

Exercices programmation objet

Exercice 1 : des dates

Définir une classe `Date` pour représenter une date, avec 3 attributs `jour`, `mois`, `année`

1. Ecrire son constructeur
2. Ajouter une méthode `__str__` qui renvoie une chaîne de caractères de la forme « 8 mai 1945 ». On pourra se servir d'un attribut de classe qui est un tableau donnant les douze mois de l'année. Tester en construisant des objets de la classe `Date` puis en les affichant avec **print**.
3. Ajouter une méthode `__lt__` qui permet de déterminer si une date `d1` est antérieure à une date `d2` en écrivant `d1 < d2`. La tester.

En fonction de votre temps et de votre niveau d'acquisition sur la POO.

Exercice 2 : des grandes boîtes et des petites boîtes et des moyennes boîtes...

1. Créer une classe `Boite`.

Cette classe a pour attributs :

- Longueur
- Largeur
- Hauteur
- Ces trois attributs sont dans un ordre décroissant $\text{longueur} \geq \text{largeur} \geq \text{hauteur}$

Elle a pour méthodes :

- `Volume`, qui comme son nom l'indique donne le volume d'une boîte
- `RentreDans` (`autre_boite`), qui renvoie vrai si l'objet `Boite` rentre dans `autre_boite`.

2. Créer aléatoirement une liste d'une vingtaine de boîtes (on peut choisir des dimensions entre 1 et 50).
3. A l'aide d'un algorithme glouton, donner une suite de boîtes aussi grande que possible qui rentrent les unes dans les autres.

Rappel : pour trier les boîtes, on fera appel à la fonction `sorted`.

Syntaxe :

```
def cle_titre(ligne):  
    """  
    Renvoie la valeur du champ 'title' d'un enregistrement de la table  
    """  
    return ligne['title']
```

```
films_tries = sorted(mesDonnées, key = cle_particulière)
```

On peut rajouter `reverse = True` pour avoir l'ordre décroissant.

4. Pour les révisions (algorithmes à connaître pour le bac), il peut être intéressant de créer une méthode `__lt__(self)` afin de comparer deux boîtes, puis d'utiliser cette méthode pour faire le tri préalable à l'algorithme glouton ci-dessus. On utilisera bien sûr un des algorithmes à connaître, à savoir le tri par insertion et/ou le tri par sélection.