

Notice explicative du code développé

DE OLIVEIRA MORENO NEVES, José Afonso

Cette petite notice sert à expliquer le code développé pour la phase individuelle du projet et à montrer les diagrammes UML obtenus. Dans le paquetage «executables», j'ai aussi créé 5 classes exécutables simples pour illustrer le comportement des différents éléments du projet. Dans la fin du document, je présente les diagrammes UML du projet. Le projet a beaucoup de classes donc je les ai divisé en diagrammes d'accord avec son paquetage.

Profiles

Tout d'abord, j'ai choisi de mettre en œuvre les consommateurs et les producteurs comme les mêmes objets car, du moins à ce stade, ils ne sont caractérisés que par leurs profils de puissance qui sont équivalents. Ensuite, pour pouvoir regrouper des profils implementés par différentes classes, j'ai créé une interface que j'ai appelé «Profile». Si, à l'avenir, des fonctionnalités sont ajoutées au simulateur qui nécessitent la distinction entre producteurs et consommateurs, il faudra modifier cette construction. Cependant, l'interface que j'ai créée peut être utilisée comme un attribut des nouveaux producteurs ou consommateurs.

Pour préciser les profils au long de l'année, j'ai utilisé 3 mécanismes. Tout d'abord, j'ai créé la classe «YearVariation» qui permet de définir un jour de début et un jour de fin d'opération. Deuxièmement, j'ai créé la classe «WeekVariation» qui permet de définir les jours de la semaine où le profil fonctionnera. Troisièmement, j'ai créé l'interface «Parameter» qui permet de créer des paramètres pour chaque profil dont la valeur varie tout au long de l'année. Pour l'instant, j'ai créé deux types de paramètres, constants et sinusoïdaux. Le diagramme UML que représente les paramètres est présenté dans la Figure 1.

Pour les variations dans le même jour, j'ai considéré deux types de profils : les profils qui durent toute la journée et les profils qui durent seulement une partie de la journée et qui peuvent se répéter. Les premiers ont une implémentation simple et je présente un exemple dans la classe «DayConstantProfile». Les seconds peuvent en fait représenter les premiers si la durée appropriée est définie et pour eux, je présente trois types différents, constants, linéaires et quadratiques. Pour représenter le fait qu'ils commencent à un certain instant, ont une certaine durée et peuvent être périodiques, j'ai créé la classe «Square» et les ai fait hériter de la classe abstraite «SquaredProfile» qui a un attribut «Square» et les attributs des différentes variations annuelles.

D'autre part, j'ai également défini une classe qui implémente un groupe de profils pour pouvoir représenter différents appareils ou profils composés de différents profils que j'ai nommé «ProfilesGroup». Les classes que j'ai mentionné sont représentés dans la Figure 2.

Outils de visualisation des résultats

Jusqu'à présent, j'ai créé deux types d'outils de visualisation de données, une classe pour écrire les données dans un fichier .csv appelée «DataToCSV» et une classe pour afficher des graphiques de puissance, d'énergie et de puissance perdue appelée «DataToPlots». J'ai essayé de rendre simple la création de graphiques de nouvelles quantités en écrivant cette classe. J'ai également défini une interface appelée «DataOutputs» pour les outils de visualisation afin qu'il soit simple d'en ajouter d'autres. Le diagramme correspondant aux outils de visualisation est présenté dans la Figure 3.

Simulateur

J'ai également créé un simulateur que j'ai appelé «City», qui reçoit des «ProfileGroups» de producteurs et de consommateurs et des «DataOutputs» et simule la production et la consommation. Il envoie également les résultats à travers la classe «Résultats», présentée dans la Figure 4, que j'ai définie pour les outils de visualisation, crée un dossier et un fichier dédié à la simulation et applique les outils aux résultats. La classe «City» est représenté dans le diagramme de la Figure 5.

