

Exercices de Programmation & Algorithmique 1

Série 11 – Créer un nouveau type — Les nombres rationnels

(8 décembre 2021)

Département d'Informatique – Faculté des Sciences – UMONS

Pré-requis : notions de classe et d'objet; attributs; méthodes d'instances; méthodes spéciales (`__init__` et `__str__`); surcharge des opérateurs (cours jusqu'au **Chapitre 13**).

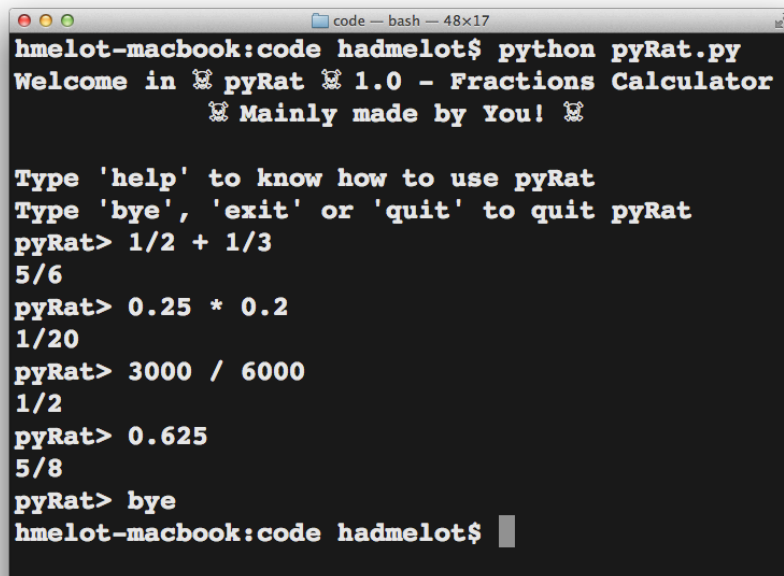
Objectifs : s'initier à la programmation orientée objet (POO); être capable de créer un nouveau type de données numériques en utilisant la surcharge des opérateurs.

1 Le contrat

1.1 Créer et manipuler des fractions – la calculatrice pyRat

Votre mission – si vous l'acceptez – est de créer un nouveau type de données numériques : des fractions. Le but est de pouvoir les utiliser en Python de manière aussi naturelle et proche de la notation mathématique que possible (comme on le fait par exemple pour des entiers grâce aux opérateurs arithmétiques). Pour ce faire, vous allez créer une classe `Rational` (à placer dans un module `rational.py`) et utiliser des notions de la programmation orientée objet comme la surcharge d'opérateurs. Concrètement, vous allez construire cette classe en suivant une série d'étapes qui vous guideront pas à pas vers la possibilité d'utiliser vos fractions dans une petite calculatrice adaptée nommée `pyRat`.

Au terme de cette séance, voici un exemple de session qui sera réalisable avec `pyRat` :



```
hmelot-macbook:code hadmelot$ python pyRat.py
Welcome in 🐍 pyRat 🐍 1.0 - Fractions Calculator
      🐍 Mainly made by You! 🐍

Type 'help' to know how to use pyRat
Type 'bye', 'exit' or 'quit' to quit pyRat
pyRat> 1/2 + 1/3
5/6
pyRat> 0.25 * 0.2
1/20
pyRat> 3000 / 6000
1/2
pyRat> 0.625
5/8
pyRat> bye
hmelot-macbook:code hadmelot$
```

1.2 A réaliser sur machine

Téléchargez l'archive `code_rat.zip` et ouvrez la. Celle-ci contient le script `pyRat.py`, toute une série de fichiers nommés `test_ratx.py` où x varie de 1 à 10; ainsi que quelques fichiers nommés

`exosup_raty.py` qui servent à certains exercices supplémentaires. Il manque cependant un fichier : `rational.py`. C'est à vous de le créer.

- 1 Créez une classe `Rational` (dans le fichier `rational.py`). Chaque fichier `test_ratx.py` importe la classe `Rational` et contient quelques instructions créant et manipulant des objets de type `Rational`. De plus, chacun de ces fichiers contient un commentaire précisant ce qui devrait être affiché dans la console en exécutant ces instructions. Votre objectif est de faire en sorte que les 10 étapes proposées fonctionnent correctement, en construisant petit à petit la classe `Rational`. Les étapes doivent être réalisées dans l'ordre.

Par exemple, la première étape nécessite que vous définissiez un constructeur basique (`__init__`) ainsi qu'une première version simple de la méthode `__str__`. En effet, voici le contenu du fichier `test_rat1.py` :

```
""" Devrait afficher dans la console:
1/2
2/4
2/1
3/0
"""

from rational import Rational

print(Rational(1, 2))
print(Rational(2, 4))
print(Rational(2, 1))
print(Rational(3, 0))
```

- 2 Si les différentes étapes de l'exercice 1 ont toutes été réalisées correctement, vous devriez maintenant pouvoir utiliser `pyRat`. testez votre implémentation des nombres rationnels à l'aide de `pyRat`.

2 Exercices complémentaires

- ★☆☆ 3 Améliorez `Rational` pour pouvoir interpréter `exosup_rat1.py` (opérateurs `-` et `/`).
- ★☆☆ 4 Améliorez `Rational` pour pouvoir interpréter `exosup_rat2.py` (opérateur `[]`).
- ★★☆ 5 Améliorez `Rational` pour pouvoir interpréter `exosup_rat3.py` (opérateurs de comparaison).
- ★★☆ 6 Améliorez `pyRat` pour qu'il gère également les opérations ajoutées via les exercices supplémentaires ci-dessus. **Note** : il faut veiller à gérer correctement le symbole `/` (utilisé pour définir une fraction mais également comme opérateur binaire).
- ★★☆ 7 Améliorez `pyRat` pour permettre de définir des variables. **Aide** : utilisez une variable d'instance de type dictionnaire pour stocker les variables ; imposez un format simple à détecter pour les noms des variables (par ex., uniquement des lettres).
- ★★★ 8 Améliorez `pyRat` pour permettre d'utiliser des parenthèses et de définir des expressions chaînées faisant intervenir plus de deux fractions (par ex. $1/2 + (0.2 + 2/3) * 3 / 4$). **Aide** : essayez de définir récursivement une expression chaînée de telle sorte que la précedence des opérateurs soit respectée.
- ★★☆ 9 Créez un nouveau type de données numériques `Complex` pour créer et manipuler des nombres complexes. Surchargez les opérateurs adéquats et ajoutez des méthodes spécifiques propres aux nombres complexes (le calcul du conjugué, etc.)
- Exam 10 (Juin 2011) Problème 5 : un petit réseau social.