SAÉ.S01 2022-2023 : Implémentation d'un besoin client

Sujet : Il s'agit de construire une application de gestion d'un annuaire numérique pour une organisation qui possède un ensemble de données sur ses clients (nom, prénom, code postal, ville, numéro téléphone, adresse mél, profession). Ces données sont stockées dans un fichier au format texte csv (*comma-separated values*) avec éventuellement des champs manquants (vides ou constitués uniquement de caractères séparateurs (espace ' ' ou tabulation '\t'); le caractère fin de ligne '\n' ne sera utilisé au maximum qu'une fois en fin de ligne). Chaque ligne du fichier correspond à un client, et les informations d'un client sont toujours données dans le même ordre. La virgule ',' est un caractère utilisé exclusivement pour séparer les champs. Lors de la lecture d'une ligne, il faudra donc parcourir ses caractères pour extraire les informations de chaque champ en se servant du caractère séparateur pour terminer la lecture d'un champ et commencer la suivante. Deux contraintes sont imposées, le champ adresse mél d'un client ne pourra pas être vide et une adresse mél ne pourra être utilisée que pour un seul client. Ainsi, l'adresse mél jouera un rôle similaire à celui d'une clef primaire en base de données et permettra d'identifier de manière unique un client. Un fichier annuaire respectant toutes ces contraintes est dit valide. L'application doit être capable de lire et modifier ce fichier tout en garantissant sa cohérence (respect de toutes les contraintes). En outre, elle doit offrir des fonctionnalités pour une consultation structurée et lisible des données. Parmi ces fonctionnalités, l'organisation souhaite retrouver :

- L'ajout de nouveaux clients (sans créer de doublons)
- La modification des données sur un client
- La suppression d'un client
- L'affichage de la liste de tous les clients, avec la possibilité de trier cette liste sur le nom
- Un filtre sur le nom, le prénom, le code postal, la ville ou la profession permettant d'afficher la liste des clients répondant au critère donné. Par exemple, on peut vouloir afficher la liste des clients dont le prénom commence par la lettre "J" ou dont le nom est "Leroy" ou encore dont la profession est "enseignant"
- L'affichage de la liste de tous les clients pour lesquels des données sont manquantes dans l'annuaire
- La sauvegarde des données dans un fichier (on pourra distinguer l'écrasement du fichier original et l'écriture dans un nouveau fichier)

L'interface de l'application se limitera à un affichage sur une console, mais vous pourrez optionnellement créer une interface graphique une fois le développement des fonctionnalités nécessaires terminé.

Il vous est demandé de mettre en place une interface de programmation d'application (API) permettant de simplifier les interactions avec votre application. Elle permettra, entre autres, d'automatiser le test des fonctionnalités implémentées. Ainsi, il faudra écrire une fonction associée à chaque fonctionnalité dont la spécification formelle est donnée ci-dessous. Cette interface se chargera d'interagir avec votre application pour réaliser les traitements attendus.

- **ajouter_client(nom_annuaire, nom_p, prenom_p, code_postal_p, ville_p, telephone_p, mel_p, profession_p)**

:**entrées** : **nom_annuaire, nom_p, prenom_p, code_postal_p, ville_p, telephone_p, mel_p, profession_p** : chaînes de caractères non modifiables

:**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide et **mel_p** n'est pas une chaîne vide

:**sorties** : **resultat_ajouter.txt** : fichier texte (dans le même répertoire que l'exécutable)

:**post-conditions** : **resultat_ajouter.txt** est un fichier annuaire valide qui contient les données des clients de l'annuaire en entrée plus les données du nouveau client à la fin du fichier si **mel_p** n'est pas déjà présent dans l'annuaire en entrée, sinon **resultat_ajouter.txt** est un fichier vide

- **modifier_mel_client(nom_annuaire, mel_p, nv_mel_p)**

  :**entrées** : **nom_annuaire, mel_p, nv_mel_p**: chaînes de caractères non modifiables

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide, **mel_p** et **nv_mel_p** ne sont pas des chaînes vides. **mel_p** est bien présent dans l'annuaire, alors que **nv_mel_p** ne l'est pas

  :**sorties** : **resultat_modifier_mel.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :**post-conditions** : **resultat_modifier_mel.txt** est un fichier annuaire valide dont le contenu correspond à celui de l'annuaire en entrée dans lequel **mel_p** est remplacé par **nv_mel_p**

- **modifier_autres_que_mel_client(nom_annuaire, mel_p, nom_champ, nv_valeur)**

  :**entrées** : **nom_annuaire, mel_p, nom_champ, nv_valeur** : chaînes de caractères non modifiables

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide. **mel_p** n'est pas une chaîne vide et correspond au mél d'un client présent dans l'annuaire. **nom_champ** est le nom d'un champ parmi (nom, prénom, code postal, ville, numéro téléphone, profession)

  :**sorties** : **resultat_modifier_autre_que_mel.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :**post-conditions** : **resultat_modifier_autres_que_mel.txt** est un fichier annuaire valide dont le contenu correspond à celui de l'annuaire en entrée dans lequel la

valeur du champ correspondant à **nom_champ** pour le client ayant pour mél **mel_p** est remplacée par **nv_valeur**

- **supprimer_client(nom_annuaire, mel_p)**

  :**entrées** : **nom_annuaire, mel_p** : chaînes de caractères non modifiables

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide. **mel_p** n'est pas une chaîne vide et correspond au mél d'un client présent dans l'annuaire