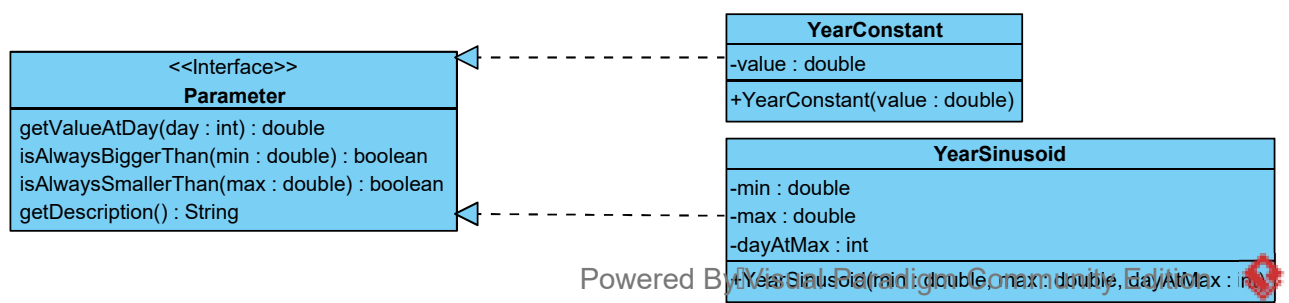


FIGURE 1 – Diagramme UML associé au paquetage «parameters».



