

# Compléments en Programmation Orientée Objet

## TP n° 5 : Adaptateurs

### 1 Adaptateur : Visualiseur d'images

#### Exercice 1 : Adaptateur

Vous prenez en charge un projet de développement d'un visualiseur d'images. La version actuelle du logiciel permet de visualiser des images matricielles dans un format spécial que nous nommerons 'RGB', et les fichiers de ce format ont l'extension '.rgb'. Une image dans ce format peut être considérée comme une matrice de couleurs RGB. Une couleur RGB est un triplet de trois entiers compris entre 0 et 255, représentant respectivement le rouge (red), le vert (green) et le bleu (blue).

Les fichiers fournis `RGBColor.java`, `RGBImageCanvas.java`, `RGBImageReader.java` and `RGBImageViewer.java` constituent la version actuelle. L'exécution des commandes suivantes

```
javac RGBColor.java RGBIm*.java
java RGBImageViewer ./cat.rgb
```

Compilera le programme, exécutera le visualiseur, et affichera l'image d'un chat qui est fournie dans le fichier.

Votre objectif est d'étendre ce visualisateur pour prendre en charge l'affichage des images dans un autre format que nous nommerons 'RGBA'. Les fichiers de ce format auront l'extension '.rgba'. La principale différence par rapport au format RGB est que les couleurs ne sont plus simplement des triplets de rouge, vert et bleu, mais elles incluent également un quatrième paramètre, l'alpha, qui permet de décrire l'opacité ou la transparence du pixel. Cependant, le code de la version actuelle est stable et utilisé par de nombreux autres développeurs, c'est pourquoi nous ne souhaitons pas le modifier. Pour ce faire, nous allons utiliser le patron adaptateur (adapter pattern).

1. En réutilisant le contenu de fichier `RGBColor.java`, définir la classe `RGBAColor` où vous ajoutez la composante alpha (une entier compris entre 0 et 255 où 0 représente un pixel totalement transparent et 255 signifie l'opposé).
2. De manière similaire, créez une classe `RGBAImage` représentant des images avec des couleurs RGBA.
3. La classe `RGBImageViewer` est le point d'entrée de notre visualisateur, prenant en paramètre une image `RGBImage`. Nous n'avons pas l'intention de modifier cette classe ni aucune des autres classes sur lesquelles elle dépend. Afin de résoudre ce défi, écrivez une classe adaptateur qui permettra de transmettre une image `RGBAImage` au visualisateur tout en ignorant le canal alpha des pixels. Ce faisant, l'adaptateur devra être utilisable comme s'il s'agissait d'une `RGBImage` avec le visualiseur.
4. En utilisant la classe que vous avez écrit dans la question précédente et la classe `RGBAImageReader` fournit avec le TP, vous seriez capable d'écrire la classe `RGBAImageViewer` avec seulement une méthode statique `main` qui permet d'afficher l'image RGBA passé comme argument que vous pouvez tester avec le fichier `dices.rgb` fourni avec le TP.