

# TD – LA PROGRAMMATION ORIENTEE OBJET

---

## Introduction et objectifs

Ce TD consacré à la programmation orientée objet doit permettre de vous familiariser avec les concepts fondamentaux : notion de classe, d'attribut, de méthode (éventuellement spéciale), de méthode surchargée et d'héritage.

## La classe `cplx`

### Exercice N°1

Compléter la classe `cplx` vue en cours en ajoutant des méthodes permettant d'obtenir :

- Le conjugué.
- L'inverse.
- L'argument compris entre  $-\pi$  et  $\pi$ .

### Exercice N°2 – Surcharge d'opérateurs

Compléter la classe `cplx` en redéfinissant les méthodes spéciales `__add__`, `__radd__`, `__sub__`, `__rsub__`, `__mul__`, `__rmul__`, `__truediv__` et `__rtruediv__` pour pouvoir additionner, soustraire, multiplier et diviser deux complexes/entiers `z1` et `z2` « naturellement », c'est-à-dire en écrivant : `z1 + z2`, `z1 - z2`, `z1 * z2` et `z1 / z2`.

## La classe `ratio`

### Exercice N°4

Dans cet exercice, on vous demande de définir une classe `ratio` permettant de manipuler des éléments de  $\mathbb{Q}$  (fraction de deux entiers).

Les attributs seront :

- `self.num` : le numérateur de la fraction.
- `self.denum` : le dénominateur de la fraction.

Ainsi, pour créer le rationnel  $-\frac{5}{3}$ , on utilisera :

`r = ratio(-5,3)` ou `r = ratio(5,-3)`

On écrira les méthodes :

- `__init__` : cette méthode devra procéder à d'éventuelles simplifications... de sorte que l'attribut `self.num` soit un élément de  $\mathbb{Z}$ , que `self.denum` soit un élément de  $\mathbb{N}^*$  et que la fraction soit irréductible (on écrira et on utilisera une fonction PGCD).
- `__repr__`
- `inversion` : renvoie l'inverse du rationnel considéré.

### Exercice N°5 – Surcharge d'opérateurs

Compléter la classe `ratio` en redéfinissant les méthodes spéciales `__add__`, `__radd__`, `__sub__`, `__rsub__`, `__mul__`, `__rmul__`, `__truediv__` et `__rtruediv__` pour pouvoir additionner, soustraire, multiplier et diviser deux rationnels/entiers `r1` et `r2` « naturellement », c'est-à-dire en écrivant : `r1 + r2`, `r1 - r2`, `r1 * r2` et `r1/r2`.

## La classe `poly`

### Exercice N°6

Construire la classe `poly` permettant de créer et manipuler des polynômes à coefficients réels.

Les attributs seront :

- `self.c` : liste des coefficients non nuls.
- `self.d` : liste des degrés associés (dans l'ordre décroissant).

Ainsi, pour créer le polynôme :  $-5x^{13} + 24x^8 - 567$ , on utilisera :

`P1 = poly([-5, 24, -567], [13, 8, 0])`

On écrira les méthodes :

- `__repr__`
- `eval` : renvoie la valeur prise par le polynôme pour un réel `x` passé en argument.

## Exercice N°7 – Surcharge d'opérateurs

Compléter la classe `poly` en redéfinissant les méthodes spéciales `__add__`, `__radd__`, `__sub__`, `__rsub__`, `__mul__` et `__rmul__` pour pouvoir additionner, soustraire et multiplier deux polynômes/entiers/flottants `P1` et `P2` « naturellement », c'est-à-dire en écrivant : `P1 + P2`, `P1 - P2` et `P1 * P2`.

## Exercice N°8

Ecrire les méthodes complémentaires suivantes :

- `deriv` : renvoie le polynôme dérivé.
- `integrale` : renvoie l'intégrale de la fonction polynôme entre deux réels `a` et `b` passés en arguments.

## Exercice N°9 – La classe `Poly2d`

Pour définir d'autres méthodes spécifiques aux polynômes du second degré, on crée la classe `Poly2d` héritant de la classe `Poly`.

On surchargera la méthode `__init__` de façon à simplifier la création du polynôme.

Pour créer le polynôme  $5x^2 - x + 17$ , on utilisera par exemple :

```
P1 = Poly2d(5, -1, 17)
```

Dans cette même méthode, on aura bien sûr : `self.d = [2, 1, 0]`.

On pourra aussi (mais ce n'est pas obligatoire) surcharger la méthode `__repr__`.

On écrira les méthodes :

- `delta` : renvoie la valeur du discriminant associé.
- `sommet` : renvoie un tuple contenant l'abscisse et l'ordonnée du sommet de la parabole associée.
- `racine` : renvoie, selon le cas le booléen `False` (pas de racine réelle), un flottant (racine double) ou deux flottants (2 racines distinctes).
- `ertan` : renvoie les coefficients de l'équation réduite de la tangente à la parabole associée au point d'abscisse `x` (ce réel étant passé en argument).

## La classe `FR`

### Exercice N°10

Construire la classe `FR` permettant de créer et manipuler des fractions rationnelles à coefficients réels. La classe `poly` pourra (éventuellement) être revue pour hériter de la classe `FR`.