar que entre la posició inicial i la posició final (inclosa la posició final) no hi ha cap altra fitxa al tauler de cap dels dos jugadors. Ha de retornar un booleà indicant si el moviment és vàlid (no hi ha cap fitxa) o no.