

---

## UFR ST - Besançon- L2 Info

# POA - TD 2 - Héritage et classes abstraites

---

### Exercice. Modélisation et implantation des pièces d'un jeu d'échec

On s'intéresse dans cet exercice à vérifier la validité du déplacement des pièces dans un jeu d'échec. Cela permettrait par exemple d'apprendre à déplacer les pièces, en distinguant simplement les mouvements autorisés et les mouvements interdits.

Une **Position** sur un échiquier est constituée de deux entiers positifs. Une méthode `isValid(...)` renvoie `true` pour une position dont les deux composantes sont entre 1 et 8, et `false` sinon. La **Position** sera une classe non-mutable : on ne définit pas de *setter* permettant de la modifier, seuls les constructeurs permettent de construire une position donnée.

Une pièce d'un jeu d'échec possède un nom : pion, fou, cavalier, tour, reine ou roi, ou de manière équivalente *pawn*, *bishop*, *knight*, *rook*, *queen* ou *king* si vous préférez les nommer en anglais. En plus de son nom, une pièce possède aussi une couleur (noir ou blanc).

Une méthode `moveOK(...)` détermine si un mouvement est autorisé pour la pièce entre deux positions : sa position actuelle et une position cible. Elle renvoie `true` si les deux positions sont valides et que le mouvement est conforme à la règle de déplacement de la pièce. Elle renvoie `false` sinon. **N.B. Cette méthode ne tient pas compte des autres pièces sur l'échiquier, la prise de pièces sera gérée plus tard.**

Les règles de mouvement des pièces sont comme suit :

- *pion* : déplacement de une ou deux cases en avant au premier mouvement, déplacement de une seule case en avant ensuite ;
- *fou* : déplacement en diagonale, d'un nombre quelconque de cases ;
- *cavalier* : déplacement « en L », c'est à dire de deux cases horizontalement ou verticalement, puis de une case dans une direction perpendiculaire, ou encore de une case horizontalement ou verticalement, puis de deux cases dans une direction perpendiculaire ;
- *tour* : déplacement vertical ou horizontal, d'un nombre quelconque de cases ;
- *reine* : déplacement vertical, horizontal ou en diagonale, d'un nombre quelconque de cases ;
- *roi* : déplacement vertical, horizontal ou en diagonale, mais d'une seule case.

**Question 1.** Proposez une modélisation objet pour ce problème, à base d'héritage et de classe abstraite.

**Question 2.** On voit que les mouvements des pièces se composent de mouvements horizontaux, verticaux, diagonaux, et d'une certaine longueur. Codez en java une classe utilitaire **Move** avec des méthodes statiques pour tester qu'un mouvement (d'une position **from** à une position **to**) est vertical, horizontal, diagonal. Codez dans cette classe deux méthodes déterminant la longueur respectivement horizontale et verticale d'un mouvement.

**Question 3.** Donnez le code des différentes classes.