ACTIVITE 4 : POO

L'objectif de cette activité est de définir et d'utiliser trois *classes* correspondant respectivement à un temps ou une durée (sous la forme heures-minutes-secondes), à une chanson et à un album.

Vous utiliserez ensuite ces classes pour réaliser la génération d'albums souhaitée. Vous travaillerez par **groupes de deux.**

Les chansons sont fournies dans un fichier contenant des lignes de texte:

```
"TITRE,AUTEUR,MIN:SEC"
```

où "TITRE", "AUTEUR", "MIN" et "SEC" correspondent respectivement au titre, à l'auteur, au nombre de minutes et de secondes de la chanson, par exemple:

```
Stand Up,Cynthia Erivo,3:58
Hallelujah , Jeff Buckley,6:0
The Sound Of Silence ,Disturbed,4:22
```

Ce format facilite leur lecture. On vous rappelle que la méthode split(« , ») permet de séparer un string contenant des « , » en mots individuels.

Au sein de votre binôme, un étudiant pourra se concentrer sur le développement d'une classe tandis que l'autre préparera des tests unitaires.

Les spécifications des méthodes étant donnée par avance, il est tout à fait possible de préparer des tests en même temps que la classe à tester.

Etapes

1. Chargez le fichier "chansons.txt". Ce fichier contient une liste de chansons, au format décrit ci-dessus.
2. Créez une nouvelle classe **Duree** qui représente une durée exprimée en heures (h), minutes (m) et secondes (s).
3. Créez une série de tests unitaires que vous utiliserez pour tester la classe Duree. Ajoutez des tests au fur et à mesure que vous ajouterez des méthodes à la classe Duree. Vous devez avoir un ou plusieurs tests, avec différents scénarios de test, par méthode de la classe Duree. Par exemple, si votre classe Duree a une méthode ajouter(), vous pourrez implémenter une fonction test_ajouter() pour tester les différents scénarios d'utilisation de la méthode ajouter() de la classe Duree.

4. Dans la classe **Duree**, définissez une méthode d'initialisation avec comme paramètres __init__(self, h, m, s), qui requiert que m et s soient dans l'intervalle [0..60[et crée un nouveau temps de h heures, m minutes et s secondes. Veillez à la spécification et vérifiez que les conditions requises soient respectées.
5. Dans la classe **Duree**, ajoutez des méthodes correspondant aux spécifications suivantes:

```
def toSecondes(self) :
    """
    Retourne le nombre total de secondes de cette instance de Duree (self).
    """


def delta(self,d) :
    """
    Retourne la différence entre cette instance de Duree (self) et la Duree d passée en paramètre,
    en secondes (positif si ce temps-ci est plus grand).
    """


def apres(self,d):
    """
    Retourne True si cette instance de Duree (self) est plus grand que la Duree d passée en paramètre;
    retourne False sinon.
    """


def ajouter(self,d):
    """
    Ajoute une autre Duree d à cette instance de Duree (self).
    Corrige de manière à ce que les minutes et les secondes soient dans l'intervalle [0..60[,
    en reportant au besoin les valeurs hors limites sur les unités supérieures
    (60 secondes = 1 minute, 60 minutes = 1 heure).
    """


def __str__(self):
    """
    Retourne cette durée sous la forme de texte "heures:minutes:secondes".
    Astuce: la méthode "{:02}:{:02}:{:02}".format(heures, minutes, secondes)
```

```
    retourne le String désiré avec les nombres en deux chiffres en ajoutant
    les zéros nécessaires.
    """
```

6. Faites en sorte que les attributs de la classe **Duree soient privés**
7. Ecrivez des fonctions test pour chacune de ces méthodes de la classe Duree, exécutez-les et assurez-vous que les résultats sont conformes aux spécifications, en corrigeant au besoin.
8. Créez une nouvelle classe **Chanson** représentant une chanson, caractérisée par un titre t (string), un auteur a (string) et une durée d (Duree). Définissez une méthode d'initialisation avec comme paramètres __init__(self, t, a, d).
9. Dans la classe **Chanson**, implémentez une méthode de conversion __str__ correspondant à la description suivante:

```
def __str__(self):
    """
    Retourne un String décrivant cette chanson sous le format "TITRE - AUTEUR - DUREE".
    Par exemple: "Let's_Dance - David_Bowie - 00:04:05"
    """
```

10. Dans la classe **Chanson**, implémentez une méthode de comparaison __lt__ correspondant à l'opérateur de comparaison « < » et dont la description est:

```
def __it__(self,chanson):
    """
    Retourne la chanson dont la durée est la plus petite
        """
```

11. Créez une série de tests que vous utiliserez pour tester la classe Chanson.
12. Créez une classe **Album** représentant un album contenant une ou plusieurs chansons.
13. En plus d'une méthode d'initialisation pour créer un album vide, cette classe contient une méthode add(self,chanson) pour ajouter une chanson à un album. Cette méthode retourne False si lors de l'ajout d'une chanson l'album a atteint 10 chansons ou la durée dépasserait 75 minutes. Sinon la chanson est rajoutée et la méthode add retourne True.
14. Dans la classe **Album**, implémentez également une méthode de conversion __str__ pour imprimer la description d'un album selon le format suivant:

```
Album 14 (5 chansons, 00:22:01)
01: White_Wedding - Billy_Idol - 00:04:12
02: Stand_And_Deliver - Adam_&_The_Ants - 00:03:33
03: You_Spin_Me_Around - Dead_Or_Alive - 00:03:14
```

```
04: Wired_For_Sound - Cliff_Richard - 00:03:38
05: Some_Like_It_Hot - The_Power_Station - 00:03:45
```

15. Créez une série de tests pour tester la classe Album.
16. Complétez votre programme par une série d'instructions qui:
    1. lit dans le fichier "chansons.txt", ligne par ligne, des descriptifs de chanson au format décrit en tête de cette section;
    2. pour chaque ligne lue, construite une instance de Chanson;
    3. stocker ses chansons dans un album;
    4. lorsque le nombre de chansons stockés dans un album a atteint 10 chansons ou la durée dépasserait 75 minutes, imprime à la console un descriptif de cet album avec les chansons accumulées, suivant l'exemple donné ci-dessous;
    5. poursuit la lecture du fichier et l'ajout des chansons dans un album suivant;
    6. répète ces étapes jusqu'à ce que le fichier chansons soit vide et imprime le dernier album.