

Durée : 2h, documents autorisés, surf interdit.

Sommaire

M11	2
Mise en place.....	2
Création du workspace et import du projet.....	2
Liste doublement chaînée	2
La cellule double.....	2
La liste doublement chaînée	3
Les quadrees	4
Création d'un quatree	4
Inversion des couleurs.....	5
Comparaison de deux quatree	5
Restitution du travail.....	6

M11

Mise en place

Vous trouverez tous les documents nécessaires à l'url suivante <http://berdjugin.com/temp/examLP> le login est « titi » et le mot de passe « toto ».

Création du workspace et import du projet

Vous pouvez utiliser votre « workspace » habituel, vous devrez lancer eclipse, importer le projet et enfin le renommer :

1. Lancer eclipse
2. Ouvrir son workspace
3. Télécharger « exam-lp-m11-2010-2011-etu.zip»
4. Importer le projet “ exam-lp-m11-2010-2011-etu” avec File->Import->General->Existing projects into workspace -> select archive file.
5. Renommer votre projet avec refactor -> rename en exam_nom où nom est votre nom¹.

Liste doublement chaînée

La cellule double

Vous disposez de la classe « CelluleD » caractérisée par sa valeur, une référence vers la cellule suivante et une référence vers la cellule précédente.

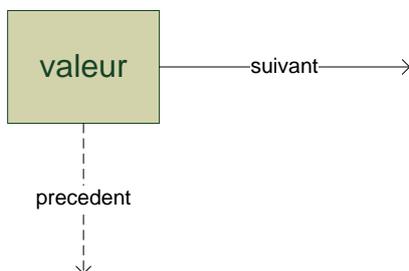
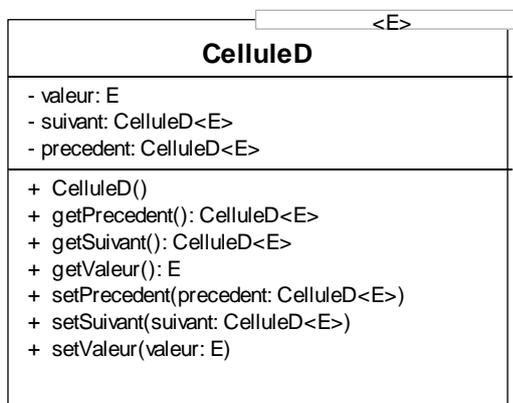


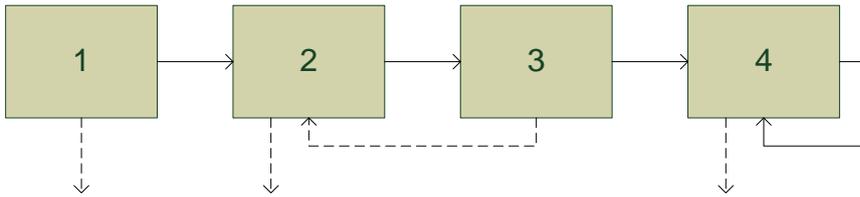
Figure 1 Une cellule double

La classe « CelluD » est fournie et dispose des méthodes suivantes :

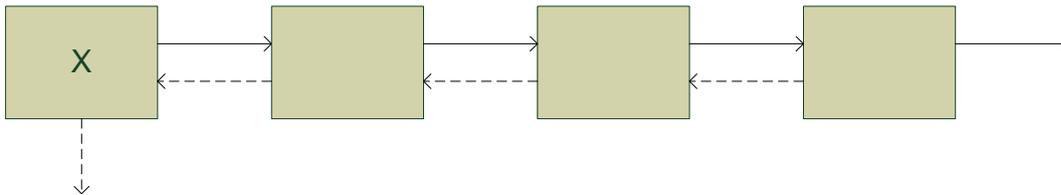


¹ Le nom du projet ne doit pas contenir d'espaces ou d'accents.

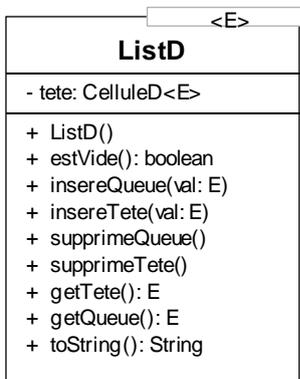
Donner le code permettant de réaliser la structure suivante :



La liste doublement chaînée



Nous allons implémenter une liste doublement chaînée en utilisant une « celluleD ».



La méthode « estVide() » est supposée donnée, vous devez donner le code des méthodes suivantes :

- « getTete() » et « getQueue() » qui renvoient respectivement la valeur du premier et du dernier élément.
- « insereTete() » et « insereQueue() » qui insèrent un élément en première et en dernière position.
- « supprimeTete() » qui supprime le premier élément.

