

### TD 5 : éditeur d'images

#### *Éditeur d'image*

On veut programmer un éditeur d'images (genre gimp) qui permet de charger des images, de les afficher et de les transformer de diverses façons, notamment les flouter et les transformer en leur négatif. Indiquez les classes qui vont intervenir.

On a une classe `image` représentant une image. Écrivez les déclarations des classes `flou` et `negatif` qui permettent respectivement de flouter et de transformer en négatif une image. L'éditeur d'images est représenté par la classe `ngimp` (nano gimp) suivante :

```
class ngimp {
public :
    ngimp();
    ~ngimp();
    void run();
private :
    void afficheImage();
    void transformeImageFlou();
    void transformeImageNégatif();
};
```

Quelle donnée faut-il ajouter, écrivez le constructeur. La méthode `run()` lance une boucle d'exécution organisée autour d'un menu, quand l'utilisateur veut flouter l'image, (resp. la transformer en son négatif), la méthode `transformeImageFlou()` (resp. `transformeImageNégatif()`) sera appelée : il faut transformer l'image et la ré-afficher. Écrivez les définitions de ces deux méthodes.

#### *Annuler la dernière transformation*

On veut pouvoir annuler (seulement) la dernière transformation (méthode `annuleDerniereTransformation()`). Quel est le moyen simple de faire cela ? Écrivez toutes les modifications nécessaires.

Quelque chose commence à ne pas aller avec les méthodes `transformeImageFlou()` et `transformeImageNégatif()` : quoi, que faire pour éviter ce problème ?

#### *Transformations d'images*

Écrivez la classe `transformation` représentant une transformation d'image et modifiez en conséquence les classes `flou` et `negatif`. Ajoutez et écrivez la méthode `transformeImage(tr)` qui transforme l'image avec la transformation `tr`. Modifiez en conséquence les méthodes `transformeImageFlou()` et `transformeImageNégatif()`.

#### *Ré-appliquer la dernière transformation*

On veut maintenant pouvoir ré-appliquer la dernière transformation effectuée en appelant la méthode `appliqueDerniereTransformation()`. Que faut-il faire pour cela, modifiez `ngimp` en conséquence.

Si on le fait naïvement cela pose un problème : le programme plante, pourquoi ?

Effectuez les modifications nécessaires pour résoudre le problème et faire attention à ce qu'il n'y ait pas de fuites de mémoires (utilisez les pointeurs intelligents).

Que se passe-t-il si on veut ré-appliquer la dernière transformation avant d'en avoir appliqué une ? Comment initialiser la dernière transformation pour corriger ce problème ?

#### *Transformation composée*

Certaines transformations d'images sont en fait l'application consécutive de plusieurs transformations. Écrivez la classe `transformationComposee` représentant une transformation composée de plusieurs transformations.