La table pouvait être interprété soit comme pleins d’individus avec différentes caractéristiques, soit comme pleins de variables avec plusieurs observations. C’est cette deuxième possibilité que nous avons choisie car elle rend les opérations sur les colonnes plus simples au détriment des opérations sur les lignes. Nous avons donc fait le choix de représenter les données sous forme de dictionnaire car ce type de stockage permet de gérer plus facilement les variables.

Lorsque l’utilisateur souhaite modifier le jeu de donnée, il transmet au programme la liste des modifications qu’il souhaite réaliser sur quelles variables et quelles données. L’ensemble de ces modifications élémentaires sont traités une à une grâce aux classes dérivant de la classe Transformation.

Le programme permet d’ajouter des données à notre table. Si l’on veut ajouter des variables, on peut soit importer des données grâce à la méthode import contenu dans la classe Table soit effectuer une jointure grâce à la classe Jointure. Et on peut ajouter des individus grâce à la classe Jointure.

Si l’on souhaite retirer des données, On peut le faire grâce aux classes Selection\_variable et Fenetrage. Selection\_variable permet de choisir quelles variables on souhaite conserver et donc supprimer les autres, et Fenetrage permet de choisir quelles observations garder et donc de supprimer celles restantes. Ces deux classes permettent de supprimer n’importe quelle donnée que l’on ne souhaiterait plus voir apparaître dans la table que ce soit une variable ou une observation.

Lorsqu’on souhaite aussi appliquer d’autres modifications telles que la normalisation ou le centrage des variables. On peut également le faire grâce aux classes restantes dérivant de la classe Transformation. La classe Normalisation permet de normaliser les variables données et la classe Centrage de les centrer.

Pour ce qui est de la gestion des valeurs manquantes, nous avons laissé le choix à l’utilisateur en lui mettant à disposition plusieurs possibilités. Il peut soit décider de supprimer toutes les observations présentant des valeurs manquantes pour les variables qui l’intéresse, soit remplacer ces valeurs manquantes par des 0, ou s’il le souhaite par la moyenne.

L’utilisateur peut également enregistrer les données une fois modifié grâce à la méthode export présente dans la classe Table.

Les principaux tests unitaires qu’il faudra réaliser sont le bon fonctionnement de l’importation pour les 2 types de fichiers, JSON et CSV. Il faudra également tester toutes les opérations par rapport au type de la variable sur laquelle on souhaite effectuer ces opérations, par exemple vérifier que le programme renvoie bien que ce n’est pas possible pour une variable de type caractère pour une moyenne ou que ce n’est pas possible car la variable ne contient que des valeurs manquantes.

Les tests fonctionnels se font grâce au fichier test, c’est-à-dire qu’on teste en important les données et en faisant différentes manipulations que cela renvoie ce qu’on veut et qu’il n’y ai pas de problèmes.