Prise de notes

On souhaite développer l'application **MyNotes** destinée à éditer et gérer un ensemble de notes (des mémos) qui peuvent contenir du texte et/ou des images. Une note peut par exemple correspondre au compte-rendu d'une réunion ou à des notes prises lors d'une séance de cours.

Dans cet exercice, il s'agit d'implémenter les classes suivantes.

1 - Définir la classe `Note`

D'un point de vue implémentation, une note est un objet instance de la classe `Note`.

Tout objet `Note` est caractérisé par un attribut `titre` de type `string`. L'unique constructeur `__init__()` de cette classe a un paramètre qui permet d'initialiser cet attribut. On utilisera un getter() et setter() pour accéder aux attributs de l'objet.

La méthode `print` permettra d'afficher l'attribut `titre`

2 - Définir la classe `Article`

Les objets de la classe `Article` sont des objets `Note` qui représentent des notes contenant uniquement du texte.

En plus de l'attribut `titre`, les objets `Article` possèdent un attribut `texte` de type `string`. On utilisera un getter() et setter() pour accéder aux attributs de l'objet.

L'unique constructeur `__init__()` de cette classe a deux paramètres qui permettent d'initialiser le titre et le texte d'un article.

La méthode `print` permettra d'afficher en plus du `titre`, le `texte` d'un article

3 - Définir la classe `Image`

Les objets de la classe `Image` sont des objets `Note` qui représentent des images ou des photos.

En plus de l'attribut `titre`, les objets `Image` possèdent un attribut `description` de type `string` (qui contient des informations sur le contenu de l'image) et un attribut fichier de type `string` indiquant le chemin du fichier physique de l'image. On utilisera un getter() et setter() pour accéder aux attributs de l'objet. Dans le setter() on vérifiera que le fichier existe en utilisant la méthode `os.path.isfile()`.

L'unique constructeur de cette classe `__init__()` a trois paramètres qui permettent d'initialiser le titre, la description et le fichier d'une image.

La méthode `print` permettra d'afficher en plus du `titre`, la `description` ainsi que l'`image`. On utilisera la librairie `Pil` pour afficher une image.

```
# Exemple d'utilisation de PIL.Image
img = PIL.Image.open("/temp/image.jpg")
img.show()
```

4 - Définir la classe `Document`

On décide aussi de créer la classe `Document` dont les instances sont des objets `Note` qui regroupent virtuellement plusieurs autres objets `Note` afin de constituer un contenu de taille plus importante.

Pour cela, un objet `Document` possède un attribut `grp_notes` contenant les objets `Note` regroupés par le document.

L'unique constructeur **init**() de cette classe a un paramètre qui permet d'initialiser le titre du document.

Initialement, un objet `Document` ne regroupe aucun objet `Note`. La méthode `ajouter_note()` permet d'ajouter une note de n'importe quel type (`Article` , `Image` , `Document`) au document. La méthode `supprimer_note()` permet de supprimer une note de n'importe quel type (`Article` , `Image` , `Document`) au document. Quand un objet `Document` reçoit le message `print_document()` , son titre est affiché suivi des différentes notes qui forment le document. La méthode `print()` permet d'afficher les informations d'un objet Note. **La définition de cette méthode dépend du type de note**. On distinguera quatre types de notes particulières : `Note` , `Article` , `Image` et `Document` . Pour terminer, on ajoutera un itérateur qui nous permettra d'itérer sur chacun des éléments contenus dans le document.

5 - Main

Écrire le module principal permettant d'entrer différentes notes dans un `Document` et de les visualiser.