

Exercice 1

On considère une classe **Personnage** représentant un personnage de Jeu.
Le plateau de jeu est représenté par un repère **orthonormé à trois axes**.
La position du joueur dans le plateau est repérée par **ses attributs x, y, z**.

- 1) Ecrire un constructeur initialisant les mesures.
- 2) Ecrire les méthodes **avance**, **droite** et **saute** permettant respectivement de faire avancer, aller à droite et sauter le personnage, c'est-à-dire d'augmenter de 1 respectivement x, y et z.
- 3) Implémenter une autre méthode **coord** renvoyant les coordonnées sous forme d'un triplet.
- 4) Essayer avec : `Laura = Personnage(0, 0, 0)`

Exercice 2

Voici un programme en Python :

```
import random

class Piece :
    def alea(self) :
        return random.randint(0,1)

    def moyenne(self, n):
        tirage = [ ]
        for i in range (n) :
            tirage.append( self.alea() )
        return sum(tirage) / n

p = Piece()
print( p.moyenne(100) )
```

Expliquer en détail ce qu'il permet d'afficher

Exercice 3

On considère une classe **Carre** admettant la mesure des côtés d'un carré en attribut.

- 1) Ecrire un constructeur initialisant les mesures.
- 2) Ecrire les méthodes :
 - **perimetre**, permettant de retourner le périmètre du carré.
 - **aire** permettant de retourner son aire.
- 3) Créer des exemples

Exercice 4

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

- 1) Ecrire un constructeur de cette classe.

Le constructeur doit lever une **ValueError** si le dénominateur fourni n'est pas strictement positif.

Pour cela, on utilise :

raise :

Definition and Usage

The **raise** keyword is used to raise an exception.

You can define what kind of error to raise, and the text to print to the user.

More Examples

Example

Raise a `TypeError` if `x` is not an integer:

```
x = "hello"

if not type(x) is int:
    raise TypeError("Only integers are allowed")
```

- 2) Ajouter une méthode `__str__` qui renvoie une chaîne de caractère de la forme "12 / 13", ou simplement de la forme "12" lorsque le dénominateur vaut 1.
(`__str__`(self) est une méthode de Python : renvoie une chaîne de caractères)
- 3) Ajouter des méthodes `__eq__` et `__lt__` qui reçoivent une deuxième fraction en argument et renvoie True si la première fraction représente respectivement un nombre égal ou un nombre strictement inférieur à la fraction.
(`__lt__`(self, u) est une méthode de Python : Pour `self = t`, elle renvoie True si `t` est strictement plus petit que `u`)
(`__eq__`(self, u) est une méthode de Python : Pour `self = t`, elle renvoie True si `t` est égal à `u`)
- 4) Ajouter des méthodes `__add__` et `__mul__` qui reçoivent une deuxième fraction en argument et renvoie une nouvelle fraction représentant respectivement la somme et le produit des deux fractions.
- 5) Tester ces opérations
- 6) **Question bonus** : S'assurer que les fractions sont toujours sous forme réduite.