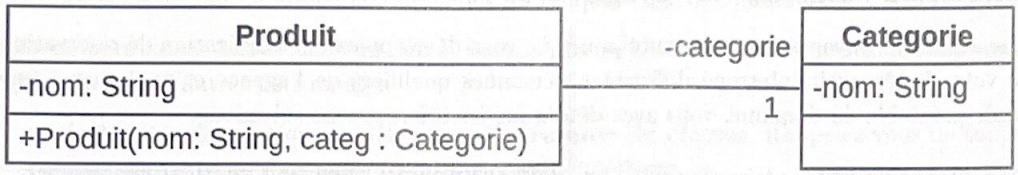
R3.03 - Analyse

Complément de TD N°3
Classes, Associations, Héritage, Polymorphisme

Exercice 1 : Analyse d'associations

Association simple

On considère le **diagramme de classes** suivant :



1 - Précisez le sens de la **navigabilité** / de la **dépendance** de cette **association**.

2 - Donnez le code **Java** correspondant aux squelettes des deux classes de ce diagramme.

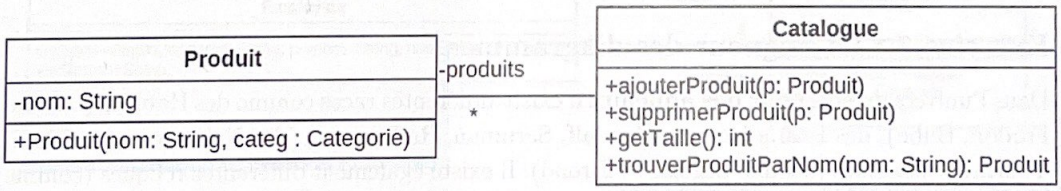
3 - Peut-on avoir un constructeur **sans aucun paramètre** dans **Produit** ? Pourquoi ?

4 - Si on manipule un objet de type **Produit**, peut-on **connaître** sa **catégorie** ? Si non, que faut-il modifier ou ajouter dans le diagramme ?

5 - Si on manipule un objet de type **Produit**, peut-on **modifier** sa **catégorie** ? Si non, que faut-il modifier ou ajouter dans le diagramme ?

Cardinalités multiples

On considère maintenant le **diagramme de classes** suivant :



6 - Précisez la **navigabilité** de l'association.

7 - Donnez les **attributs** (en **Java**) de chaque classe.

8 - Peut-on obtenir l'ensemble des produits d'un objet de type **Catalogue** ? Pourquoi ?

9 - Décrivez les **méthodes** de la classe **Catalogue** (ce qu'elles permettent de faire).

10 - D'après vous, combien de produits sont présents dans le catalogue à la création de celui-ci ?

11 - Est-ce que l'association pourrait être d'un autre type ? Expliquez pourquoi.

Association à double sens

Reprenons le diagramme final de la première partie de l'exercice (avec les classes **Produit** et **Categorie**). Nous souhaitons à présent pouvoir fournir, dans la classe **Categorie**, la **liste des produits de cette catégorie**.

12 - Modifiez le **diagramme** en conséquence.

13 - Associez à présent les 2 diagrammes pour avoir les **3 classes ensemble**. Peut-on demander au catalogue la liste des catégories de produits ? Si non, que faut-il rajouter ?

Exercice 2 : Agence de voyages

Une agence de voyages vous a recruté pour que vous développiez une application de réservation de vols. Après avoir interrogé différentes personnes qualifiées de l'agence et avoir mené une étude préalable du domaine, vous avez déterminé les informations suivantes :

- Des compagnies aériennes proposent différents vols.
- Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.
- Un client peut réserver un ou plusieurs vols.
- Une réservation concerne un vol et un client.
- Une réservation peut être confirmée ou annulée.
- Un vol a un aéroport de départ et un aéroport d'arrivée.
- Un vol a un jour et une heure de départ et un jour et une heure d'arrivée.
- Un vol peut comporter des escales dans des aéroports.
- Une escale a une heure d'arrivée et une heure de départ.
- Chaque aéroport dessert une ou plusieurs villes.