Page 3 de 6

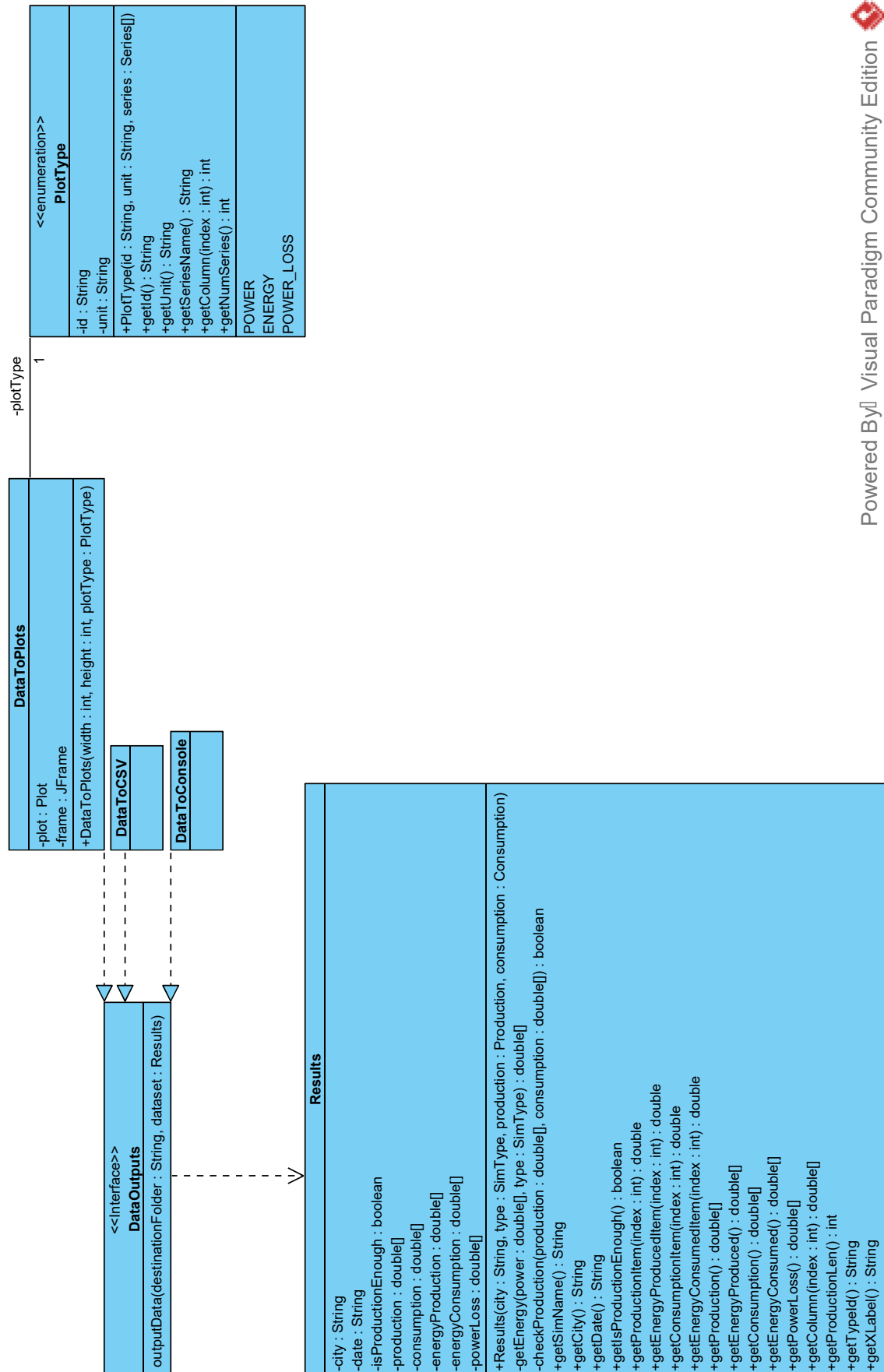


FIGURE 3 – Diagramme UML associé au paquetage «dataOutputs».

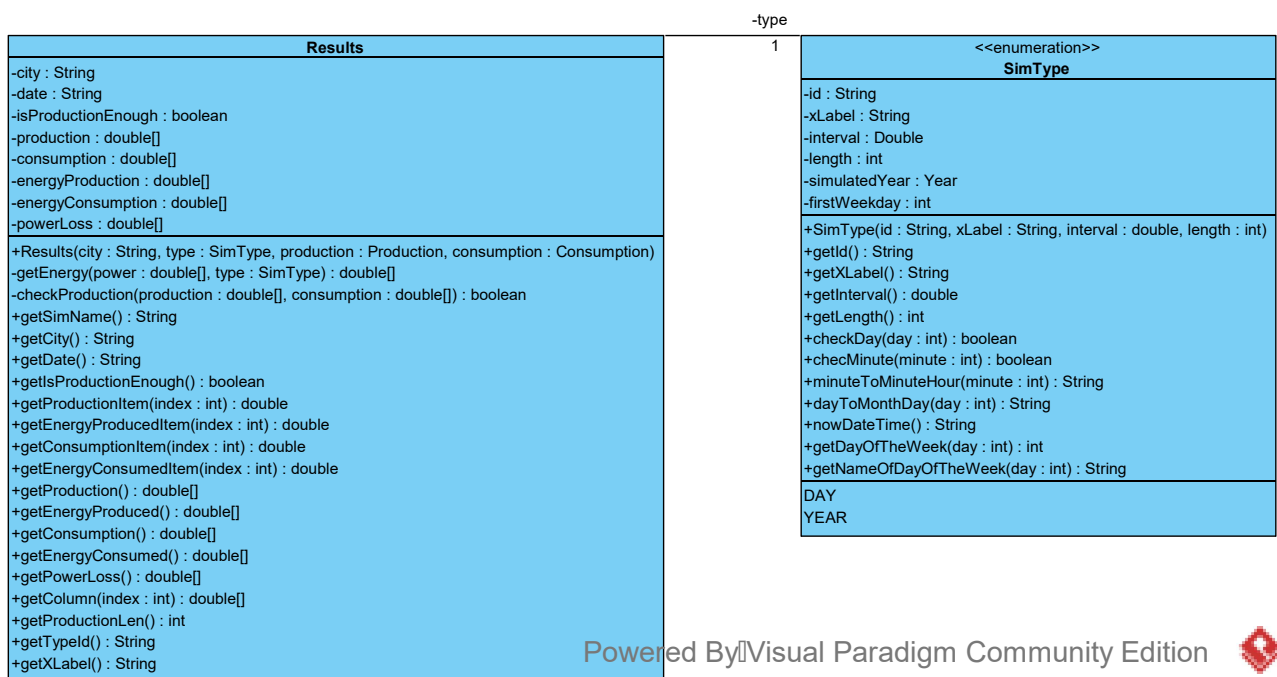


FIGURE 4 – Diagramme UML au paquetage «results».

