

TP7– R2.01

Semaine 12

Objectifs

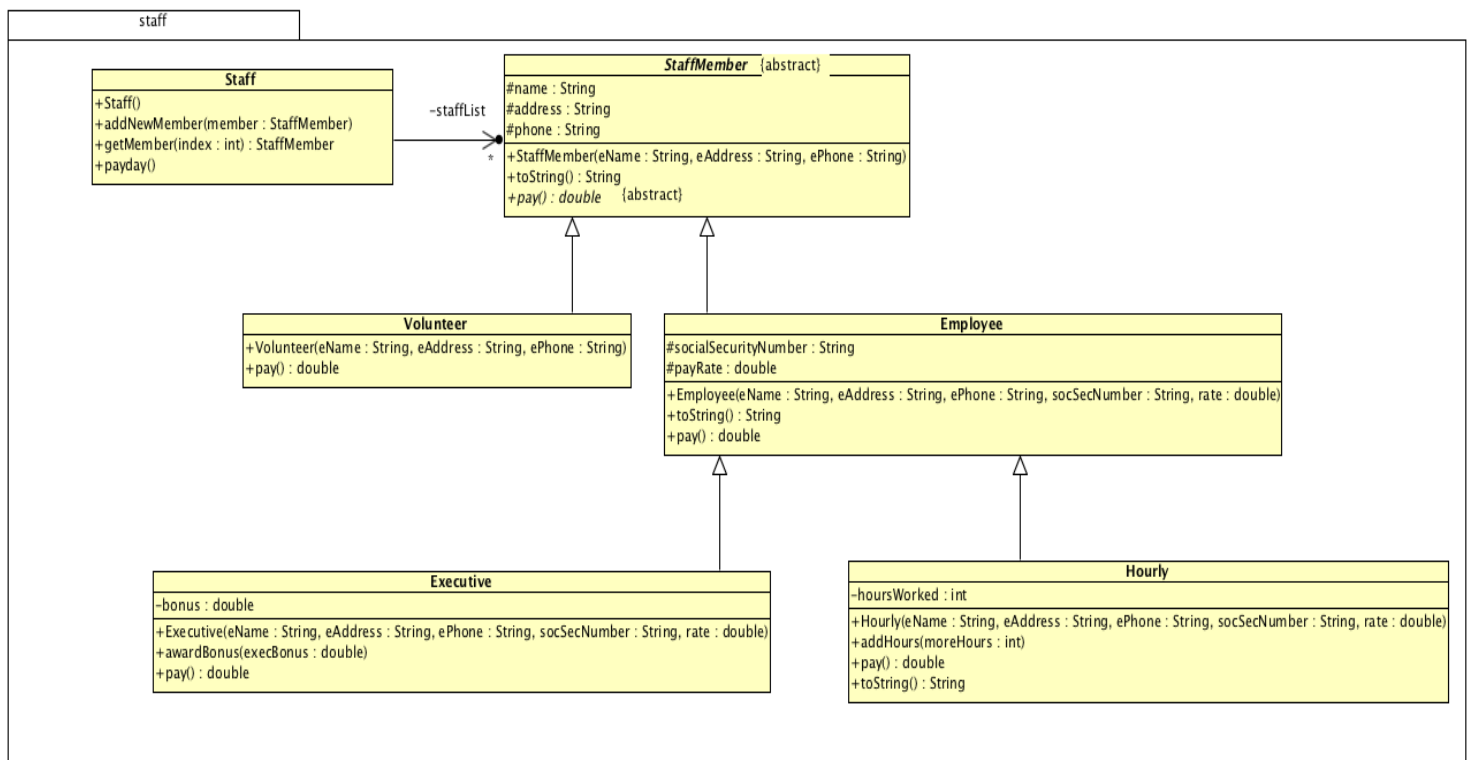
Continuer à étudier l'héritage
 Utiliser des classes et des méthodes abstraites
 Comprendre l'utilisation des visibilité protégées avec l'héritage
 Utiliser le polymorphisme (méthodes)

Rappels de cours : reprendre les photocopiés des cours 5 et 6**L'application Staff**

L'application **Staff** (Personnel) permet de modéliser les différentes catégories de personnel dans une entreprise.

Un héritage est utilisé pour classer ces différentes catégories de personnels.

