Générer une liste d'apprentis pour des tests

Objectifs

- Accéder à un fichier texte en mode lecture ou écriture.
- Lire et traiter le contenu d'un fichier texte

Spécifications, contenus

- Vous créez un programme (*fichier040.py*) avec l'en-tête de l'EPTM.
- But du programme :
  - Dans le cadre d'une implémentation d'une base de données, il est fréquemment nécessaire d'utiliser des jeux de données pour des tests. Si ces données sont proches de la réalité, c'est un plus.
  - Aussi, le but de ce programme est de fournir une liste d'apprentis (nom, prénom, npa, localite) sur la base de fichiers sources de type texte.

- Fichiers sources disponibles :
  - La liste des localités de Suisse dans le fichier *be-t-00.04-agv-01.xlsx*. Cette liste est tirée du site internet de la confédération.
  - Une liste de prénoms usuels dans le fichier *prenoms.xlsx*.
  - Une liste de noms de famille usuels dans le fichier *noms.xlsx*.

- Actions du programme :
  - Demander à l'utilisateur le nombre d'éléments que devra contenir la liste.
  - Générer une liste d'un certain nombre d'apprentis (nom, prénom, npa, localité) sans redondance en ce qui concerne le nom et le prénom.
    - Les localités doivent être uniquement valaisannes.
  - Afficher cette liste à l'écran, en gérant l'affichage par blocs d'apprentis.
    - Par exemple, afficher les apprentis par bloc de 10 unités, avec une action spécifique de l'utilisateur pour poursuivre l'affichage ou le stopper.
  - Enregistrer dans un fichier de destination, de type .csv (utf-8) la liste d'apprentis générée, pour autant que l'utilisateur le demande.
    - L'utilisateur doit pouvoir spécifier le nom du fichier de destination. Toutefois, l'utilisation d'un fichier par défaut doit être possible.
    - L'utilisateur doit pouvoir confirmer ou renoncer à l'enregistrement si le fichier existe déjà.
  - Mettre en place un interface utilisateur (mode console) ergonomique pour ces actions.

- Modalités d'exécution :
  - Respecter les best-practices de l'école, notamment :
    - Utilisation systématique de noms explicites pour les variables, listes et objets utilisés.
    - Commenter dynamiquement les prototypes de fonction.
    - …
  - Généraliser de manière optimale du code.
  - Partager les fonctionnalités du programme en fonctions qui ont du sens :
    - Une seule fonctionnalité par fonction.
    - Permettre la lecture aisée de votre code par un informaticien.
    - Une fonction *main()* pour le programme lui-même.
  - Proposer un interface utilisateur ergonomique :
    - En mode console.
    - Informer clairement l'utilisateur sur les actions possibles et en cours.
    - Rendre intuitive l'utilisation du programme.
  - Mettre en place les éléments de contrôle de sécurité nécessaires pour éviter les erreurs système.
    - Rendre conditionnelles les actions à risques par des tests adéquats.
    - Intercepter les erreurs systèmes si les contrôles précédents sont impossibles.