# Exercice 1

Voir sur le diagramme tp01\_ex01.jpg

# Exercice 2

## a) Corrigez le code Java fournit en annexe selon les notions vues en cours. Vous avez le

## droit de créer/modifier des classes.

## b) Justifiez chacune de vos corrections

**Généralisatioon des classes Assistants & Professeur en une Classe Employee**

=> seule valeur qui diffère nom de l'emploi

=> ajout de la valeur Position dans le constructeur pour la définir à l'instantiation

**Généralisation des classes EtudiantTempsPlein et EtudiantTempsPartiel en une Classe Student**

=> seul attribut qui diffère le jours de travail

=> création de deux constructeur 1 avec workDay pour les temps partiel, l'autre sans pour les plein temps

=> ajout méthode isPartTime qui contrôle s'il y a un workday set afin de savoir si la personne est en temps partiel

- generalisation des classe Employee et Student en une classe Person

=> nom et prenom commun à tous

- Déplacer tous les codes d'affichage d'information dans les toString des classes respectives.

=> méthode héritée de la classe object et donc commune à toutes les classe qui pourra être appelé pour toutes de la même manière

## c) Réalisez le diagramme de classes correspondant à la correction que vous avez

## effectuez.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquementre

## d) Expliquez la différence de fonctionnement entre votre version du programme « Main » et celle qui vous a été fournie.

=> Instanciation de 2 types d'objet isof 4

=> Avec la surcharge de la classe toString qui est présente dans toutes les classes, je suis capable de loopé dans toutes les classes et d'appeler la même méthodes sans avoir à tester le type d'objet

## e) Décrivez, avec vos propres mots, la différence conceptuelle qu’il y a entre une généralisation et une spécialisation.

=> Lorsque on identifie que des objets ont des attributs commun, des méthodes communes et autres comportements alors ils probable que nous pouvons les généralisé afin de rentre le code plus générique

=> Au contraire, lorsque certaine spécificité d'un objet ne s'applique qu'à un certain type, il est probable qu'une spécialisation est nécessire

## f) Décrivez ce qu’est le polymorphisme et trouvez-en en application dans votre code.

Une méthode implémenter avec des signatures différentes pour répondre à différents uses cases et exécuter des algos différents