

Method Factory, Factory, Abstract Factory

Exercice 1 - Factories en tout genre

Le but de cet exercice est de créer un petit programme permettant d'appliquer une série de filtres à une série d'images.

Commencer par télécharger le fichier `td1convert.zip` (`td1convert.zip`) . Puis importer-le dans le répertoire source de votre projet courant dans Eclipse.

Pour cela, importez l'archive dans Eclipse. avec clic droit sur votre répertoire contenant les sources sélectionnez "import" puis "archive file". Indiquez alors le fichier téléchargé.

Au besoin, faites un clic droit sur le paquetage importé, puis sélectionnez "refactor" pour donner un nom qui correspond à l'arborescence voulue.

Voici un exemple de fichier (`filter.txt`) décrivant une succession de filtres à appliquer sur une image :

```
gray
rotate
blur
```

Avec la ligne de commande suivante :

```
java Convert filter.txt image.gif
```

Le programme va appliquer l'ensemble des filtres décrit dans le fichier `filter.txt` à l'image `image.gif` et ressortir le résultat sous forme d'une image PNG nommée `image.gif.png`.

- 1 Pourquoi les constructeurs de `Images` et de `ImageOps` sont déclarés `private` ?
- 2 Proposer une solution de conception permettant, sans changer les classes existantes, de pouvoir appliquer indifféremment n'importe quel filtre à une image.
Coder la solution proposée.
- 3 Utiliser le plugin Omondo pour visualiser le design que vous venez de créer. Pour cela, sur la "vue explorer", sur le paquetage concerné, utiliser le bouton droit puis "Open UML" / "class diagram UML". Dans la nouvelle fenêtre cochez "Inheritance", et valider. Assurez vous que toutes les classes sont sélectionnées et validez.
Vous pouvez maintenant comparer la représentation UML avec le code Java.
- 4 Dans le but d'associer un filtre à une chaîne de caractères, écrire une méthode `createFilter(String name)` qui renvoie un filtre en fonction de son nom.
Dans quelle classe doit-on mettre cette méthode ? Cette méthode doit-elle ou non être statique ?
Comment s'appelle ce design pattern ?
- 5 Remarquer que les objets filtres sont indépendants de l'image qu'il manipule. Il est donc possible de renvoyer plusieurs fois le même filtre pour une même opération.
Changer votre code en correspondance.
- 6 Comment faire pour éviter les `if .. else` dans le code de la méthode `createFilter(String name)` ?
Changer votre code en conséquence.
- 7 Écrire le code qui prend en paramètre un fichier et ressort sous forme d'une liste

- l'ensemble des filtres à appliquer, vous utiliserez pour cela la classe `java.util.Scanner`.
- 8 On souhaite maintenant pouvoir choisir dans le code précédent la façon dont les filtres sont implantés, que doit-on faire ?
Quel est le design pattern correspondant ?
Modifier votre code en conséquence.
 - 9 Écrire la classe `Convert`.

Exercice 2 - Chemin dans un système de fichier

Voici le code de la classe `PathName` décrivant un chemin dans l'arborescence de fichier.

```
package fr.uml.vgl.td1.pathname;

public class PathName {
    public PathName(PathName directory, String name) {
        this.directory=directory;
        this.name=name;
    }
    public PathName getDirectory() {
        return directory;
    }
    public String getLocalName() {
        return name;
    }
    public String getPathName() {
        return directory.getPathName()+ '/' +name;
    }

    private final PathName directory;
    private final String name;
}
```

- 1 On souhaite ajouter un moyen de créer `PathName` qui prend une chaîne de caractère correspondant à un chemin dans le système de fichier et qui découpe celui-ci en `PathName`.
Comment doit-on faire ?
Rappel : pour extraire le dernier slash d'une chaîne de caractère, on peut utiliser `lastIndexOf`.
- 2 On souhaite maintenant être indépendant de l'OS et donc que cela marche pour Windows où le séparateur de fichier est `\` pour Unix où le séparateur de fichier est `/`.
Utiliser pour savoir sur quel système le code est exécuté la propriété `system os.name` et pour l'obtenir la méthode `System.getProperty()`.
Changer votre code pour être indépendant de l'OS.
- 3 On souhaite maintenant pouvoir gérer MacOS qui possède un double système de fichier, cela doit marcher avec le séparateur `/` ou le séparateur `:`.
Changer votre architecture en introduisant une nouvelle classe `FileSystemTokenizer` qui s'occupera de la gestion des séparateurs.
Quelles sont les méthodes de cette classe.
- 4 Comment garantir que l'on aura qu'un seul représentant de cet objet ?

Exercice 3 - Filtre avec argument

En reprenant l'exercice sur les filtres, on souhaite maintenant permettre que les filtres

puissent prendre des paramètres :

```
gray  
rotate 1,5  
blur
```

Attention, tous les changements effectués doivent permettre que les anciens fichiers de filtrage continuent à fonctionner.

- 1 Quelles parties du code précédent doit-on changer pour prendre en compte le changement de la spécification du fichier des filtres ?
Attention, petit piège, il y a plusieurs parties qui changent !
- 2 Quel design-pattern doit-on utiliser pour que le parseur du fichier des filtres délègue à chaque créateur de filtre le soin de lire les arguments du filtre correspondant ?
- 3 Modéliser avec un diagramme de séquence les échanges entre les différents objets
Implanter ce design-pattern.