

Nous rappelons que les méthodes sont maintenant à écrire dans la classe correspondante.

Sauf mention contraire, il est interdit d'utiliser des méthodes des exercices précédents pour résoudre l'exercice courant!

Les boucles sont interdits, mais certains exercices se font tout de même sans récursivité !

Exercice 1. Polynômes (faisable en TP) Dans cet exercice, on écrira les méthodes demandées sans boucles, et sans méthodes auxiliaires.

On représente des polynômes avec la classe suivante.

```
class Polynome{
    private int coeff;
    private int deg; //deg >= 0
    private Polynome suiv;
    //represente le monosme coeff*x^{deg}
    //suiv==null represente le polynome nul

    public Polynome(){
        suiv = null;
    }

    public boolean estVide(){
        return suiv==null;
    }

    public Polynome(Polynome p){
        //constructeur par copie en profondeur
        if(p.estVide()){
            suiv=null;
        }
        else{
            coeff = p.coeff;
            deg = p.deg;
            suiv = new Polynome(p.suiv);
        }
    }
}
```

Par exemple, le polynôme : $1 + 3x^2 + x + x^2$ est représenté par $(1, 0) :: (3, 2) :: (1, 1) :: (1, 2) :: (*, null)$.

Question 1.1.

Munir la classe Polynome d'une méthode `String toString()`.

Question 1.2.

Munir la classe Polynome d'une méthode `int eval(int x)` qui évalue le polynôme en x .

Question 1.3.

Munir la classe Polynome d'une méthode `int degre()` qui calcule le degré du polynôme (attention aux coefficients nuls ...). La méthode a pour prérequis que le polynome est réduit (réfléchir à ce qui pourrait se passer si ça n'était pas le cas).

Question 1.4.

Munir la classe `Polynome` d'une méthode `Polynome simplifie()` qui calcule un nouveau polynôme indépendant en enlevant tous les monômes ayant un coefficient de 0.

Question 1.5.

Munir la classe `Polynome` d'une méthode `Polynome derivier()` qui calcule un nouveau polynôme indépendant correspondant au polynome dérivé.

Question 1.6.

Munir la classe `Polynome` d'une méthode `Polynome reduire()` qui calcule un nouveau polynôme indépendant en fusionnant les monômes de même degrés. Par exemple, le polynôme $1 + 3x^2 + x + x^2$ doit devenir $1 + 4x^2 + x$ (l'ordre des monômes est arbitraire)

Question 1.7.

Munir la classe `Polynome` d'une méthode `Polynome ajouter(Polynome p)` qui calcule un nouveau polynôme indépendant de `this` et de `p` en ajoutant les deux polynômes.

Question 1.8.

Munir la classe `Polynome` d'une méthode `Polynome multiplier(Polynome p)` qui calcule un nouveau polynôme indépendant de `this` et de `p` en multipliant les deux polynômes (vous pouvez utiliser la méthode `ajouter`).