Inversion de contrôle

L'objectif de ces exercices est de comprendre le principe de l'inversion se contrôle et de l'appliquer en Java et en Python. Nous partons d'un problème déjà résolu avec le programme correspondant écrit en Java et Python. Nous envisageons (et réalisons) quelques extensions afin de voir les limites de la solution fournie et aller vers une meilleure architecture.

Exercice 1 : Point de départ
Nous partons de la classe Analyseur qui analyse les données d'un fichier texte. Elle calcule, par position (couple défini par une abscisse x et une ordonnée y), la somme des valeurs associées.

La classe est fournie en Java et en Python.

Exécuter les programmes pour vérifier qu'ils donnent les résultats attendus.

Exercice 2 : Évolution : Plusieurs fichiers
On peut pouvoir charger plusieurs fichiers dans un même analyseur. Bien sûr, ces différents fichiers auront des noms différents.

Modifier la classe Java Analyseur pour permettre au programme principal de charger plusieurs fichiers.

Exercice 3 : Évolution : Vérification du fichier
Le fichier ne devrait contenir que des valeurs positives ou nulles. Une valeur négative signifie que le fichier est incorrect. Dans ce cas, une exception `MalformedFileException` sera levée et aucune des données qu'il contient ne sera comptabilisée.

Modifier la classe Java Analyseur.

Exercice 4 : Évolution : Autre format de fichier
On veut traiter un autre type de fichier. Chaque ligne contient, dans l'ordre, séparés par des blancs, un identifiant (ignoré), une abscisse (entier), une ordonnée (entier), un texte (ignoré) et, enfin, une valeur (réel). Ici, ce sont les fichiers avec le suffixe `-f2.txt` mais on pourrait les nommer autrement.

On veut pouvoir charger les deux formats de fichier (l'initial et le nouveau).

Modifier la classe Java Analyseur en conséquence.

Exercice 5 : Nouvelle architecture
On pourrait envisager d'autres évolutions :

1. On veut pouvoir lire les données de fichiers csv, XML, etc.
2. On veut pouvoir demander les données à l'utilisateur.
3. ...

1. Indiquer les défauts de l'architecture initiale.
2. Proposer une nouvelle architecture de l'application.

3. Mettre en œuvre cette architecture en Java et en Python sans intégrer, dans un premier temps, les extensions envisagées dans cet exercice. Pour la version Python, on ajoutera la possibilité de lire un fichier au format csv en utilisant la bibliothèque Python fournie par le module csv.