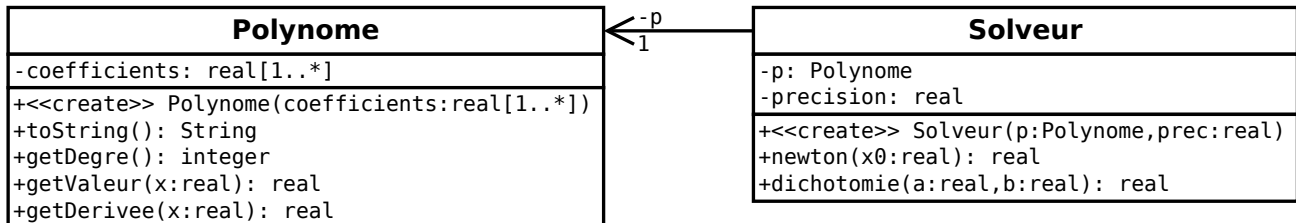


Ce TP est constitué d'exercices dont le but est :

- d'apprendre à créer des classes
- d'apprendre à instancier des objets et à les utiliser
- de créer des classes dont les attributs sont des objets (composition)

ÉNONCÉ

On souhaite réaliser un programme permettant de calculer les racines d'une fonction f soit par la méthode de Newton soit par dichotomie. Dans ce TP, on s'intéressera aux classes `Polynome` et `Solveur` vues en TD. Le diagramme UML de l'application est donné ci-dessous.



EXERCICE 1: POLYNOME

Le but de cette partie est d'implémenter la classe `Polynome` qui sera utilisée par le solveur.

1. En vous aidant du travail réalisé en TP2, créer et implémenter la classe `Polynome`.

On rappelle qu'un polynôme $p(x) = \sum_{i=0}^n a_i x^i$ sera représenté par un tableau contenant les différents coefficients a_i (par ordre croissant de puissance).

2. Écrire un programme principal qui crée le polynôme $p(x) = 0.0714x^4 + 0.0714x^3 - 0.9286x^2 - 0.0714x + 0.8571$, l'affiche à l'écran et calcule sa valeur en différents points afin de vérifier le bon fonctionnement de la classe `Polynome`.
Corriger si nécessaire les erreurs détectées.
3. Implémenter la méthode `__eq__` de la classe `Polynome` afin de permettre le test d'égalité de 2 `Polynome`.
4. **(Facultatif)** Implémenter une méthode `__add__` qui redéfinit l'opérateur `+` pour la classe `Polynome`

EXERCICE 2: RÉOLUTION D'ÉQUATIONS

Les polynômes ayant été déclarés, on s'intéressera maintenant à la résolution de l'équation $f(x) = 0$. On repar-tira du code réalisé durant le TP2 pour créer la classe `Solveur`.

1. En vous aidant du travail réalisé en TP2, créer et implémenter la classe `Solveur`.
2. Compléter le programme principal précédent pour créer un solveur et rechercher le zéro de la fonction $f(x)$ avec une précision $\varepsilon = 10^{-5}$ par la méthode de Newton ($x_0 = 2$) et dichotomique (intervalle $[0; 5]$).
Corriger si nécessaire les erreurs détectées.