Test et conception objet

Toutes les questions demandant une réponse en français et non en code sont à rendre dans un `README`.

Exercice 1 : Filtre de fichiers de log

On souhaite écrire et tester un filtre de fichier de log. Un fichier de log contient un message de log par ligne. Un message de log est composé de 3 informations :

- une date au format `yyyy-mm-dd`
- une priorité (un entier entre 1 et 10)
- le contenu du message : du texte ne contenant pas de virgule

Les différentes informations sont séparées par des virgules. Chaque ligne est terminée par un retour-chariot. On aura par exemple le fichier :

```
2010-02-25, 5, error in database
2010-02-25, 10 , system crash
2010-02-26, 2, error
```

Le but du filtre est de :

- lire les messages d'un fichier de log
- filtrer les messages selon une condition donnée : le filtre conserve les messages dont la priorité est supérieure ou égale à un entier n compris entre 0 et 10 ;
- écrire les messages conservés dans un nouveau fichier de log

Dans l'ex ci-dessus, pour $n = 5$, le filtre garde les 2 premiers messages et les écrit dans un fichier de nom donné.

Le but du TP est d'appliquer le SRP et l'injection des dépendances pour obtenir des tests simples et maintenables. On commence par la classe qui représente le filtre. Après / avant / pendant le développement, pour vérifier vous-même que vous avez une architecture qui tient la route, demandez-vous quel impact a sur vos tests le changement du format des messages (ex : ajouter ou retirer un champ), et quel impact a sur vos tests un changement de la condition de filtrage (par ex, prendre aussi en compte la date du message). Si ces changements ont un impact, c'est que la responsabilité de votre classe est trop grande.

Q 1.1 : (à faire avant ou après la question suivante) : quelle responsabilité a votre classe qui représente le filtre ? Quelles responsabilités ont les autres classes ? □

Q 1.2 : Quels cas tester pour la classe qui représente le filtre ? □

Q 1.3 : Écrire en TDD la classe qui représente le filtre. □

On veut maintenant tester et développer en TDD la classe qui transforme l'entrée textuelle en flot de messages.

Q 1.4 : Allez-vous construire des messages en dur dans le code ? Quel pattern allez-vous utiliser ? □

Q 1. 5 : Que pensez-vous d'un test automatisé qui irait lire des fichiers de test (ou attendrait des données au clavier) ? Regardez la fiche « tester avec des E/S » pour paramétrer votre classe par un flot d'entrée. □

Q 1.6 : Quels cas proposez-vous de tester ? Penser à ne pas traiter uniquement le cas nominal. □

Q 1.7 : Tester et développer en TDD la classe qui extrait les messages d'un flot d'entrée. Les cas d'erreur syntaxique ou lexicale étant nombreux et fastidieux à écrire (c'est là qu'on aimerait une bibliothèque de tests paramétrés bien faite !), il suffit d'en écrire un échantillon représentatif. □

Q 1.8 : **Facultatif** Si vous n'êtes pas convaincu par l'approche ajouter un test qui lit vraiment un flot d'entrée issu d'un fichier. □