**Prosit 2 : Ce n’est pas un ouvrage de BTP**

# Contexte :

2F Roaming essaie d’adopter des bonnes pratiques de développement afin de minimiser les couts.

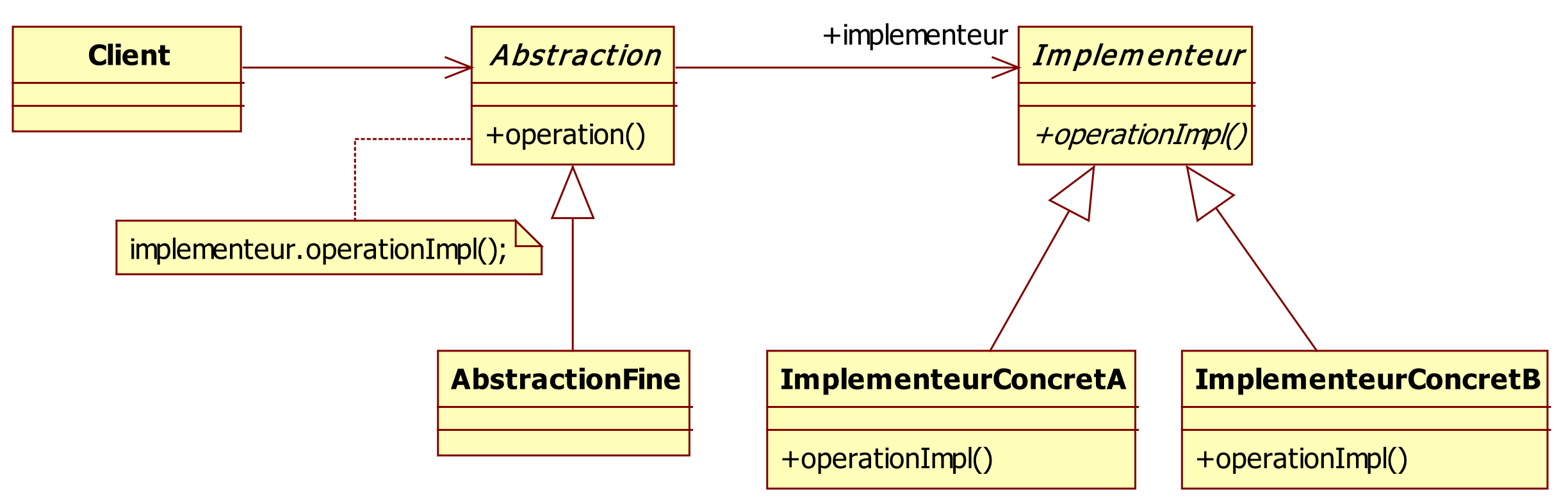
# Mot clés :

Design Pattern (DP)

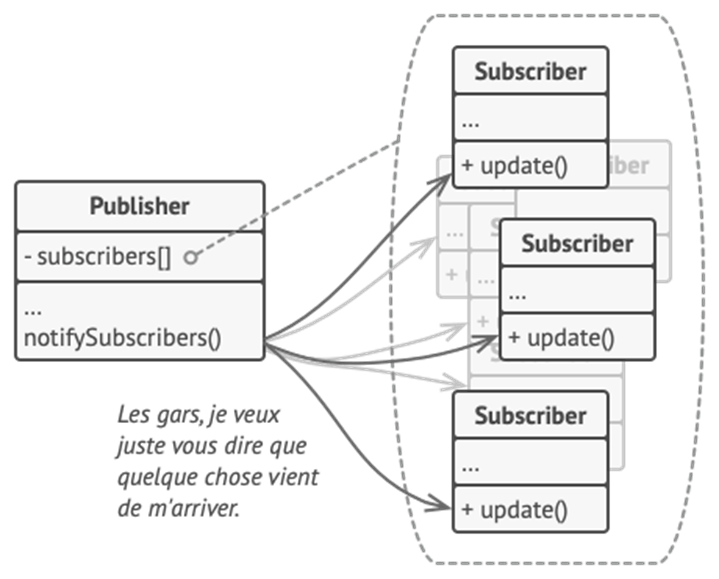
Singleton : Le Singleton est un Design Pattern qui garantit qu'une classe n'a qu'une seule instance et fournit un point d'accès global à cette instance. Il est souvent utilisé pour gérer des ressources partagées, comme des connexions à une base de données.

Usine : Le Factory Pattern est un Design Pattern de création qui fournit une interface pour créer des objets dans une super-classe, mais permet aux sous-classes de modifier le type d'objets qui seront créés. Il est utilisé pour promouvoir la modularité et la réutilisabilité du code

Pont: Le DP Bridge (Pont) sépare une abstraction de son implémentation afin que les deux puissent évoluer indépendamment. Cela permet de réduire la complexité en évitant une liaison permanente entre une abstraction et son implémentation.



Observateur : Le design pattern Observateur (Observer) est un modèle de comportement qui définit une dépendance entre objets de manière que lorsque l'état d'un objet change, tous ses dépendants soient notifiés et mis à jour automatiquement.



Stratégie : est prévu pour fournir le moyen de définir une famille d'algorithmes, encapsuler chacun d'eux en tant qu'objet, et les rendre interchangeables. Ce patron laisse les algorithmes changer indépendamment des clients qui les emploient.

Décorateur : est un design pattern structurel qui permet d'attacher de manière dynamique des responsabilités supplémentaires à un objet. Il est utilisé pour étendre ou modifier le comportement d'objets individuels sans altérer leur structure.

ROO : Reusable objet-oriented : Ce terme fait référence à la conception et à la programmation orientées objet qui encouragent la réutilisation des composants logiciels. L'objectif est de créer des objets qui peuvent être facilement réutilisés et adaptés pour différentes applications.

# Problématique :

**Quels sont les concepts à s’approprier pour diminuer les couts de développements ?**

**Quels sont les types de Design Pattern et lequel utiliser ?**

# Contraintes :

DP orienté C#

# Généralisation :

Maitriser les DP

# Livrables :

Coder les 6 DP en C#

Singleton Hana

Usine Roméo

Pont Hugo

Observateur Emile

Stratégie Florian

Décorateur Stéphanie

# Pistes de solution :

Se servir des DPs pour trouver des solutions aux problèmes récurrent

Singleton, Usine, pont, Observateur, Stratégie et décorateur

# Plan d’action :

Faire la corbeille :

Recherche des DP :

# Réalisation du plan d’action :

Texte