Université Paris Cité

Compléments en Programmation Orientée Objet
TP n° 4 : inversion de dépendance, fabriques abstraites, adaptateurs

1 Programmer à l'interface

Finir le TP précédent.

2 Inversion de dépendance

Exercice 1 :

Téléchargez le zip de l'exercice sur Moodle et décompressez le dans votre dossier de travail.
Vous trouverez une bibliothèque déjà programmée dans le sous-dossier/package `stats`.
Écrivez une classe contenant une méthode `main` utilisant cette bibliothèque. Pour cela vous devrez implémenter l'interface `DataSeries` à l'aide de listes (`java.util.List`) au travers d'un adaptateur.
Dans votre `main`, vous afficherez la moyenne et l'écart type de la liste `[64.51, 138.89, -25.5, 22.87]`.

Exercice 2 :

Téléchargez le zip de l'exercice sur Moodle et décompressez le dans votre dossier de travail.
Vous trouverez un programme dans le sous-dossier/package `client`.
Il s'agit d'un programme exécutable qui ne fonctionne pas en l'état, car il dépend de la bibliothèque du package `logger` qui n'est pas encore programmée. Pour cause : ce sera à vous de le faire !
Implémentez la classe `logger.Logger` et les interfaces `logger.LogEntry` et `logger.LogEntryAbstractFactory` de la bibliothèque `logger` afin que le `main` de `client.main` fonctionne comme prévu, c'est à dire comme dans l'exemple d'éxécution ci-dessous :

```
> treu
Réessayer !
> true
> 56
> 6546
> print
true,
56,
6546,
> prettyprint
Booléen : true;
-------------
Entier : 56;
-------------
Entier : 6546;
-------------
>

Process finished with exit code 0
```

Explication : la classe logger stocke chaque valeur fournie par l'utilisateur dans le `main` dans une nouvelle instance appropriée de `LogEntry`, qui est aussitôt ajoutée au journal (une liste).
Quand l'utilisateur demande un affichage du journal, l'instance de `Logger` lit la liste et demande l'affichage approprié (simple ou mis en page) pour chaque entrée sauvegardée.