  :**sorties** : **resultat_supprimer.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :**post-conditions** : **resultat_supprimer.txt** est un fichier annuaire valide dont le contenu correspond à celui de l'annuaire en entrée dans lequel le client ayant pour mél **mel_p** a été supprimé

- **trier_clients_par_nom(nom_annuaire)**

  :**entrées** : **nom_annuaire** : chaîne de caractères non modifiable

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide

  :**sorties** : **resultat_afficher_tries_nom.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :**post-conditions** : **resultat_trier_par_nom.txt** est un fichier annuaire valide dont le contenu correspond aux clients de l'annuaire en entrée triés de manière croissante sur le nom (ordre du dictionnaire ou lexicographique)

- **filtrer_clients_donnees_manquantes(nom_annuaire)**

  :**entrées** : **nom_annuaire** : chaîne de caractères non modifiable

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide

  :**sorties** : **resultat_afficher_donnees_manquantes.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :**post-conditions** : **resultat_filtrer_donnees_manquantes.txt** est un fichier annuaire valide dont le contenu correspond aux clients de l'annuaire en entrée pour lesquels au moins un champ est vide

- **filtrer_un_champ(nom_annuaire, nom_champ, val_chaine)**

  :**entrées** : **nom_annuaire, nom_champ, val_chaine** : chaînes de caractères non modifiables

:**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide. **nom_champ** est le nom d'un champ parmi (nom, prénom, code postal, ville, numéro téléphone, adresse mél, profession). **val_chaine** n'est pas une chaîne vide

:**sorties** : **resultat_filtrer_un_champ.txt** : fichier texte (dans le même répertoire que l'exécutable)

:**post-conditions** : **resultat_filtrer_un_champ.txt** est un fichier annuaire valide dont le contenu correspond aux clients de l'annuaire en entrée pour lesquels le champ **nom_champ** contient la chaîne **val_chaine**

- **filtrer_combiner_deux_champs(nom_annuaire, nom_champ1, val_chaine1, nom_champ2, val_chaine2)**

  :**entrées** : **nom_annuaire, nom_champ1, val_chaine1, nom_champ2, val_chaine2** : chaînes de caractères non modifiables

  :**pré-conditions** : **nom_annuaire** est le nom d'un fichier annuaire valide. **val_chaine1** et **val_chaine2** ne sont pas des chaînes vides. **nom_champ1** et **nom_champ2** sont des noms de champs parmi (nom, prénom, code postal, ville, numéro téléphone, adresse mél, profession) et sont différents

  :**sorties** : **resultat_filtrer_combiner_deux_champs.txt** : fichier texte (dans le même répertoire que l'exécutable)

  :post-conditions : **resultat_filtrer_combiner_deux_champs.txt** est un fichier annuaire valide dont le contenu correspond aux clients de l'annuaire en entrée pour lesquels les champs **nom_champ1** et **nom_champ2** contiennent respectivement les chaînes **val_chaine1** et **val_chaine2**

Modalités : Ce sujet est à réaliser en binôme ou trinôme dont la composition est fixée par les enseignants.

Jeu d'essai : Un fichier annuaire vous sera fourni au début du projet afin que vous puissiez effectuer des tests. Il est conseillé de vous en inspirer pour générer d'autres fichiers afin de constituer un jeu d'essai complet.

Jeu de validation : Un fichier annuaire vous sera fourni au moment de l'évaluation pour valider les fonctionnalités développées.

Délivrable : Code de l'application + API (interface de programmation d'application) (à rendre au plus tard le 08/01 sur Tomuss).

L'évaluation portera tout d'abord sur les fonctionnalités développées. Cependant, toutes les fonctionnalités n'ont pas le même niveau d'importance ou de complexité. Le **chargement** des

données d'un fichier et leur **sauvegarde** dans un nouveau fichier sont "critiques" car elles sont fondamentales pour le bon fonctionnement de l'application. Les membres du groupe devront collaborer pour leur mise au point. Les autres fonctionnalités peuvent être divisées en 3 groupes de complexité :

- Difficile : les filtres
- Moyen : l'ajout d'un client, la suppression d'un client, la modification des données d'un client, le tri des clients sur le nom
- Facile : l'affichage des clients, l'affichage des clients avec des données manquantes

La réalisation de chacune de ces autres fonctionnalités sera confiée à un membre du groupe en veillant à une répartition équitable notamment en termes de complexité et de diversité. Il faudra donc préciser l'auteur pour chaque fonction du code fourni. Une note individuelle sera attribuée sur la base des fonctionnalités développées par chaque membre.

L'évaluation concernera également la qualité de l'implémentation, à savoir la modularité du code (définition de fonctions, séparation entre déclaration et définition, compilation séparée), le choix des structures de données, le choix d'une norme pour la compilation, le traitement des messages du compilateur (notamment le nombre de messages d'avertissement), le nommage des variables et fonctions, les spécifications formelles des fonctions et les autres commentaires du code, la mise en place de tests unitaires en boîte noire pour les fonctions développées. On s'attardera également sur l'utilisation de git.

Enfin, une note de validation évaluera la qualité de la présentation des fonctionnalités implémentées et leurs tests sur le jeu de validation. L'auteur d'une fonctionnalité sera éventuellement interrogé sur celle-ci lors de l'évaluation.

Barèmes (toutes les notes seront individualisées) :

- Fonctionnalités implémentées en tenant compte de leur complexité (sur 10pts)
- Qualité de l'implémentation et de l'API (sur 5pts)
- Présentation et test des fonctionnalités sur un jeu de validation (sur 5pts) → semaine 09/01