Rapport Programmation Orientée Objet

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

Mode d'emploi de notre programme

a) Les Entrées

Pour entrer les données nous avons décidé de séparer tous les fichiers textes d'entrée pour plus de clarté lors du remplissage des données et une possibilité de restreindre des accès à certains fichier textes pour certains employés de l'entreprise.

Quelques notes avant de vous expliquer le format des fichier textes :

- Les matricules choisis sont « arbitraires » mais par logique nous avons choisi 2 lettres qui indique le rôle puis des chiffres ordonnés (les chiffres sont notés par des « xx » ici mais sont bien 2 chiffres dans le fichier texte).
- Les missions doivent obligatoirement être écrite en ordre chronologique de date de début.
- Un matricule client est déjà utiliser sans être à rentrer dans la base de données car c'est celui de l'entreprise elle-même (CL00).

Voici le format de chaque fichier texte, entre les guillemets se trouve ce qui doit figurer à chaque ligne du fichier texte :

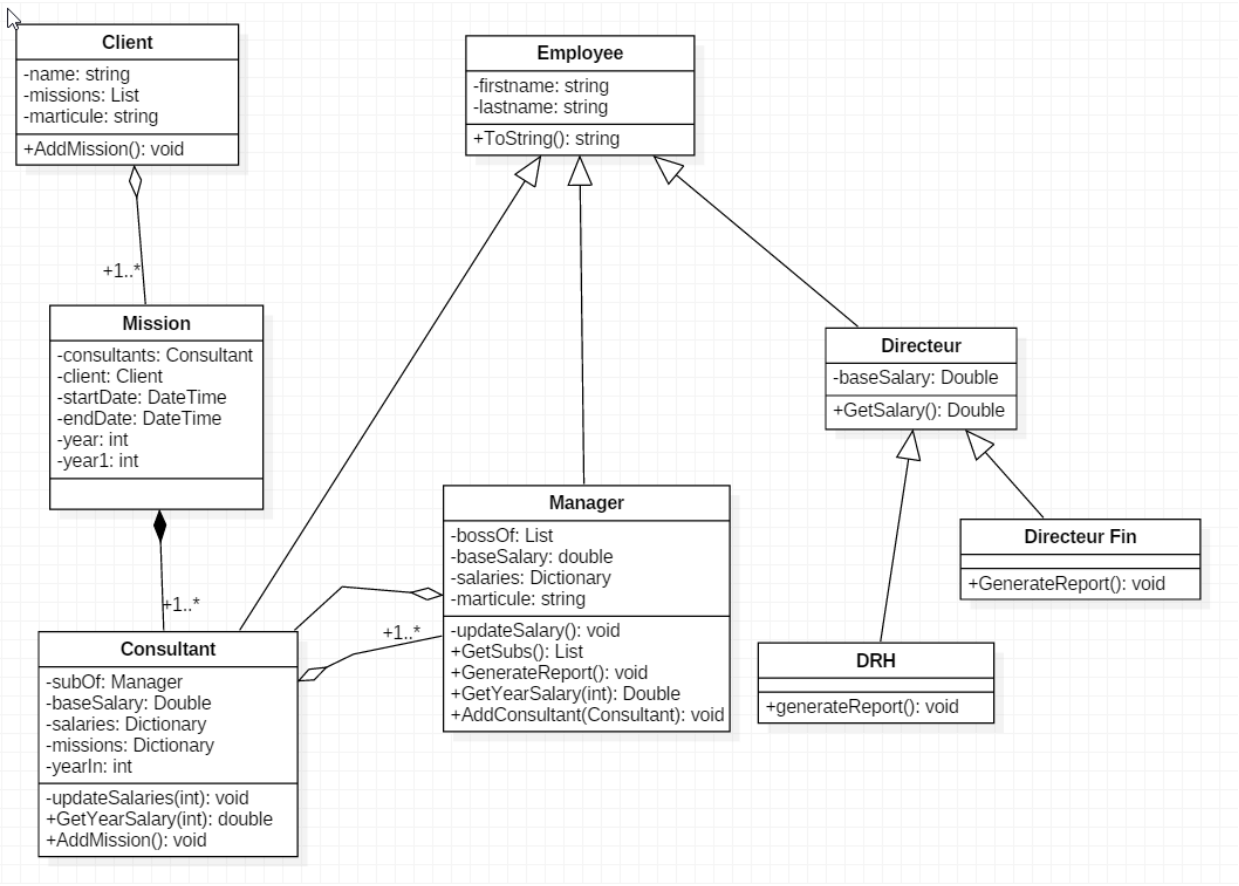
- Clients : « Nom CLxx » (CLxx est bien le matricule)
- Consultant : « Prénom Nom MAxx COaaxx » (Maxx est le matricule du manager du consultant et le « aa » est l'année d'arrivée dans l'entreprise du consultant)
- Directeurs : « CodeDirecteur Prénom Nom » (les codes directeurs sont DI simple directeur, DF directeur financier et DH directeur des ressources humaines)
- Managers : « Prénom Nom MAxx »
- Missions : « CLxx COaaxx Date Date » (les deux dates sont les dates de début et de fin de mission dans l'ordre écrite avec des tirets de type *dd-mm-yyyy*)

Si nécessaire les fichiers d'entré son déjà remplis d'exemples pour toute hésitation quant aux entrées.

b) Les Sorties

Nos sorties sont très simples, par l'interface console vous sélectionner le rapport désiré, une fois le rapport généré il se trouve dans le répertoire du programme sous format TXT prêt à être lus.

L'UML et structure de code



Le code est structuré en Class. Tout d'abord nous avons une class « Employee » qui sera la super class de toutes les class qui sont des employés de l'entreprise à gérer. Les class Directeur, Manager et Consultant héritent toute les trois d'Employee et apporte à chaque type d'employé de l'entreprise des fonction particulières à leur poste. Il y a deux types de directeurs spécifiques. Le DRH et le Directeur financier héritent donc de la super class directeur et son toute les deux uniquement créent pour générer les rapports. Manager et Consultant sont deux class agrégée entre elles car chaque consultant a un manager et chaque manager à un ou plusieurs Consultants. Pour le bien des entrées et de la clarté des rapports nous avons créé une class Mission. La class consultant est composée de missions, car à tout moment dans la vie du consultant dans l'entreprise il est en mission. La mission doit aussi être liée à un client, ce client peut être l'entreprise elle-même. Client et mission sont donc agrégés car un client à une ou plusieurs missions.

Une fois toutes ces classes crées et liée ensemble nous avons la class program, cette class est à exécuter pour prendre les entrées et générer les rapports. Elle va d'abord chargée tous les fichier texte d'entrée et crée toutes les variables correspondant à ce qui est mis dans les inputs. Il va ensuite y avoir une interface console avec laquelle il faut interagir pour générer un rapport. Le rapport est alors écrit et posté dans le répertoire du programme dans un format .txt.