1 - A partir de ces informations, concevez un **diagramme de classes** permettant de modéliser l'application de réservation de vols.

Exercice 3 : Le seigneur des diagrammes

Dans l'univers du **seigneur des anneaux** il existe différentes races comme des Hobbits (comme Frodon, Bilbo), des Istaris (comme Gandalf, Saruman, Radagast...), des Nains (comme Gimli, Thorin) et des Elfes (comme Légolas et Elrond). Il existe également différentes reliques (comme l'Arkenstone, l'anneau unique, les silmarils...). Enfin, il y a aussi différents lieux (comme par exemple la Comté, le Gondor, ou bien les montagnes d'Erebor...).

Il est possible de connaître le nom et l'âge d'un Hobbit ainsi que le nombre de petits-déjeuners requis pour le satisfaire.

Pour les Istaris, on peut aussi connaître leur nom et leur âge ainsi que leur nvieau de magie (exprimé entre 1 et 100).

Pour les Nains, on doit pouvoir connaître leur nom , leur âge et le nombre de pièces d'or qu'ils possèdent.

Enfin, pour les elfes, il est nécessaire de connaître le nom, leur âge et leur acuité visuelle. (entre 0 et 1000).

Chaque personnage possède aussi un genre (homme, femme ou neutre)

Pour les différentes reliques, il doit être possible de connaître leur nom et le niveau de rareté (banal, commun, rare, très rare).

Pour les lieux, on doit pouvoir connaître le nom et la localisation exprimée par la longitude et la latitude du lieu.

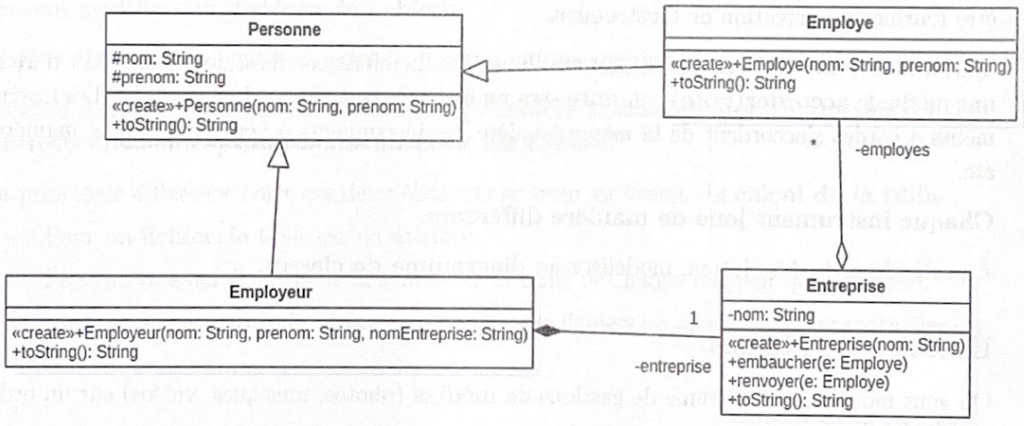
Il existe des lieux spéciaux appelés caches. Ces caches renferment plusieurs reliques. Aussi, une cache aura un niveau de rareté, comme ,pour les reliques (banal, commun, rare, très rare).

Chaque personnage habite dans un lieu et possède des reliques. Les personnages peuvent aussi être amis avec d'autres personnages.

A partir de cette description, modélisez un **diagramme de classes**. Rappelez-vous de toutes les notions qui peuvent vous aider (**énumérations, héritage...**)

Exercice 4 : Héritage, constructeurs, redéfinition

On considère le diagramme de classes suivant :



1 - Quels sont les attributs de la classe **Employeur** ?

2 - A votre avis, pourquoi le constructeur de la classe **Employeur** possède trois paramètres ? Qu'est ce que dois faire le constructeur ?

3 - Donnez le code java de chaque classe en précisant les attributs et le code du constructeur (on met de côté la fonction **toString**, pour le moment).

