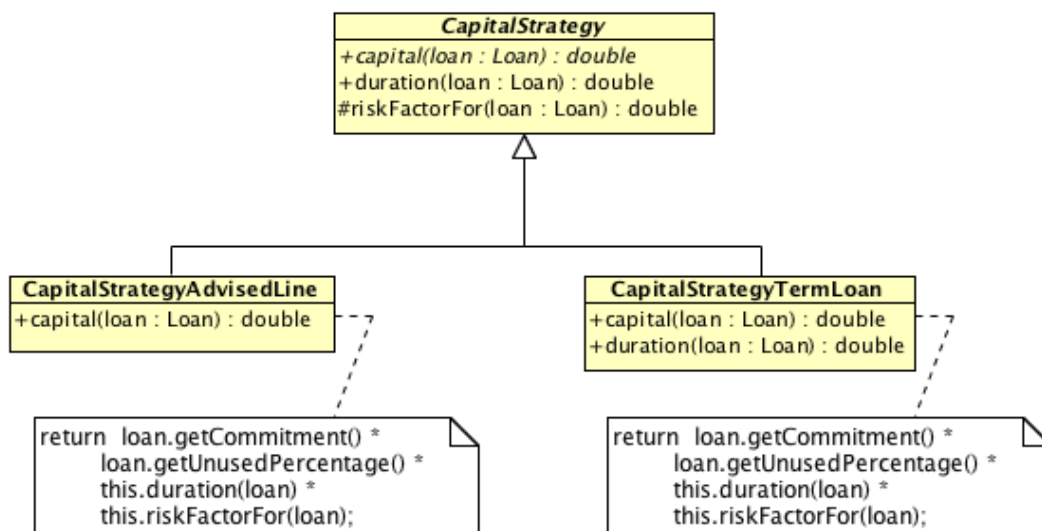


Exercice 1 : Template Method

Etudier le diagramme ci-dessous. On suppose qu'il existe une classe Loan qui représente un prêt.

Deux méthodes dans les sousclasses réalisent des étapes similaires dans le même ordre, cependant les étapes sont différentes. Nous allons appliquer le patron Template Method pour optimiser la situation.

Généraliser les méthodes en extrayant leurs étapes dans des méthodes avec des signatures identiques, puis remonter les méthodes généralisée pour former une Template Method.



Donner le diagramme de classes résultant.

Exercice 2 : Abstract Factory

Abstract Factory a pour objectif la création d'objets regroupés en familles sans devoir connaître les classes concrètes destinées à la création de ces objets.

L'exemple ici est un système de vente de véhicules qui 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.

Nous avons deux produits : les automobiles et les scooters.

Ces produits sont organisés en deux familles

Question 1 : Le code des classes « produit » est donné sur la zone moodle

Faire un « reverse java code » pour obtenir le diagramme de classes des produits.

Question 2 :

Maintenant il faut concevoir les classes Factory pour les 2 familles afin d'avoir, d'une part, la création des produits de la première famille (un véhicule à essence et d'un scooter à essence) et,

d'autre part, la création des produits de la deuxième famille (un véhicule électrique et un scooter électrique).

- Compléter le code de l'application avec l'implémentation des factory.
- Créer également la classe Catalogue qui contient la méthode « main » et qui peut créer et manipuler des véhicules sans devoir connaître les familles et les classes concrètes d'instanciation correspondantes.

Modifier le diagramme de classes en ajoutant les nouvelles classes et les liens (association, dépendance)

Exercice 3 : Command

Le pattern Command a pour objectif de transformer une requête en un objet, facilitant des opérations comme l'annulation, la mise en file des requêtes et leur suivi.

1. Lire le document « Pattern Command »
2. Récupérer le code de l'exemple sur moodle
3. Compléter la classe Utilisateur
4. Compiler et faire tourner votre programme
5. Dessiner le diagramme de séquence complet de l'extrait suivant de la méthode main de la classe Utilisateur

```
CommandeSolder commandeSolder = new CommandeSolder(10, 5, 0.1);
catalogue.lanceCommandeSolder(commandeSolder);

CommandeSolder commandeSolder2 = new CommandeSolder(10, 5, 0.5);
catalogue.lanceCommandeSolder(commandeSolder2);

catalogue.annuleCommandeSolder(1);

catalogue.retablitCommandeSolder(1);
```

A rendre :

- Exercice 1 : le diagramme de classes final
- Exercice 2 : le code complet de l'application et le diagramme de classes correspondant.
- Exercice 3 : Le code de la classe Utilisateur et le diagramme de séquence