S04 – TP1 (15%)

Gestion des fichiers

**(équipe de 2) Vos noms et DA :**

**-**

**-**

Objectif :

Pour ce TP, vous allez développer des scripts Python 3 visant à automatiser des tâches liées à la gestion de fichiers sous Linux et Windows.

# Nettoyage de Clé USB ( / 50 pts)

## Mise en situation :

Le cégep souhaite faciliter la réutilisation des clés USB en automatisant leur nettoyage et leur sauvegarde. Vous devez développer un programme qui :

* Détecte automatiquement la connexion d'une nouvelle clé USB.
* Sauvegarde et/ou supprime les fichiers présents sur la clé en fonction des options choisies.
* Enregistre un historique des modifications effectuées.
* Peut traiter plusieurs clés sans nécessiter un redémarrage.

Votre code doit être documenté de manière claire et répondre à la question "Pourquoi ?". Les noms des variables et des fonctions doivent être explicites pour minimiser les commentaires superflus.

## Votre script ( /40 pts) :

1. Le script doit accepter les arguments suivants (5 pts) :

* –sauvegarder : Drapeau booléen par défaut à vrai. active la sauvegarde des fichiers dans un fichier ZIP.
* –effacer : Drapeau booléen par défaut à faux. supprime tous les fichiers de la clé USB si activé.

1. Le script doit détecter automatiquement la connexion d'un nouvel appareil USB. (5 pts)
   * afficher un message lors de la connexion
2. Ajouter dans un fichier log.txt sur une nouvelle ligne : le timestamp, le nom de la clé et les opérations qui seront effectué. (5 pts)
   * Le fichier log.txt doit être créé (ou mis à jour s'il existe déjà).
3. Lorsqu'une clé USB est connectée, le script doit (15 pts) :
   * Créer une archive ZIP contenant les fichiers de la clé, stockée dans un dossier de sauvegarde à la racine du projet (uniquement si --sauvegarder est activé).
   * Supprimer les fichiers de la clé USB si l'option --effacer est activée.
     1. Le programme doit s'assurer que :
        1. Les dossiers nécessaires sont créés automatiquement s'ils n'existent pas encore.
        2. Les erreurs de création de dossier sont bien gérées.
4. Un fichier historique.csv doit être maintenu à la racine du projet. Chaque sauvegarde doit y être ajoutée sous forme de ligne avec les colonnes suivantes (10 pts) :
   * Nom de l’archive
   * Nom du fichier sauvegardé
   * Timestamp de la sauvegarde
     1. Le fichier doit être mis à jour en mode ajout (append), sans écraser les données existantes.
5. Une fois les opérations terminées, le programme retourne en attente d’une nouvelle clé USB, sans nécessiter de redémarrage. (5 pts)

Contrainte :

* Les dossiers et fichiers doivent être créés sans erreur s’ils ne sont pas déjà créés, sinon vous travailler en mode ajout
* Afficher des messages en console à chaque étape pour suivre l'exécution (exemple : "Clé USB détectée", "Copie terminée", "Suppression effectuée", etc.).

# Vos tests : ( / 5 pts)

* Vous devez faire un plan de tests complet. Cela veut dire que chaque facette (bout de code) de votre script doit être testée. Ceci inclut la gestion des erreurs.

**Remplir votre plan de tests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #Test | Description (ce qui est testé) | Mise en place initiale (les fichiers initiales) | Résultat attendu | Imprime écran de l’exécution du script |
| 1 | Une clé USB est connectée | Fichier dans la clé.  Sauvegardé et effacer à faux |  | Mettre votre imprime écran |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Gestion de Fichiers et Dossiers en POO ( / 40 pts)

## Consignes générales

1. Implémentez toutes les classes dans un même fichier Python.
2. Utilisez les principes de la Programmation Orientée Objet (POO), notamment l'héritage et l'abstraction.
3. Ajoutez un bloc if \_\_name\_\_ == "\_\_main\_\_": pour tester votre code.
4. Votre script doit être fonctionnel sur Windows et sur Linux.

## Classe de base : Item ( /5 pts)

Créez une classe Item qui servira de classe de base pour les fichiers et dossiers.

* Attributs :
  + nom : nom de l’item
  + chemin\_parent : le répertoire où il se trouve (ex : \home\user\TP1\documents\)
  + date\_creation : quand l’item a été créé sur le disque
* Contraintes :
  + Lors de l'instanciation, vérifiez si le fichier ou le dossier existe sur le disque. Si ce n'est pas le cas, levez une exception personnalisée.
* Méthodes
  + Méthode abstraite ouvrir(self) :
    1. Dois être implémentée dans les classes dérivées.
    2. Lance une exception NotImplementedError si elle est appelée directement depuis Item.

## Classe Fichier ( /5 pts)

Créez une classe Fichier qui hérite de Item.

Attributs :

* hérite de tous les attributs de Item.
* extension : l'extension du fichier (txt, docx, py, etc.).

Méthodes :

* Implémentez la méthode ouvrir(self), qui permet d'ouvrir le fichier avec le bloc-notes ou l'éditeur par défaut du système d'exploitation.

## Classe Dossier ( /20 pts)

Créez une classe Dossier qui hérite de Item.

Attributs :

* Hérite de tous les attributs de Item.

Méthodes :

* Implémentez la méthode ouvrir(self), qui ouvre le dossier avec l'explorateur de fichiers.
* Implémentez retirer\_doublons(self), qui supprime tous les fichiers du dossier ayant un contenu identique.
  + Exemple : si fichier1.txt, fichier2.txt et fichier3.txt contiennent tous le mot "Bonjour", alors fichier2.txt et fichier3.txt doivent être supprimés.

## Programme principal : \_\_main\_\_ ( /10 pts)

Ajoutez un bloc if \_\_name\_\_ == "\_\_main\_\_": pour tester les fonctionnalités implémentées. Votre programme doit :

* Créer des objets Fichier et Dossier avec différents chemins.
* Tester la levée de l'exception personnalisée en essayant d'instancier un fichier ou un dossier inexistant.
* Tester les méthodes ouvrir() et retirer\_doublons().
* **Votre script doit être fonctionnel sur Windows et sur Linux.**

**\*\*Vous devez mettre au moins un imprime écran du résultat de l’exécution du main ci-dessous\*\***

Barème de correction :

* Classe Item bien définie et exception correctement gérée (10 pts)
* Héritage et abstraction bien respectés dans Fichier (10 pts)
* Héritage et abstraction bien respectés dans Dossier, et retirer\_doublons fonctionne correctement (20 pts)
* Bloc \_\_main\_\_ présent et permettant de tester toutes les fonctionnalités (10 pts)

# Remise :

* Ce document avec les imprimés-écrans et **Renommé** avec DA dans le nom du document.
* Vos deux fichier python qui contient tout le code de votre TP1.
* Remise sur Teams
* **NE PAS** compresser vos fichiers dans la remise