

Bonjour à tous

Voici quelques consignes à respecter dans le projet NFA32.

Concernant les différentes parties du projet, voici ce que j'attends d'y voir et le barème d'évaluation :

1 Structure de données (/7)

- ⇒ Liste chaînée (gestion native comme vue en NFA32. Utilisation de collection JAVA comme linkedlist interdite)
- ⇒ Gestion des exceptions (pas de test unitaire en NFA32)
- ⇒ Attention à la visibilité des méthodes et des attributs (private à privilégier)
- ⇒ Utiliser les mots clés : @override, this
- ⇒ Il n'y a pas d'héritage dans ce projet ni de classe abstraite juste une classe d'interface (voir point 4)

2 Opérations oligatoires (/6)

- Foncer ou éclaircir toutes les couleurs ayant une dominante rouge, bleue ou verte. La dominante d'une couleur est l'une des trois couleurs primaire (rouge, vert, bleu), celle qui a l'entier le plus grand. Pour foncer ou éclaircir une couleur, on ajoute ou on retranche un certain entier à chacun des trois entiers la décrivant. Par exemple, si on veut foncer les dominantes rouges, on changera par exemple la couleur 10 5 6 en 14 9 10. En revanche 2 15 10 restera inchangée parce que cette couleur n'a pas une dominante rouge mais une dominante verte.
 - Passer en noir et blanc, en utilisant des niveaux de gris. Les gris sont obtenus en donnant aux trois nombres la valeur de la moyenne des trois valeurs d'origine.
 - Afficher la taille courante de l'image (largeur x hauteur), en nombre de pixels.
 - Découper et garder seulement une partie de l'image, entre les lignes *l1* et *l2* et les colonnes *c1* et *c2*. Les valeurs *l1*, *l2*, *c1* et *c2* sont précisées par l'utilisateur, qui peut se baser sur le nombre de lignes et de colonnes de l'image, affichées par l'opération précédente. Si cela vous semble trop difficile, vous pouvez programmer une variante simplifiée, où l'image est découpée entre les lignes *l1* et *l2*, en gardant toutes les colonnes.
- ⇒ En plus des impératifs ci-dessus, récursivité attendue pour ces fonctions. Attention à bien placer les méthodes dans les bonnes classes (les deux premières fonctions concernent le pixel, alors que la dernière concerne l'image par exemple)

3 Opérations optionnelles (/4)

- Créer le négatif d'une image. Pour cela, on change chaque couleur dans son complémentaire.
- Elargir une image.
- Foncer ou éclaircir les couleurs comprises dans une certaine plage de couleurs, caractérisées par des intervalles de couleurs primaires (ou autre caractérisation).
- Agrandir ou rapetisser une image dans les deux dimensions (c'est beaucoup plus difficile que les opérations précédentes).
- Incruster dans l'image des rectangles d'une certaine couleur (c'est également une opérations difficile).
- incruster une image sur un nouveau fond. Il s'agit de remplacer dans l'image à incruster un fond d'une couleur unie (par exemple blanc) par des couleurs venant d'un autre image (voir l'exemple à la fin de ce document). Pour simplifier la tâche, on peut imposer que les deux images aient la même taille. C'est là l'opération la plus difficile.

~~Vous pourrez également imaginer et réaliser d'autres transformations.~~

⇒ Mêmes consignes qu'au dessus

4 Interface (/3)

Vous réaliserez un menu proposant de lire une image contenue dans un fichier, d'écrire une image transformée dans un fichier et de réaliser les différents opérations.

En option, vous pourrez proposer d'avoir plusieurs images en mémoire à un moment donné.

- ⇒ Utiliser une classe d'interface pour limiter les accès au stricte nécessaire au niveau de ce menu.
- ⇒ Un tableau natif ou une liste chaînée native (pas de collection) pour la liste des images ouvertes (cette partie n'est plus en option !!!)

N'hésitez pas à me contacter par mail, ou via le forum si vous avez des questions. En cas d'incompréhension vous pouvez aussi dialoguer entre vous via le forum.

Bon courage et prenez soin de vous,

Cordialement

Estelle