UNIVERSITE DE MONTREAL

TRAVAIL PRATIQUE #3

PAR

YOANN DEGUIRE

DEPARTEMENT D'INFORMATIQUE

TRAVAIL PRESENTE POUR SATISFAIRE AUX EXIGENCES DU COURS IFT1169-A-A24 – C++ AVANCÉE DANS LE CADRE DU CERTIFICAT EN INFORMATIQUE APPLIQUE

DÉCEMBRE 2024

Gestionnaire d’interface

Dans la même lignée que mon TP2, j'ai décidé d'écrire une classe GestionnaireInterface qui permettrait de regrouper les fonctions d'affichage et qui ferait appel à la classe LecteurFichier ainsi qu'au GestionnaireEmployes. En ce sens, j'ai commencé par afficher la sélection des fichiers de lecture afin de m'assurer que l'utilisateur puisse entrer les bons fichiers et les valider dès le départ. Par la suite, on accède au menu principal où j'ai divisé les trois principales fonctionnalités de l'application : rechercher, supprimer et ajouter. Chacune de ces fonctions est implémentée dans différents boutons qui ouvrent des modals permettant d'entrer les informations nécessaires.

Pour faciliter l'ajout et la suppression de données, j'ai implémenté une option déroulante (dropdown) permettant de sélectionner un employé par son nom de famille, ce qui réduit ainsi les erreurs. Ensuite, j'ai ajouté les menus dans la barre du haut : le menu "Fichier" pour lire et sauvegarder des fichiers, ainsi que le menu principal avec le bouton "À propos" qui affiche un modal d'informations.

Fenêtre multitâche

Pour réduire la répétition dans mon code, j'ai créé deux fonctions qui prennent plusieurs paramètres en arguments. Cela me permet ainsi d'afficher des informations différentes avec des chaînes de caractères, mais également d'utiliser une action différente en passant en paramètre une fonction, et donc, une méthode de la classe GestionnaireEmployes. Pour l'ajout et la suppression de propriétés, j'ai également ajouté un vector de type générique afin de permettre de passer soit les managers, soit les programmeurs, et ainsi construire l'option déroulante (dropdown) avec leurs noms.

Sauvegarder un fichier

Pour sauvegarder les fichiers, j’ai essentiellement repris les méthodes de lecteurFichier qui lissait les fichiers pour simplement inverser la logique et faire une sortie. Ainsi, on peut écrire dans un fichier, la seule différence c’est plutôt que boucler dans les lignes on doit boucler dans le vector pour chaque membre de la liste.