**TD n°1 Design Patterns**

**Factory, AbstractFactory et Singleton**

**But du TD**

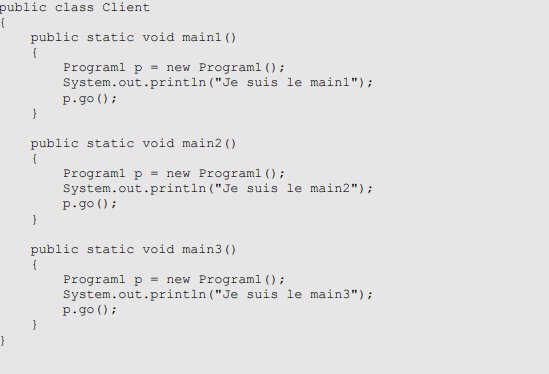
Comprendre les patterns de création par leurs mise en pratique

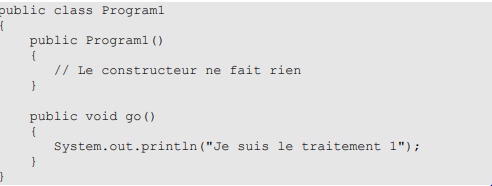
langage de développement: java

**Exercice n°1:**

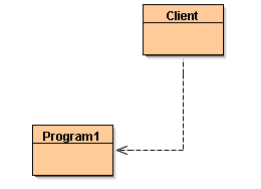
Dans ce projet, on trouve deux classes :

* une classe Program1 qui se contente, lorsqu’on appelle go, d’afficher un message (dans la réalité, un traitement particulier aurait lieu), et
* une classe Client qui appelle Program1.





le diagramme de classe ressemble à ceci



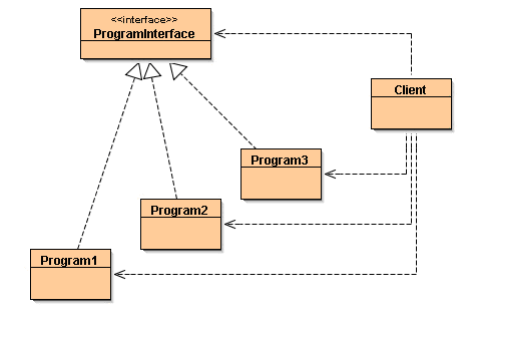
1. On souhaite rajouter deux classes Program2 et Program3 à notre projet.

Ces classes afficheront à l’écran le même message que Program1, excepté qu’elles y feront apparaître leur numéro de programme (2 ou 3).

Ajoutez ces classes (en dupliquant et modifiant le code de Program1).

Dans le client, on souhaite modifier le code des fonctions main pour que, selon le paramètre entier qui leur est passé en argument (1, 2, ou 3), ces dernières lancent le traitement du Program1, Program2 ou Program3.

***Pouvez-vous réaliser cette fonctionnalité ?***

en codant de la bonne manière, le diagramme de classe que vous devriez obtenir: ******

***Qu'est que vous remarquez?(quels sont les problèmes posés par une telle solution)***

1. ***Appliquer le pattern Factory***

Déléguer la création des objets de type Program à une classe dont le nom sera ProgramFactory

* Elaborer le diagramme de classes répondant à ce besoin
* Donner le code des différentes entités participantes à la réalisation du programme(classes, interfaces)
* Pouvez-vous ajouter un Program4 ? Cela a-t-il été compliqué à mettre en place ? Avez-vous dû modifier le code du Client ?

**Exercice n°2**

Le système de vente de véhicules gère des véhicules fonctionnant à l’essence et des véhicules fonctionnant à l’électricité.

Cette gestion est confiée à l’objet **Catalogue** qui crée de tels objets. Pour chaque produit, nous disposons d’une classe abstraite, d’une sous-classe concrète décrivant la version du produit fonctionnant à l’essence et d’une sous-classe décrivant la version du produit fonctionnant à l’électricité.

**Questions**

1) proposer une conception à ce problème en adoptant le pattern Factory

* donner le diagramme de classes tout en expliquent le rôle de chacune des composantes
* implémenter la solution

2) On envisage d'ajouter de nouvelles familles de produits en l'occurrence des Véhicules ainsi que des scooter Diesel

* Apporter les modifications nécessaires à la solution précédente afin de répondre à ce besoin. Que remarquez-vous?
* Proposer une solution en utilisant le pattern Abstract-Factory.