Le diagramme de classes ci-dessous montre l'organisation des différentes classes.

**Vocabulaire**

Staff : Personnel	Hourly (employee) : salarié horaire
Staff Member : membre du personnel	Executive : cadre
Employee : salarié	Volunteer : bénévole

La classe racine est **StaffMember** (membre du personnel). Elle est abstraite et définit les coordonnées d'un personnel : nom, adresse, téléphone.

Il y a deux catégories de personnels : les bénévoles (Volunteer) et les employés salariés (Employee).

La méthode **double pay()** permet de calculer le salaire de chaque type de personnel :

pour **StaffMember** : la méthode est abstraite ;

pour **Volunteer** : son salaire est égal à 0 ;

pour **Employee** : définit un taux de base pour le salaire (**payRate**) qui est retourné par la méthode **pay()** ;

pour **Hourly** : définit le nombre d'heures travaillées (**hoursWorked**), son salaire est le taux de base (qui doit être le taux horaire) multiplié par le nombre d'heures travaillées ;

pour **Executive** : le bonus est le surplus auquel il a droit. Son salaire est le taux de base augmenté du bonus.

Une fois payé, le nombre d'heures travaillées du salarié horaire est remis à 0, de même pour le bonus du cadre.

La classe **Staff** (Personnel) représente le service du personnel de l'entreprise et utilise un **ArrayList** pour implémenter la liste du personnel.

- Elle peut ajouter un nouveau personnel avec : **addNewMember(StaffMember member)**

- Elle propose d'obtenir la référence d'un personnel si on connaît son index dans la liste

StaffMember getMember(int index)

– Elle réalise le paiement de tout le personnel avec la méthode **payDay()**

Travail à réaliser :

- Il s'agit d'implémenter toutes les classes du package **staff** ainsi que la classe **Staff** (voir explications ci-dessous) sans oublier la javadoc et sa génération. On vous propose de suivre les étapes ci-dessous.
- Il faudra également écrire une classe de scénario à l'extérieur du package qui créera des instances de toutes les sous-classes de **StaffMember**, une instance de **Staff** et qui vérifiera le bon fonctionnement des classes et déclenchera la paye avec la méthode **payday()**.

Etape 1- Héritage des attributs et des méthodes

1. Remarquer sur le diagramme de classes les propriétés {abstract}
2. Revoir les visibilités (voir cours 6)
3. Ecrire le code de la classe **StaffMember**.

Etape 2- Chaînage des constructeurs (cours 6)

1. Ecrire le code des classes **Employee** et **Hourly**, **Executive** et **Volunteer**.
2. Ecrire une classe **StaffScenario** (en dehors du package staff) et vérifier les fonctionnements de vos classes.

Remarque :

La méthode **void awardBonus(double execBonus)** de la classe **Executive** permet de donner une valeur pour le bonus.

Etape 3- Finaliser l'application

1. Ecrire le code de la classe **Staff** (voir explications ci-dessous)
2. Compléter le code de **StaffScenario**

La classe Staff

La classe **Staff** (Personnel) représente le service de personnel de l'entreprise :

- Elle utilise un `ArrayList` pour implémenter la liste du personnel.
- Notez bien que le super type (`StaffMember`) est utilisé pour typer tous les objets qui seront insérés dans l'`ArrayList`. Cela signifie par exemple que le type de retour de la méthode **`getMember()`** est `StaffMember`.
- Elle peut ajouter un nouveau personnel avec la méthode
`addNewMember(StaffMember member)`
- Elle propose d'obtenir la référence d'un personnel si on connaît son index dans la liste
`StaffMember getMember(int index)`
- Elle réalise le paiement de tout le personnel avec la méthode **`payDay()`** qui fonctionne de la manière suivante :

Pour chaque membre du personnel contenu dans l'`ArrayList` :

1) la méthode commence par imprimer ce membre (utilise le `toString()` de la classe) ;

2) puis elle récupère le montant de la paye en envoyant le message `pay()` au membre

- si ce montant est égal à 0 elle imprimera « Thanks ! », car il s'agit d'un bénévole

- sinon elle imprimera « Paid : » suivi du montant du salaire ;

3) enfin elle imprimera une ligne de séparation "-----" entre chaque personne.