

Problème 2 (60 min ; 7 points)

Avec vos amis du cours d'INGE 1225, vous décidez d'écrire un programme java pour jouer aux échecs. Vous vous répartissez le travail et vous devez programmer une méthode `CheckMove` qui vérifie si les déplacements des différentes pièces de l'échiquier sont permis ou non. Un échiquier est composé de 8x8 cases (numérotées de 1 à 8 dans les deux sens) et, dans notre cas, la couleur des cases n'a pas d'importance. Vous voulez commencer par trois pièces uniquement : le fou, la tour et le cavalier. Pour rappel, le fou se déplace uniquement en diagonale, la tour en ligne droite et le cavalier en « L » (voir schéma, un x indique un mouvement permis).



Source : fr.wikipedia.org/wiki/Echecs (consulté le 12/06/13).

Vous devez donc écrire une méthode qui renvoie *true* si le mouvement (en fonction des 5 paramètres, voir plus bas) est permis, et *false* sinon. Attention à deux cas particuliers : si le mouvement se termine ou débute hors de l'échiquier, la méthode doit renvoyer *false*. De plus, si $X = X_{\text{new}}$ **et** $Y = Y_{\text{new}}$ (c-à-d que la pièce ne se déplace pas), alors la méthode doit aussi renvoyer *false*.

X : l'ancienne position en X (abscisse) de la pièce

Y : l'ancienne position en Y (ordonnée) de la pièce

Xnew : la nouvelle position en X de la pièce

Ynew : la nouvelle position en Y de la pièce

Type : le type de la pièce : vous pouvez faire l'hypothèse que vous n'aurez que les types = "fou" ou "tour" ou "cavalier".

Signature de la méthode :

```
public static boolean CheckMove(int X, int Y, int Xnew, int Ynew, String Type)
```