

Exemple de Factory :

Objets préliminaires

Dans le package **objets**, on crée :

Une interface ILister qui contient

```
public List<String> list() throws Exception ;
```

```
public void display() ;
```

2 classes qui implémentent ILister, FileLister et MemLister. Ces deux classes sont dans le package **objets**. Dans un premier temps, list() renvoie null.

Dans le package **main**, on crée une classe Test avec juste un main() dedans.

La Factory

On va créer dans le package **process** une classe ListerFactory. Cette classe contient une fonction static dont la signature est la suivante :

```
ILister newInstance() ;
```

Comme son nom l'indique, cette classe crée un objet d'interface ILister (en pratique, soit un FileLister, soit un MemLister).

Pour déterminer lequel des deux est créé, on utilise un fichier de configuration lister.cfg qui contient une seule entrée :

```
class=objets.FileLister
```

ou

```
class=objets.MemLister.
```

Premier Test

Dans le main(), on invoque ListerFactory.newInstance() pour récupérer un ILister et on appelle display().

Changer la valeur de class dans le fichier de configuration et vérifier que l'affichage change.

Deuxième Test

On implémente la fonction `list()` dans `MemLister` et `FileLister`.

Dans `MemLister` : on renvoie une `List` avec des valeurs qu'on a rentré en dur dans la classe.

Dans `FileLister` : on parcourt un fichier texte (au choix) et on renvoie une `List` des lignes de ce fichier.

Dans le `main`, on appelle `list()`. Vérifier que le comportement change en fonction de ce qu'il y a dans le fichier de configuration.

Exemple instanceof

Juste après l'appel de `ListerFactory.newInstance()` dans le `main`, on teste avec `instanceof` le type retourné par `ListerFactory.newInstance()` et on affiche quel est ce type.

Dernier exemple

On rajoute une classe `NetLister` implémentant `ILister` et qui lit sur le réseau.

Ajout d'une façade pour écrire sur la console

Créer une interface `IDisplay` avec la method `display(String s)` ;

Créer une classe `ConsoleDisplay` qui implémente `IDisplay`. Dans sa fonction `display`, écrire sur la console.

Créer une `DisplayFactory` qui utilise `lister.cfg` pour retourner un objet `IDisplay` (on ajoute évidemment une entrée dans le fichier de configuration). Remplacer dans toutes les méthodes `System.out.println` par la fonction `display` de `IDisplay`.

Vérifier que tout marche.