

TP 01 - Nettoyage

Objectifs du TP

Examiner du code écrit par d'autres développeurs et nettoyer/améliorer le code.

Description :

- 1) Faites un fork du projet suivant :
<https://github.com/DiginamicFormation/code-propre.git>
- 2) Une fois forké :
 - a. Si vous êtes sur STS, clonez le dans votre workspace puis importez-le en tant que projet maven
 - b. Si vous êtes sur IntelliJ, créez un nouveau projet en saisissant directement l'URL de GitHub (File -> New -> Project from version control)
- 3) **Exercice 1 :**
 - a. Corrigez les classes en respectant les règles de nommage puis renseignez la **javadoc**.
- 4) **Exercice 2 :**
 - a. Dans cet exercice, vous êtes lead developer et un développeur de votre équipe était en charge de développer les classes **CompteBancaire** et **LivretA** en respectant les consignes suivantes :
 - i. Un compte bancaire a un solde et un montant de découvert autorisé
 - ii. Un livret A a les mêmes attributs avec en plus un taux de rémunération annuel. Par ailleurs un livret A ne peut jamais être à découvert.
 - iii. La méthode **ajouterMontant** permet de créditer un montant sur le compte.
 - iv. La méthode **débiterMontant** permet de débiter un montant sur le compte. teste le type de compte.
 1. S'il s'agit d'un compte courant, on peut débiter de l'argent même s'il ne reste pas assez d'argent sur le compte. Attention cependant, il n'est pas possible de dépasser le découvert autorisé.
 2. S'il s'agit d'un livret A on ne peut pas retirer plus qu'il ne reste sur le compte. Les découverts ne sont pas autorisés.
 - v. La méthode **appliquerRemunerationAnnuelle** permet de calculer la rémunération du livret A
 - b. **Identifiez** les problèmes de conception et **corrigez-les**.
- 5) **Exercice 3 :**
 - a. Dans cet exercice, une personne est en charge de développer une application permettant de gérer un Zoo mais il n'est pas convaincu par la décision prise. Il a écrit une classe Zoo mais il vous demande conseil.
 - b. Faites-lui remarquer que son code comporte un contexte flou. Vous l'aidez à corriger ce problème.

6) Exercice 4 :

- a. Dans cet exercice un développeur a écrit une classe DateUtils qui a de la duplication de code.
- b. Re-écrivez les méthodes afin de supprimer la duplication de code.

7) Exercice 5 :

- a. Dans ce dernier cas de figure, un peu plus difficile, on souhaite aider une personne à réorganiser son code.
- b. L'idée dans ce cas est de supprimer les if/else. Comment ? en mettant en place une boucle et en « demandant » à la classe Caisse si elle accepte ou non un item.
 - i. Dès que vous commencez à avoir des if/else dans du code, on peut souvent : soit mettre en place de l'héritage ou une indirection.