

FEUILLE D'EXERCICES : POO

Exercice 1

Définir une classe `Fraction` pour représenter un nombre rationnel. Cette classe possède deux attributs, `num` et `denom`, qui sont entiers et désignent respectivement le numérateur et le dénominateur. On demande que le dénominateur soit un entier strictement positif.

1. Ecrire le constructeur de cette classe. Le constructeur doit lever une `ValueError` si le dénominateur est négatif ou nul.

Explication : une exception (par exemple `ValueError`) peut être levée lorsqu'un paramètre inadapté est donné à une fonction, voici la syntaxe :

```
if condition :  
    raise ValueError("Texte expliquant l'erreur")
```

2. Ajouter une méthode `__str__` qui renvoie une chaîne de caractères de la forme "12/35", ou simplement "12" lorsque le dénominateur vaut 1.
3. Ajouter des méthodes `__eq__` et `__lt__` qui reçoivent une deuxième fraction en argument et renvoie `True` si la première fraction est égale (resp. strictement inférieure) à la seconde.
4. Ajouter des méthodes `__add__` et `__mul__` qui reçoivent une deuxième fraction en argument et renvoie la somme (resp le produit) de ces deux fractions.
5. Tester et documenter ces méthodes.

Exercice 2

Définir une classe `Date` pour représenter une date, elle aura trois attributs :

- le `jour`, un entier qui doit être entre 1 et 31 ;
- le `mois`, un entier qui doit être entre 1 et 12 ;
- l'`année` un entier relatif.

1. Ecrire le constructeur de cette classe. Le constructeur doit lever une `ValueError` si une des paramètres fournis n'est pas valide.
2. Ajouter une méthode `__str__` qui renvoie une chaîne de caractères de la forme "14 juillet 1789". On pourra se servir d'un tableau avec les mois de l'année ...
3. Ajouter une méthode `__lt__` qui permet de déterminer si une date `d1` est inférieure strictement à une date `d2`.
4. Tester et documenter ces méthodes.

Exercice 3 (Tableaux)

Dans certains langages de programmation, comme Pascal et Ada, les tableaux peuvent être indexés sur des plages d'indices choisis par le programmeur. Par exemple, un tableau pourra être indexé de -10 à 9. Vous allez construire une classe `Tableau` pour réaliser de tels tableaux. Un objet de cette classe aura trois attributs :

- `i_min` un entier qui est le premier indice ;
- `i_max` un entier qui est le dernier indice ;
- `v` la valeur qui initialise chaque case du tableau ;

1. Écrire le constructeur de cette classe. `Tableau(-10, 9, 42)`instanciera un tableau de vingt cases indexées de -10 à 9 et toutes initialisées à la valeur 42.
2. Écrire une méthode `__len__(self)` qui renvoie la taille du tableau.
3. Écrire une méthode `__getitem__(self, i)` qui renvoie l'élément `i` du tableau `self`. De même écrire une méthode `__setitem__(self, i, v)` qui modifie l'élément `i` du tableau `self`. Ces deux méthodes doivent vérifier que l'indice `i` est bien valide et dans le cas contraire, lever l'exception `IndexError`.
4. Enfin, écrire une méthode `__str__(self)` qui renvoie une chaîne de caractères décrivant le contenu du tableau.