

TD Patron

Pour ce TD, veuillez installer IntelliJ IDEA (<https://www.jetbrains.com/fr-fr/idea/>) pour effectuer des implémentations des patrons de conception qu'on a vu pendant le cours.

1. Identifiez le patron concerné par ce programme :

App.java :

```
public class App {  
    public static void main(String[] args) {  
        Connection c1 = SConnection.getConnection();  
        Connection c2 = SConnection.getConnection();  
        Connection c3 = SConnection.getConnection();  
    }  
}
```

Connection.java :

```
public class Connection {  
    public Connection(){  
        System.out.println("Création d'une connection");  
    }  
}
```

Implémentez ce patron pour que ce programme fonctionne. D'après vous, « Création d'une connection » est affiché combien de fois quand on l'exécute ? Qu'est-ce que cela signifie ?

2. Implémentez le patron Observer. Pour cela, créez un package observable (diffuseur) qui contient une interface Observable et son implémentation ObservableImpl, et un package observers qui contient une interface Observer et ses implémentations ObserverImpl1 et ObserverImpl2. Enfin, créer une classe TestObserver pour tester vos implémentations.

3. Identifiez le patron concerné par ce programme :

```
public class App {  
    public static void main(String[] args) {  
        Context context = new Context();  
        context.appliquerStrategy();  
        context.setStrategy(new StrategyImpl1());  
        context.appliquerStrategy();  
  
        context.setStrategy(new StrategyImpl2());  
        context.appliquerStrategy();  
  
        context.setStrategy(new StrategyImpl3());  
        context.appliquerStrategy();  
    }  
}
```

Implémenter ce patron pour que ce programme fonctionne.

4. Un dossier peut contenir des fichiers ou des dossiers, comme illustré ici :

Folder :Design Patterns level 0

Folder :Comportement level 1

File : Strategy=> level2

- File : Observer=> level2
- Folder :Structure level 1
 - File : Decorator=> level2
 - File : Composite=> level2
- Folder :Création level 1
 - File : Singleton=> level2

Quel patron peut-on utiliser pour le modéliser ? Implémenter ce patron pour obtenir le même affichage.

5. Défi : Implémenter le patron Visitor pour représenter les différents façons de calculer les taxes de différents produits.