

Travail 1: Système de gestion des employés d'une entreprise

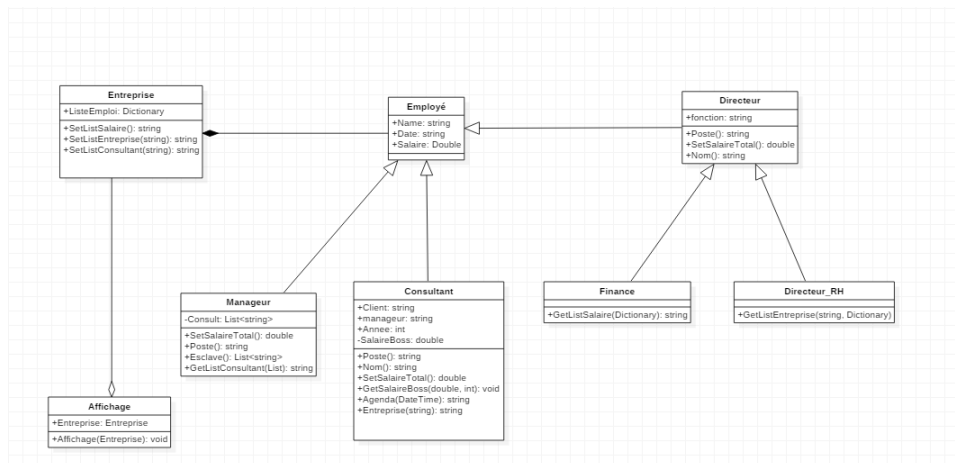
Julien Stilmant
14054

Thierry Frycia
14212

27 octobre 2017

1 Structure

1.1 Diagramme



1.2 Fonctionnement du code

1.2.1 Structure générale

La fonction `Main()` qui se trouve dans la classe `MainClass` dans le fichier `Execution.cs` appelle la fonction `Emploi()` qui est définie juste en dessous. Cette fonction va lire le contenu des fichiers json, l'interprète et mémorise les données dans les objets du type approprié.

Après ça, la fonction `Affichage()` est appelée avec en paramètre le dictionnaire qui contient tous les objets désérialisés auparavant. C'est cette fonction qui fait l'interface avec l'utilisateur et qui va par après appeler les méthodes appropriés sur les objets pour obtenir leurs attributs.

1.2.2 Classes principales

Consultant La classe **Consultant** est une sous-classe de **Employé**. Elle hérite donc simplement des attributs **Name**, **Date** et **Salaire**. En plus de ça, elle génère des attributs personnels **Client** et **manager**.

Le calcul du salaire se fait dans la méthode **SetSalaireTotal**, qui vérifie si les missions ont été faites dans une boîte externe ou non et prends en compte le salaire du manager.

Directeurs **Directeur_RH** et **Finance** héritent tous les deux de la classe **Directeur** qui définit les attributs pour la fonction et le salaire de la personne.

La classe **Directeur_RH** contient une méthode **GetListEntreprise()** qui renvoie les noms et les dates des consultants qui ont travaillé dans une certaine entreprise. Cette méthode sera appelé par la méthode **ListConsultants()** dans la classe **Entreprise**.

Le même principe est utilisé pour la méthode **GetListSalaire()** du directeur des finances.

Manager Les deux méthodes principales de cette classe sont **SetSalaireTotal()** et **GetListConsultant()**.

La première calcule le salaire en fonction du nombre de consultants sous sa responsabilité.

La deuxième est utilisée par la méthode **SetListConsultants()** dans la classe **Entreprise** pour générer le rapport du manager de la même façon que pour les directeurs.

Entreprise **SetSalaire()** envoie le salaire des managers à leurs consultants pour que ceux-ci puissent calculer leur salaire.

DirecteurFinance(), **DirecteurRH()** et **NomManager()** renvoient le nom de la personne pour pouvoir l'afficher au final.

SellistSalaire() **SellistSalaire** envoie à la classe **Finance** un dictionnaire des noms et des salaires pour que l'instance de cette classe puisse renvoyer la liste des salaires.

SellistEntreprise envoie à la classe **Directeur_RH** un dictionnaire des noms et des dates pour que l'instance de cette classe puisse renvoyer la liste des consultants travaillant dans une entreprise donnée.

SellistConsultant envoie à la classe **Manager** une liste des noms des consultants et de l'entreprise où ils travaillent pour que l'instance de cette classe puisse renvoyer cette liste.

2 Fichiers

Toutes les données sont stockées dans des fichiers texte au format `json` séparés selon le poste ou la fonction.

Dataadress.json Contient la liste des emplacements des fichiers `json` correspondant aux employés.

Dataemployé.json Contient une liste de dictionnaires chacun reprenant le nom, le poste, la date de naissance, le salaire et une liste des personnes associés.

Data(nom).json Chaque fichier représente un consultant et contient la liste de ses missions avec leur date de début et de fin.

3 Interface

Pour lancer l'application, veuillez executer le fichier "Lancer l'Application". Lorsque vous exécutez l'application, vous arrivez directement sur l'interface. vous devez alors vous identifier par votre Prenom et Nom en commençant par une Majuscule pour chacun.(Exemple : Julien Stilmant) Ensuite une session propre à votre nom et votre poste dans l'entreprise s'ouvrira. Vous ne pourrez alors n'effectuer que des actions propres à votre poste dans l'entreprise. Seul le directeur principal a le droit d'accéder à toutes les fonctions.