

TRAVAUX DIRIGES

(PLATEFORMES ET OUTILS DE DEVELOPPEMENT)

I. Single Responsibility Principle (SRP)

1. Ecrire une classe Book avec comme attributs numOfPages (int) et authorName (String)
2. Réécrire la méthode toString() permettant de retourner l'auteur séparé du nombre de pages par un tiret.
3. Ecrire une méthode qui permet d'enregistrer et d'imprimer un objet de type Book
4. Ecrire une autre (d'autres) classe(s) ou interface(s) permettant à la classe Book de respecter le principe de responsabilité unique.

II. Open/Closed Principle (OCP)

On veut écrire une application permettant de trier automatiquement une série de nombres à l'aide d'un type de tri que l'utilisateur aura choisi (tri par fusion ou tri par insertion)

Travail à faire

1. Ecrire en Java un type énuméré *TypeTri* qui contiendra les valeurs suivantes *TRIFUSION*, *TRIINSERTION*
2. Ecrire une classe abstraite nommée *Trieuse*. Elle contiendra un attribut *type* de type *TypeTri* et une méthode abstraite *tri()*.
3. Ecrire les deux classes *TriInsertion* et *TriFusion* qui héritent de *Trieuse*.
4. Ecrire une classe *GestionnaireTri*. Cette classe contient une méthode *tri* (*Trieuse trieuse*) permettant de faire l'un des deux types de tri.
5. Ecrire une classe comportant une méthode main permettant de tester tout cela.
6. Sur la base de vos connaissances, dire pourquoi l'OCP n'a pas été respecté ? Justifier
7. Proposer une amélioration permettant de ne pas violer l'OCP.

III. Liskov Substitution Principle (LSP)

Ecrire en Java deux classes Voiture et VoitureElectrique qui implémentent une (ou deux) interface(s) en respectant le principe de substitution de Liskov. L'implémentation vient du fait qu'elles sont toutes deux des véhicules.

Ecrire une classe comportant une méthode main permettant de tester tout cela.

IV. Interface Segregation Principle (ISP)

Ecrire en Java une (des) interface(s) représentant la structure de données « liste chaînée » puis écrire les classes ListeSimplementChaínee, ListeDoublementChaínee, ListeCirculaire implémentant les interfaces écrites plus haut.

V. Dependency Inversion Principle (DRP)

Ecrire une classe MySqlConnection contenant les méthodes connecter() et deconnecter() permettant d'établir une connexion avec une base de données MySQL. Ecrire une autre classe GestionnaireDb permettant d'utiliser les méthodes de la classe MySqlConnection pour interagir avec une base de données MySQL.

On souhaite ensuite changer de SGBD et utiliser plutôt Oracle. Faire les ajustements nécessaires. Dire si votre première conception obéit à l'OCP puis dire si elle obéit au DIP. Sinon, proposer une autre conception qui obéira à ces deux principes.