

Classe Object

La classe *Fraction*

Héritage (notion hors
programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

TP les fractions

Christophe Viroulaud

Terminale - NSI

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

En mathématiques une fraction est un moyen d'écrire un nombre rationnel sous la forme d'un quotient de deux entiers. La fraction $\frac{a}{b}$ désigne le quotient de a par b ($b \neq 0$). Dans cette fraction, a est appelé le *numérateur* et b le *dénominateur*¹.

1. source : wikipedia

Classe Object

La classe *Fraction*Héritage (notion hors
programme)Opérations sur les
fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Peut-on effectuer les opérations classiques sur les fractions ?

1. Classe **Object**

1.1 La classe *Fraction*

1.2 Héritage (notion hors programme)

2. Opérations sur les fractions

Classe **Object**

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Activité 1 :

1. Créer le fichier `mod_fraction.py`
2. Construire la classe **Fraction** et son constructeur. Cette classe possédera les attributs **numérateur** et **dénominateur**, deux entiers initialisés à 1.
3. Écrire les *accesseurs* et les *mutateurs*. On utilisera une assertion pour le cas de l'affectation du dénominateur nul.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Classe `Object`

La classe ***Fraction***

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe **Fraction**

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 class Fraction:
2     def __init__(self):
3         self.numerateur: int = 1
4         self.denominateur: int = 1
5
6     def get_numerateur(self) -> int:
7         return self.numerateur
8
9     def get_denominateur(self) -> int:
10        return self.denominateur
```

Code 1 – La classe et les accesseurs

Classe `Object`La classe ***Fraction***

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1  def set_numerateur(self, n: int) -> None:
2      self.numerateur = n
3
4  def set_denominateur(self, d: int) -> None:
5      assert d != 0, "Le dénominateur ne peut
6      pas être nul."
7      self.denominateur = d
```

Code 2 – Les mutateurs

1. Classe `Object`

1.1 La classe *`Fraction`*

1.2 Héritage (notion hors programme)

2. Opérations sur les fractions

Classe `Object`

La classe *`Fraction`*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Héritage (notion hors programme)

Python possède une classe native `Object` qui possède ses propres méthodes. Par exemple la méthode `__str__` renvoie une chaîne de caractère représentant l'objet. Ainsi le code 3 renvoie le résultat 4.

```
1 print(f)
```

Code 3 – La méthode est appelée automatiquement.

```
1 <__main__.Fraction object at 0x7f976feb0f0>
```

Code 4 – La méthode renvoie des informations sur l'objet (adresse mémoire...)

Classe `Object`

La classe `Fraction`

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe `Object`La classe `Fraction`Héritage (notion hors
programme)Opérations sur les
fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Quand le programmeur construit une classe, Python crée en réalité un *enfant* de la classe `Object`. On parle d'**héritage**. Ainsi la classe `Fraction` possède également la méthode `__str__`. Mieux il est possible de redéfinir le rôle de la méthode. On parle alors de **surcharge**.

Activité 2 :

1. Écrire la méthode 5 dans la classe `Fraction`.

```
1 def __str__(self):  
2     return "Bonjour je suis un objet!"
```

Code 5 – Surcharge de la méthode

2. Exécuter à nouveau `print(f)`.
3. Modifier la méthode pour qu'elle affiche la fraction sous la forme `a/b`.

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 def __str__(self):  
2     return "{}/{ {}".format(self.numerateur, self  
    .denominateur)
```

```
1 print(f)  
2 >>> 2/3
```

1. Classe Object

2. Opérations sur les fractions

2.1 Opérations mathématiques

2.2 Opérateurs Python

2.3 Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

- Multiplier deux fractions :

$$\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d}$$

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

- Multiplier deux fractions :

$$\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d}$$

- Additionner deux fractions :

$$\frac{a}{b} + \frac{c}{d} = \frac{a \times d}{b \times d} + \frac{c \times b}{d \times b} = \frac{a \times d + c \times b}{b \times d}$$

1. Classe Object

2. Opérations sur les fractions

2.1 Opérations mathématiques

2.2 Opérateurs Python

2.3 Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Pour additionner deux nombres entiers, il suffit d'utiliser l'opérateur +

```
1 2 + 3
2 >>> 5
```

Mais comment Python sait-il ce qu'il doit réaliser dans les cas suivants ?

```
1 [1, 2, 3] + [4, 5, 6]
2 "Hello " + "world!"
```

Code 6 – L'opérateur d'addition a un comportement adapté à chaque situation.

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe `Object`La classe `Fraction`

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Il est possible de redéfinir le comportement des opérations de la classe `Object` en *surchargeant* les méthodes :

- ▶ `__add__` définit l'opérateur `+`,
- ▶ `__mul__` définit l'opérateur `*`,
- ▶ `__eq__` définit l'opérateur `==`,
- ▶ `__lt__` définit l'opérateur `<`,
- ▶ ...

On peut retrouver une liste exhaustive sur <https://docs.python.org/fr/3/library/operator.html>

1. Classe Object

2. Opérations sur les fractions

2.1 Opérations mathématiques

2.2 Opérateurs Python

2.3 Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Activité 3 :

1. Redéfinir les méthodes `__eq__` et `__lt__` qui reçoivent une deuxième fraction en paramètre et renvoient `True` si la première fraction est égale ou strictement inférieure à la deuxième fraction.
2. Tester le code 7

```
1 f = Fraction()
2 f.set_numérateur(2)
3 f.set_dénominateur(3)
4
5 f1 = Fraction()
6 f1.set_numérateur(8)
7 f1.set_dénominateur(12)
8 print(f == f1)
```

Code 7 – Comparaison de fractions

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 def __eq__(self, f):
2     return self.numérateur * f.dénominateur ==
   self.dénominateur * f.numérateur
3
4 def __lt__(self, f):
5     return self.numérateur * f.dénominateur <
   self.dénominateur * f.numérateur
```

Code 8 – Méthodes de comparaison

Classe `Object`La classe `Fraction`Héritage (notion hors
programme)Opérations sur les
fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Activité 4 :

1. Redéfinir les méthodes `__add__` et `__mul__` qui reçoivent une deuxième fraction en paramètre et renvoient une nouvelle fraction représentant respectivement la somme et le produit des deux fractions.
2. Additionner et multiplier deux fractions.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 def __add__(self, f):
2     res = Fraction()
3     res.set_numerateur(self.numerateur * f.
        denominateur + f.numerateur * self.
        denominateur)
4     res.set_denominateur(self.denominateur * f.
        denominateur)
5     return res
```

Code 9 – Addition

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 def __mul__(self, f):  
2     res = Fraction()  
3     res.set_numerateur(self.numerateur * f.  
        numerateur)  
4     res.set_denominateur(self.denominateur * f.  
        denominateur)  
5     return res
```

Code 10 – Multiplication

Classe Object

La classe *Fraction*

Héritage (notion hors programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

```
1 f = Fraction()
2 f.set_numérateur(2)
3 f.set_dénominateur(3)
4
5 f2 = Fraction()
6 f2.set_numérateur(12)
7 f2.set_dénominateur(7)
8 print(f + f2)
```

Code 11 – Addition de deux fractions

Classe Object

La classe *Fraction*

Héritage (notion hors
programme)

Opérations sur les fractions

Opérations mathématiques

Opérateurs Python

Implémentation

Le code complet est accessible [ici](#).