

## TP5 : Introduction à la traçabilité logicielle avec Spoon

### Exercice 1 : Logging avec Spoon pour le profilage

#### Question 1

Créez une application simple (avec une CLI ou une GUI) qui vous permet de :

1. créer un utilisateur avec un ID, un nom, un âge, une adresse électronique et un mot de passe.
2. fournir à l'utilisateur un menu qui lui permet de :
  - afficher les produits dans un référentiel, où chaque produit a un ID, un nom, un prix et une date d'expiration.
  - Récupérer un produit par son ID (si aucun produit avec l'ID fourni n'existe, une exception doit être levée).
  - ajouter un nouveau produit (si un produit avec le même ID existe déjà, une exception doit être levée).
  - supprimer un produit par son ID (si aucun produit avec l'ID fourni n'existe, une exception doit être levée).
  - mettre à jour les informations d'un produit (si aucun produit avec l'ID fourni n'existe, une exception doit être levée) .

#### Question 2

1. regardez les vidéos suivantes :
  - [Journalisation en Java](#)
  - [Logback vs SLF4J vs Log4J2 - quelle est la différence ? Cerveaux Java Cerveaux Octets](#)
  - [Comment faire de la journalisation dans Spring Boot - Brain Bytes](#)
2. Choisissez un utilitaire de journalisation parmi ceux vus dans les vidéos et jouez un peu avec lui pour comprendre ses différentes facettes. **Note :** L'utilisation de Log4j2 ou SLF4J est fortement recommandée.

#### Question 3

Utilisez Eclipse JDT ou Spoon (recommandé) pour tracer le code de votre application en utilisant l'utilitaire de journalisation choisi dans la dernière question, de sorte que les journaux générés puissent être exploités pour créer des profils d'utilisateurs comme suit :

- un profil pour ceux qui ont principalement effectué des opérations de lecture sur votre dépôt .
- un profil pour ceux qui ont principalement effectué des opérations d'écriture sur votre dépôt.
- un profil pour ceux qui ont recherché les produits les plus chers dans votre référentiel.

La définition de la structure de votre profil vous appartient. Les informations les plus importantes à inclure sont les informations sur l'utilisateur et les caractéristiques de marquage de votre profil.

(par exemple, les opérations de lecture/écriture effectuées par un utilisateur pour les profils de lecture/écriture les plus utilisés). Vous pouvez également choisir le format de stockage de vos profils et des données qui leur sont associées. **Remarque** : l'utilisation de JSON est fortement recommandée en raison de ses qualités de légèreté et de sa facilité de sérialisation/désérialisation vers/depuis une entité de persistance (par exemple, un système de fichiers, une base de données, etc.)

#### Question 4

Définissez et exécutez une séquence de scénarios d'exécution avec différents utilisateurs pour générer vos journaux. Par exemple, vous pouvez créer 10 utilisateurs, et laisser chaque utilisateur exécuter environ 20 scénarios différents impliquant les opérations ci-dessus avec différentes valeurs d'entrée. Assurez-vous que vos scénarios sont suffisamment diversifiés pour simuler une expérience réelle afin d'avoir des profils correctement élaborés à la fin.

#### Question 5

Proposer un moyen d'analyser les logs générés et d'extraire les informations nécessaires à la construction de vos profils utilisateurs. Notez que les logs structurés nécessitent la réification d'un LPS et la définition de ses mécanismes de construction et d'impression (par exemple, Timestamp : <date\_time>, Event : <event\_info>+, User : <user\_info>+, Action : <method\_info>+, etc.) Dans ce cas, chaque partie du contenu du LPS peut être construite séparément et ensuite agrégée pour former le LPS résultant (*pensez au Builder Design Pattern*). Vous pouvez également choisir de les stocker et de les afficher sous forme de fichiers JSON si vous le souhaitez.

### Exercice 2 : Logging avec Spoon pour le débogage (*Bonus*)

#### Question 1

Utilisez Eclipse JDT ou Spoon (recommandé) pour tracer le code de votre application précédemment créée en utilisant l'utilitaire de journalisation de votre choix, de sorte que les journaux générés puissent être exploités pour identifier les erreurs les plus courantes (*lancées et capturées*). Veillez à distinguer les traces générées lorsqu'une erreur est lancée de celles générées lorsqu'elle est capturée.

#### Question 2

Définissez et exécutez une séquence de scénarios d'exécution avec différents utilisateurs pour générer vos journaux d'erreurs. Veillez à ce que vos scénarios soient suffisamment diversifiés pour simuler une expérience réelle.

#### Question 3

Proposez un moyen d'analyser les journaux générés et d'en extraire les

erreurs les plus courantes, classées dans l'ordre décroissant de leur fréquence d'apparition. Assurez-vous d'indiquer quel utilisateur et quelle action a causé l'erreur et pour quelles valeurs d'entrée.