La méthode « supprimeQueue() » est déjà codée pour que vous puissiez vous en inspirer. La classe « TestListD » peut vous aider pour vos tests.

Les quadrees

Les quadrees sont des arbres qui possèdent quatre fils. Nous allons les utiliser pour stocker une image. L'image est divisée en quatre zones associée chacune à un fils : « nw », « ne », « sw », « se ».



Figure 2 découpage de l'image

Si la couleur d'une zone est uniforme alors la feuille stocke la couleur, sinon la zone est décomposée en quatre sous-zones et le processus recommence.

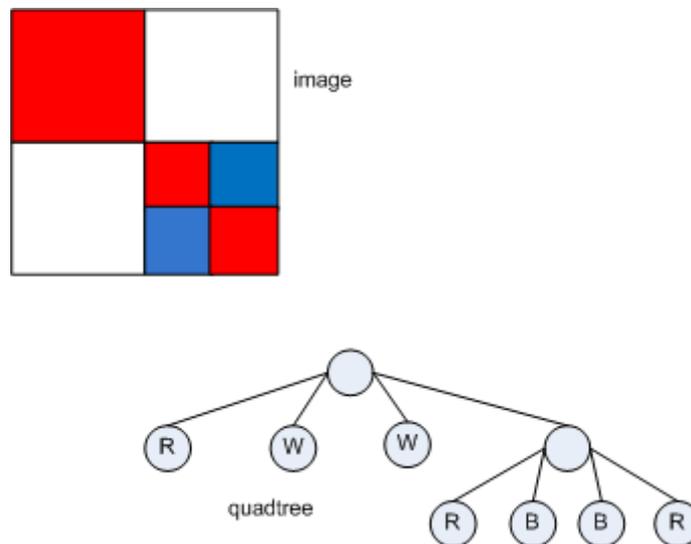


Figure 3 Image et son quadtree

Création d'un quadree

Compléter et modifier le code de la classe « TestQuadTree », pour obtenir l'image suivante. Dans le « main » l'utilisation d'un objet de type « Visualisation » permet d'avoir un rendu graphique.

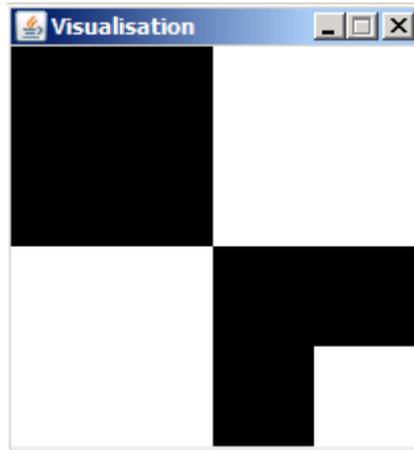


Figure 4 Image à reproduire

Vous obtiendrez du blanc avec « Color.White » et du noir avec « Color.Black ».

Inversion des couleurs

Compléter le code de la méthode « public void inverseCouleur() » qui permet d'inverser la couleur d'un quadtree. Pour l'image précédente, vous devez obtenir, après inversion, l'image suivante.

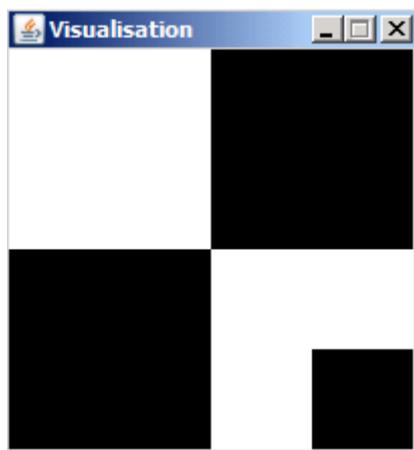


Figure 5 Image après inversion

La méthode « private static Color inverse(Color c) » est fournie et vous permet d'inverser une couleur².

Comparaison de deux quadtree

Compléter le code de la méthode « public boolean egalite(QuadTree q) » qui renvoie « true » si les quadtree sont égaux et « false » sinon.

² Pour éviter des effets de bords, vous ne pouvez utiliser « Visualisation » qu'une fois.

Restitution du travail

Pour restituer votre travail vous devez exporter votre projet puis le déposer sur une clef usb.

Pour l'export :

1. Sélectionner le projet (exam_nom)
2. Rafraichir F5
3. Bouton droit-> general -> archive file e:\nom.zip (nom étant votre nom)
4. Vérifier la taille 14ko