

# TD: classes génériques

## Exercice 1. Polynômes univariés

---

Le but de l'exercice est de créer une classe de polynômes univariés (c'est à dire avec une seule variable) aux coefficients génériques. Autrement dit, on veut que la classe permette de créer à la fois des polynômes aux coefficients entiers, rationnels, ou autre. Le type des coefficients devra respecter l'interface `ICoeff` suivante.

```
interface ICoeff {  
    public void add(ICoeff c);  
    public void mul(ICoeff c);  
    public void pow(int e);  
    public boolean isZero();  
}
```

- a. Écrivez une classe `Entier` qui implémente `ICoeff` avec des coefficients entiers.
- b. Écrivez une classe `Rationnel` qui implémente `ICoeff` avec des coefficients rationnels.
- c. Écrivez une classe générique `Monome`, qui prend en paramètre une classe `Coeff` implémentant l'interface `ICoeff`. `Monome` implémente un monôme univarié, c'est à dire un coefficient (de type `Coeff`) et un degré (l'exposant de la variable). Par exemple,  $\frac{1}{2}x^2$  est un monôme de degré 2 et de coefficient rationnel  $\frac{1}{2}$ , et 3 est un monôme de degré 0 et de coefficient entier 3.  
Cette classe doit posséder les méthodes suivantes :
  - un constructeur qui prend en paramètre un degré et un coefficient
  - une méthode `add` qui prend en paramètre un monôme et l'ajoute au monôme courant
  - une méthode `mul` qui prend en paramètre un monôme et le multiplie au monôme courant
- d. Écrivez une classe générique `Polynome`, qui prend en paramètre une classe `Coeff` implémentant l'interface `ICoeff`. `Polynome` implémente un polynôme univarié en représentation creuse et développée, c'est à dire une liste de monômes univariés.  
Cette classe doit posséder les méthodes suivantes :
  - un constructeur à partir d'un monôme
  - une méthode `add` qui prend en paramètre un polynôme et l'ajoute au polynôme courant
  - une méthode `mul` qui prend en paramètre un polynôme et le multiplie au polynôme courant
- e. Ajoutez une méthode `eval` dans la classe `Monome`, qui prend en paramètre un coefficient et renvoie le monôme évalué en ce coefficient. **Note :** Ni le monôme, ni le coefficient donné en paramètre ne doivent être altérés par la méthode. Quelle méthode devez-vous ajouter à l'interface `ICoeff` pour assurer cette propriété ?
- f. Ajoutez une méthode `eval` dans la classe `Polynome`, qui prend en paramètre un coefficient et renvoie le polynôme évalué en ce coefficient. **Note :** Ni le monôme, ni le coefficient donné en paramètre ne doivent être altérés par la méthode. Quelle méthode devez-vous ajouter à l'interface `ICoeff` pour assurer cette propriété ?

## Exercice 2. Pour aller plus loin

---

- a. Reprenez l'exercice précédent en rendant générique le type des variables.
- b. Reprenez l'exercice précédent avec des polynômes multivariés. La méthode d'évaluation de tels polynômes devra prendre en paramètre une map (c'est à dire du type `Map` fourni par Java) associant une valeur à chaque variable.