

Le patron de conception Fabrique abstraite

Source « <https://www.math.univ-paris13.fr/~chaussar/> »

Exercice 1

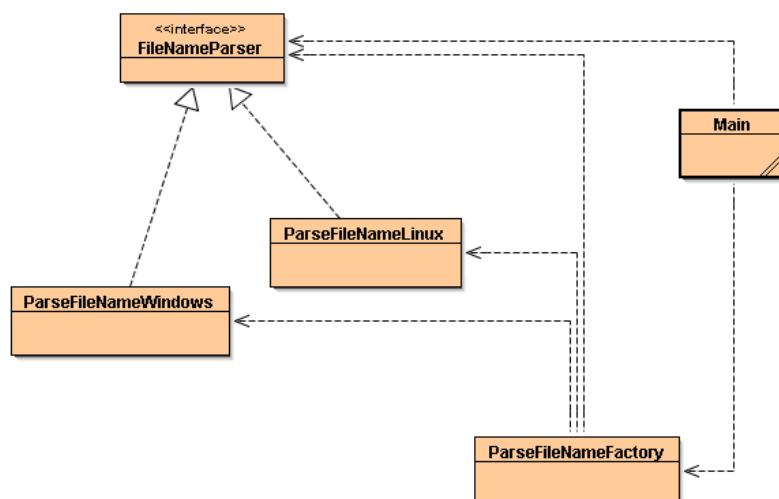
Le code fourni permet, étant donné un chemin complet vers un fichier Windows, d'afficher juste le nom du fichier. Voici le code de la seule classe ce projet :

```
public class Main {  
    public static void main_parse_filename(String path) {  
        //index est l'endroit où se situe, dans la String path, la dernière  
        //apparition du caractère \  
        int index = path.lastIndexOf("\\");  
        //On construit une String qui ne contient que la partie située à droite  
        //du dernier caractère \  
        String r = path.substring(index+1);  
        System.out.println(r);  
    }  
}
```

Si vous testez ce programme en lui passant en paramètre "C:\\Windows\\hello.dll", il devrait vous renvoyer "hello.dll".

Exercice 1

On souhaite ajouter la possibilité de trouver, étant donné un chemin vers un fichier Linux (de type "/user/share/hello.rc"), le nom du fichier en question. Pour ce faire, on procède comme à la question précédente : on crée une interface `FileNameParser` et deux sous-classes `ParseFileNameWindows` et `ParseFileNameLinux` qui l'implémentent. Ensuite, on crée une `Factory` qui permettra de créer des `FileParser` dans notre main. Le diagramme de classe devra ressembler à ceci :



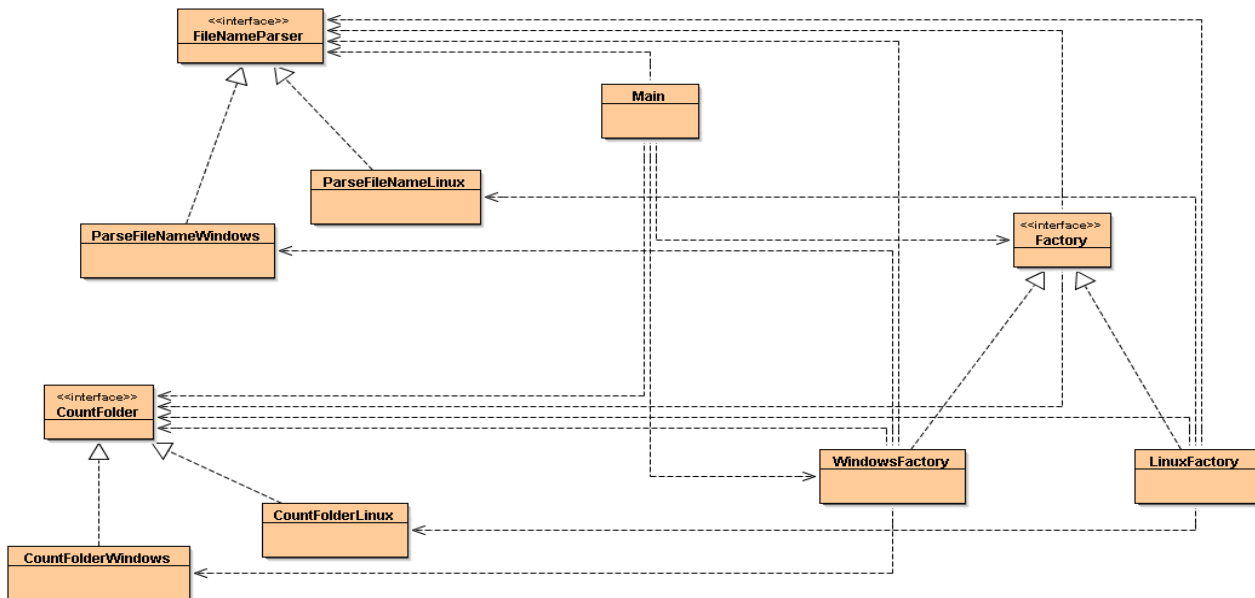
Exercice 2

On souhaite maintenant ajouter des fonctionnalités. On veut maintenant compter le nombre de dossiers qu'il faut parcourir depuis la racine pour trouver notre fichier. On va ajouter une interface `CountFolders`, et deux sous classes qui l'implémenteront (une windows, et une linux). Ensuite, il va falloir dire à la `Factory` de créer à chaque fois deux types d'objets (un `FileNameParser` et un `CountFolders`) qui seront rangés dans des variables de classe de la `fFactory`.

Exercice 3

Le problème de cette implémentation est que, si on veut encore ajouter une fonctionnalité, il faudra aller changer le code de la fabrique. On voudrait en fait, si quelqu'un veut ajouter des fonctionnalités, ne pas le forcer à ouvrir notre fabrique et avoir à comprendre tout ce qui s'y passe pour savoir où ranger son code.

On va donc faire de notre fabrique une interface et créer deux sous fabriques qui l'implémentent (une `Linux` et une `Windows`). Il faudra cependant, dans la classe `Main`, choisir explicitement quelle fabrique utiliser. Le diagramme de classe devrait ressembler à ceci :



Exercice 4

Pour éviter que, dans la classe `Main`, on ait à choisir explicitement quelle fabrique implémenter, et pour éviter que le `Main` ait connaissance de toutes nos différentes fabriques, nous allons faire une fabrique abstraite.

Exercice 5

On veut ajouter le support des chemins Macintosh (reconnaître les « / » ou les « : »).
Faîtes les modifications nécessaires.

Exercice 6

On souhaite ajouter la fonctionnalité de donner le nom du répertoire source qui contient
notre fichier. Faîtes les modifications nécessaires.