Les méthodes **toString** suivantes ont été redéfinies :

- Celle de **Personne** qui donne le nom et le prénom de la personne. Par exemple : "John Smith"
- Celle de **Employeur** qui fait de même
- Celle de **Entreprise** qui donne son nom puis le nom de chaque employé. Par exemple : "Mon entreprise - [John Smith, Jean Bon, Guy Tarenbois]"
- Celle de **Employeur** qui donne son nom puis l'affichage de l'entreprise. Par exemple : "Patrick TheBoss - Mon entreprise - [John Smith, Jean Bon, Guy Tarenbois]"

4 - Toutes ces redéfinitions sont-elles utiles ?

5 - Que peut-on dire de spécial sur la redéfinition du **toString** d'**Employeur** ? Donnez le code java de cette méthode.

Exercice 5 : Orchestre

Un **orchestre** contient des **musiciens**. Chaque **musicien** joue d'un **instrument** qui possède une méthode **jouer**, de même que le **musicien**. L'**orchestre** possède également une méthode **jouer** dont le rôle est de faire jouer tous les **musiciens**.

Il est possible d'ajouter ou de retirer des **musiciens** à l'**orchestre**.

Il existe des **instruments à vent** (trompette, cor, trombone, flûte) des **instruments à cordes** (violon, violoncelle, contrebasse), des **percussions** (timbales, cymbales, tambour). Un **instrument** possède une certaine valeur (un prix, en euros) et une date d'achat. Ces valeurs doivent être fournies à la création de l'instrument.

Tous les **instruments** s'accordent autour d'une note de référence : il est donc nécessaire d'avoir une méthode **accorder(note)** (où **note** sera un **entier**) dans chaque **instrument**. Les instruments à cordes s'accordent de la même manière, les instruments à vent de la même manière, etc.

Chaque instrument joue de manière différente.

A partir de cette description, modélisez un **diagramme de classes**.

Exercice 6 : Médias

On veut modéliser un système de gestions de **médias** (photos, musiques, vidéos) sur un ordinateur.

Un **média** est défini par :

- Un **nom** (par exemple "Ma super musique")
- Un **chemin système** (par exemple "C :/Users/humble/Musique/mamusique.mp3")

Il doit être possible d'**ouvrir** un média.

Il existe différents types de **médias** :

- Des **photos** qui possèdent une **hauteur** et une **largeur**. L'ouverture d'une photo l'affiche sur l'écran
- Des **médias "jouables"** qui possèdent une **durée**. L'ouverture d'un tel média ouvre une fenêtre chargeant le média puis le joue. Il existe deux types de médias jouables :
  - Des **musiques** qui possèdent un **format audio** (parmi **mp3, wav, ogg**)
  - Des **vidéos** qui possèdent un **format vidéo** (parmi **mp4, wmv, avi**)

L'opération qui vise à **jouer** un **média jouable** dépend du **type de média cible** (pas le même comportement si c'est une musique ou une vidéo)

1 - Donnez le **diagramme de classes** du système en vous basant sur cette description.

On souhaite maintenant gérer des **playlists**. Une **playlist** possède un nom et permet de jouer des vidéos et des musiques à la suite. Elle gère donc un ensemble de médias particuliers.

2 - Étendez votre **diagramme de classes** pour permettre de gérer des **playlists**.

On veut maintenant que nos **playlists** puissent elles-mêmes jouer **d'autres playlists** et soient représentées par un fichier dans l'ordinateur (il suffirait alors de cliquer dessus pour lancer la playlist).

3 - En modifiant **assez simplement** votre **diagramme de classes**, rendez possible cette évolution.

Exercice 7 : Système de fichiers

On veut modéliser un **système de fichiers**.

Chaque **élément du système** possède un nom, un propriétaire.

Il existe deux types d'éléments systèmes : Des **fichiers** et des **dossiers**. Un **dossier** contient différents **éléments systèmes** (des fichiers et des dossiers)

La principale différence entre ces deux éléments se situe au niveau du **calcul de la taille** :

- Pour un **fichier**, la taille est un attribut.
- Pour un **dossier** il s'agit de la somme de la taille de chaque élément qu'il contient.

1 - Donnez le **diagramme de classes** du système de fichiers en vous basant sur cette description.