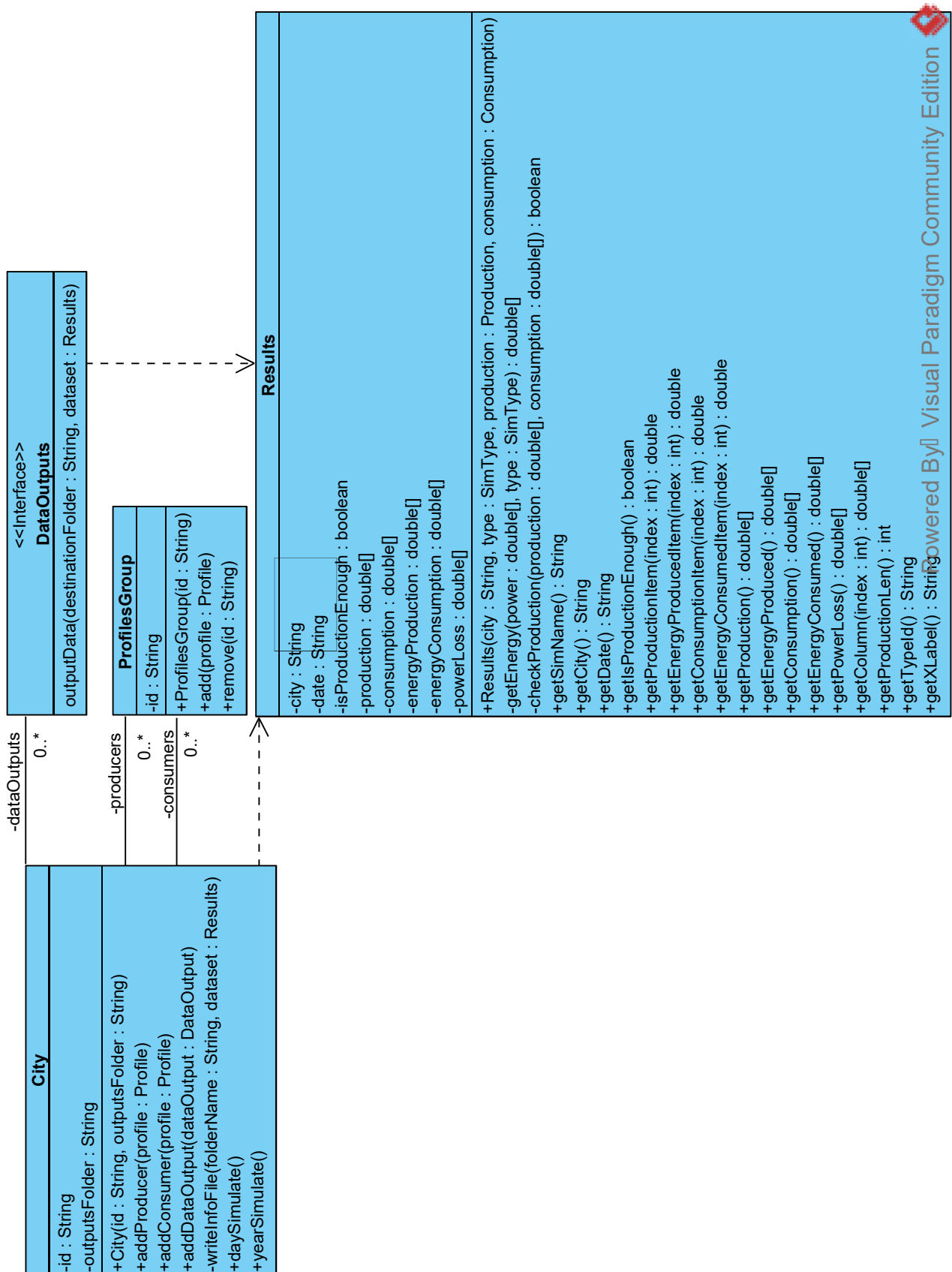


FIGURE 5 – Diagramme UML associé à la classe «City».