JAMES BÉLISLE

Programmation 3

420-4A6-VI gr. 01

TP – Voisins

Travail présenté à

M. Frédérik Taleb

Département d’informatique

Cégep de Victoriaville

Le 5 Février 2025

Projet Voisins

# Classe RessourceLoader

## DonnerLignes() et PrendreLignes()

Pour faire ces méthodes, j’ai dû regarder dans la documentation sur le site de [CPlusPlus](https://cplusplus.com/reference/fstream/ifstream/) après avoir entendu parler de la bibliothèque ifstream par un camarade de classe. Une des méthodes que j’ai vues pour lire un fichier avec cette méthode est sur le site [Geeks for Geeks](https://www.geeksforgeeks.org/read-file-using-ifstream-in-cpp/), bien que la méthode que j’ai utilisée s’apparente plus à celle utilisée dans le site [W3 schools](https://www.w3schools.com/cpp/ref_fstream_ifstream.asp). J’ai ensuite décidé de faire deux méthodes différentes : une pour calculer le nombre de lignes et une qui retournera le contenu des lignes dans un vecteur. Puisque les deux méthodes ont besoin d’un aspect du fichier seulement accessible avec ifstream. Pour la méthode PrendreLigne(), j’ai utilisé un compteur int que j’incrémentait de un à chaque ligne. La méthode DonnerLigne(), prends non seulement le nom du fichier recherché en string, mais aussi le numéro de ligne de départ et la quantité de lignes à prendre en données. Puisqu’il est impossible de sauter à un certain nombre de lignes, j’ai préféré utiliser un compteur de ligne et un if qui pourra enregistrer les données lorsque le compteur est entre le numéro de ligne de départ et la somme du numéro de ligne de départ et le nombre de lignes à enregistrer. Ainsi, elle sera compatible avec les méthodes getTrainData() et getTestData() qui ne demandent qu’une partie du document, donc des parties du document contenues entre certaines lignes.

## getTrainData()

Cette méthode a besoin de convertir la valeur de k, qui est un pourcentage, en un nombre de lignes à convertir du document. Pour ce faire, je dois convertir k en une valeur de lignes du document. J’utilise donc une variable qui contiendra le nombre total de lignes du document grâce à la méthode PrendreLignes(), puis je multiplie k par cette valeur pour ensuite diviser par 100, ce qui donnera le nombre de lignes à aller chercher. Il ne reste plus qu’à déclarer un vecteur de type string qui ira utiliser la méthode DonnerLignes() pour son contenu. Et pour conclure, il utilisera la méthode ClasserDonnees() pour séparer les données en celles dont il a besoin.

## getTestData()

Cette méthode a besoin de convertir la valeur de k(un pourcentage) en l’inverse du pourcentage, il faut donc lui soustraire 100 (getTestData prenant la deuxième partie du document, il faut donc avoir le numéro de ligne à la fin de la première partir pour commencer la deuxième) et le rendre positif en le multipliant par -1, et finalement le multiplier par le nombres de lignes présentes dans le document.

## mettreEnNoeud()

Cette méthode utilise le principe de chaînage comme montré ici : [Lien](https://www.geeksforgeeks.org/chain-of-pointers-in-c-with-examples/) pour mettre les lignes du document dans un nœud puis en créer un autre avec le lien vers le nouveau nœud dans le nœud précédent, à la répétition. La méthode retournera l’objet m\_premier pour permettre à l’utilisateur de la méthode d’avoir accès à la liste chaînée.