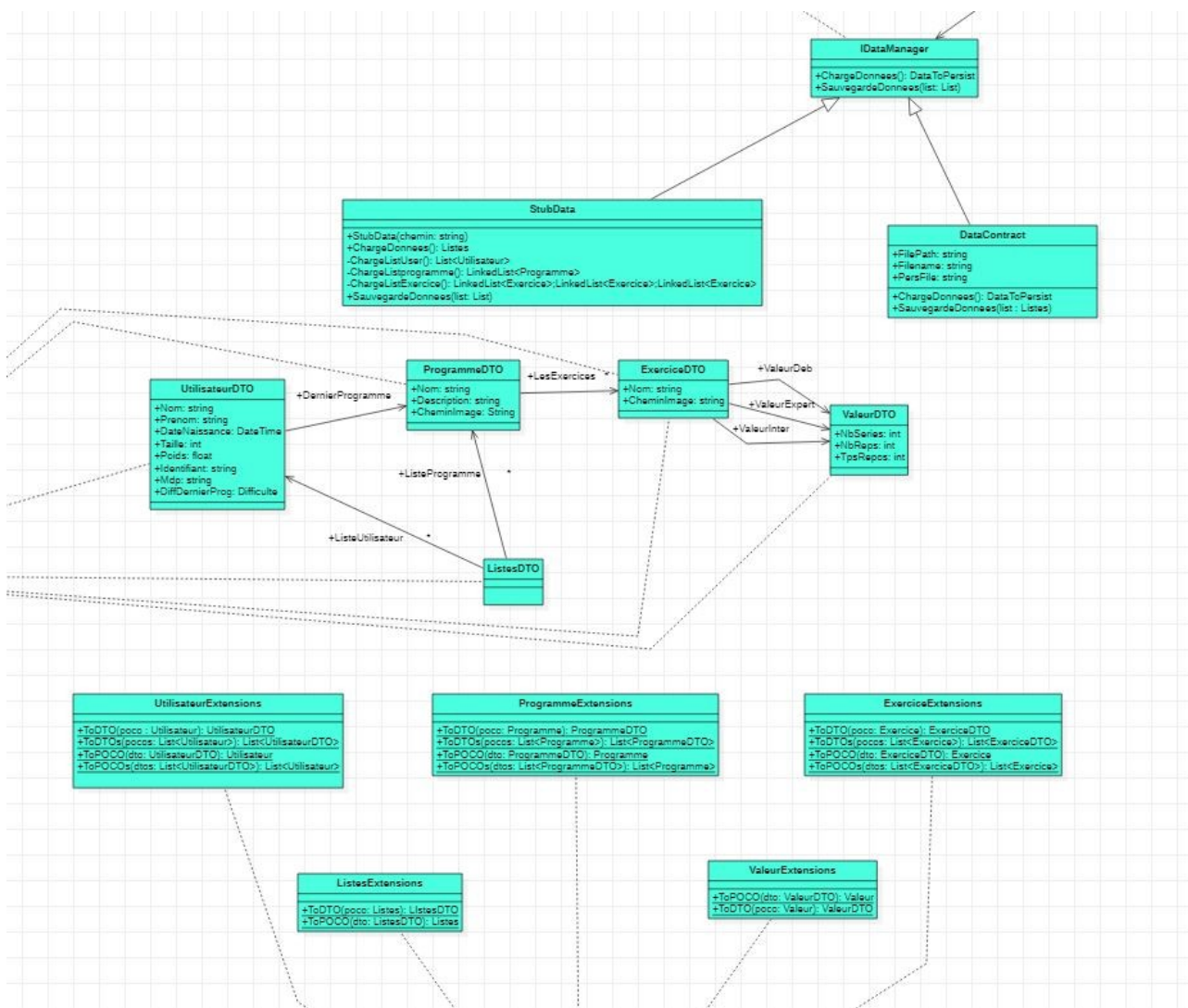


DOCUMENTATION PARTIE PROJET TUTEURÉ

◆ Partie persistance du diagramme de classe



◆ Description de la persistance

Le projet PersistenceData contient nos données virtuelles dans notre Stub, ainsi que notre persistance dans un fichier XML. Cette hiérarchie démarre avec une interface « IDataManager » que nous avons implémenté à notre stub ainsi qu'à notre DataContract, classe qui effectue la persistance. Nous avons décidé d'avoir une persistance qui n'impacte pas l'écriture de nos différentes classes dites hermétique, on ne retrouve aucune trace de l'utilisation des DataContract, DataMember dans nos classes dites POCO (Application). On utilise des objets DTO (Data transfert Object) pour persister les données. A chaque sérialisation les objets de types POCO seront transformés en DTO à l'aide des méthodes ToDtos et ToDTO et inversement lors de la désérialisation les méthodes ToPOCOs et ToPOCO présentent dans des classes d'extensions pour charger les données.

La persistance a une dépendance vers Application car il instancie dans la classe Listes des collections d'utilisateurs et de programmes essentielles lors du chargement des données à partir du fichier XML ; et dans le Stub.

◆ Description des ajouts personnels

Enfin, nous pensons que nos ajouts personnels premièrement sont la possibilité de s'inscrire et donc de se connecter à l'application ce qui donne des droits supplémentaires à l'administrateur qui se serait connecté tels que l'ajout de programme parmi-ceux déjà présents, leur suppression, l'ajout d'exercices ou leur suppression.

Mais principalement, notre grand ajout personnel est le fait de pouvoir Commencer un programme en cliquant sur un bouton qui ouvrira une nouvelle fenêtre (ExerciceWindow) dans laquelle seront affichés un par un les Exercice du Programme à effectuer. Cette fonctionnalité (Lancer un programme) rend l'application vraiment plus intéressante qu'un simple master-detail.

ExerciceWindow
-List Listes
-exenum IEnumerable<Exercice>
-nextExButton_Click()