

Partie 1 : Construction d'un Quadtree

Pour représenter une image (carrée), on peut procéder de cette manière :

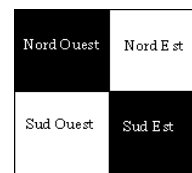
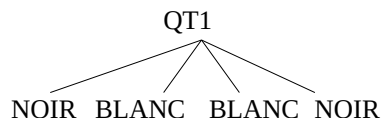
1. Si l'image est uniformément blanche, il suffit d'enregistrer ce fait. Nous écrirons BLANC pour une telle image, et si l'image est uniformément noire, il suffit d'enregistrer ce fait. Nous écrirons NOIR pour une telle image.

2. Sinon, décomposons l'image en 4 carrés égaux, situés au nord-ouest, au nord-est au sud-ouest et au sud-est de l'image carrée initiale, et la représentation de l'image sera constituée des représentations de ces 4 sous-images selon le même principe. Nous écrirons QT(I1, I2, I3, I4) pour une telle image, avec I1 la représentation du quart nord-ouest, I2 la représentation du quart nord-est, I3 la représentation du quart sud-ouest, et I4 la représentation du quart sud-est.

Représentation par une structure de données

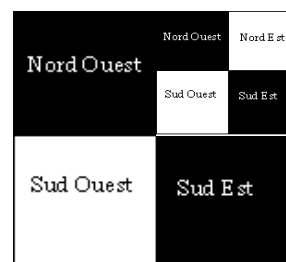
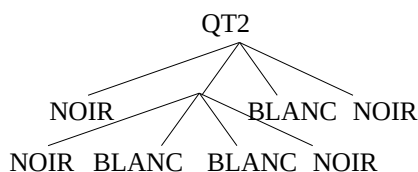
- La représentation d'une telle image se fera avec un arbre possédant quatre branches : QuadTree

QT1 = (NOIR, BLANC, BLANC, NOIR)



- Représentation d'une autre image

QT2 = (NOIR, (NOIR, BLANC, BLANC, NOIR) , BLANC, NOIR)



On peut donc dire que :

- **Blanc est un ArbreQT**
- **Noir est un ArbreQT**
- **AQTCons est un ArbreQT (décomposition en quatre parties)**
- **AQTVide est un ArbreQT**

Ouvrez le projet sur moodle et complétez en répondant aux questions

Q1: Écrivez les classes satisfaisant les définitions ci-dessus. Écrivez les constructeurs ainsi que la méthode afficher. L'affichage devra être sous cette forme :

(N (N B B N) N B)

Q2: Construisez dans un main grâce a un ArbreQT, l'image suivante :

Remarquez la redondance dans le motif, réutilisez les références...

Nord Ouest		Nord Est		Nord Ouest		Nord Est	
Sud Ouest		Sud Est		Sud Ouest		Sud Est	
Nord Ouest	Nord Est	Nord Ouest	Nord Est	Nord Ouest		Nord Est	
Sud Ouest	Sud Est	Sud Ouest	Sud Est				
Nord Ouest	Nord Est	Nord Ouest	Nord Est	Sud Ouest		Sud Est	
Sud Ouest	Sud Est	Sud Ouest	Sud Est				

Q3: Écrivez la méthode equals qui permettra d'évaluer l'égalité de deux ArbreQT.

```
public abstract boolean equals( ArbreQT a );
```

Plusieurs cas devront être envisagés pour les différents éléments des ArbreQT : Blanc, Noir, AQTVide, AQTCons.

Rmq :

- un AQTCons est égal a un autre AQTCons si les 4 composants sont égaux entre eux.



- On considère avoir tous les "setter" et getter" nécessaires.