

Thème : Arbres binaires et Tas

Exercice III.1 Expressions : Parcours d'arbre

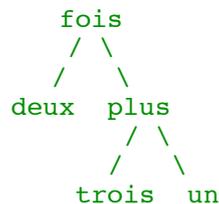
On souhaite représenter une expression arithmétique par un arbre binaire d'opérateurs.
Un opérateur est défini par les valeurs plus, moins, fois, sur, zero, un, deux, trois.

Soit la classe `ArbreBinaireOp` implémentant les arbres binaires d'opérateurs

```
public class ArbreBinaireOp {
    enum Op{ plus, moins, fois, sur, zero, un, deux, trois} ;
    Op racine ;
    ArbreBinaireOp filsgauche ;
    ArbreBinaireOp filsdroit ;

    ArbreBinaireOp (Op r, ArbreBinaireOp gauche, ArbreBinaireOp
        droit){
        this.racine=r ;
        this.filsgauche=gauche ; this.filsdroit=droit ;
    }
    Op contenu() // retourne la racine
    {retourner this.racine }
}
```

On va utiliser des arbres binaires d'opérateurs, par exemple, l'expression $2*(3+1)$ sera représentée par l'arbre :



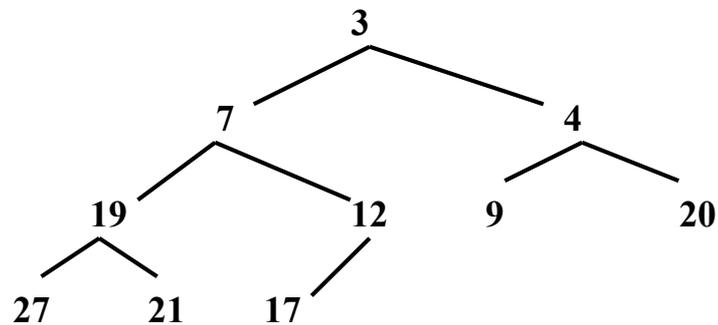
Question 1 Déclarer une instance de la classe `ArbreBinaireOp`

Question 2 Ecrire une méthode qui teste si un arbre binaire est une feuille (c'est-à-dire un arbre non vide mais dont les deux sous-arbres sont vides).

Question 3 Ecrire une méthode qui affiche les arbres binaires d'opérateur sous forme agréable (" $2*(3+1)$ ").

Question 4 Ecrire une méthode qui calcule la valeur associée à l'arbre binaire (résultat de type entier).

Exercice III.2 Soient des nombres rangés dans la structure suivante:



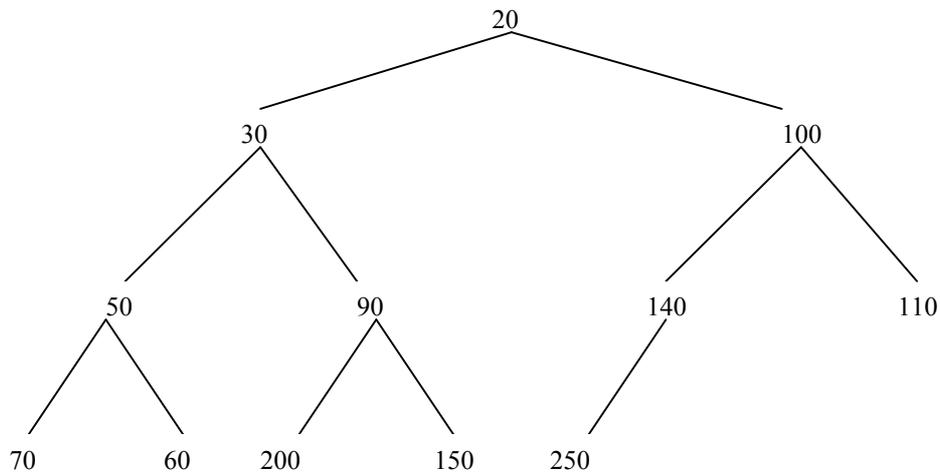
Question 1 Comment appelle-t-on une telle structure ?

Question 2 On y insère "5" selon la procédure vue en cours; donner l'arbre obtenu.

Question 3 Même question si ensuite on supprime 3 (dans l'arbre obtenu en 2).

Exercice III.3

Soit le tas suivant (on suppose que les nombres correspondent à des clés)



Question 1 On utilise un tableau T_ARB pour représenter cet arbre. Donner ce tableau.

Question 2 On utilisera la classe suivante ;

```

Public class ArbreBinaire{
    C_clé racine ;
    ArbreBinaire gauche ;
    ArbreBinaire droit ;
    ArbreBinaire(C_clé r, ArbreBinaire g, ArbreBinaire d){
        racine = r ;
        gauche = g ;
        droit = d ;

        ArbreBinaire contenu(){
            retourner this.racine ;
        }
    }
  
```

Soit la méthode récursive suivante applicable à un arbre binaire non vide(ici un tas).

```
void lecture () {  
    si gauche != null) alors  
    gauche.lecture() ;  
    finsi ;  
    ecrire (this.contenu() );// écrit la valeur de la racine  
    si droit != null alors  
    droit.lecture();  
    finsi;  
}
```

Donner la liste des valeurs obtenues en exécutant cette procédure sur l'arbre initial ci-dessus.

Question 3

On suppose ici que a , b , c , d , e et f désignent des sous-arbres de l'arbre initial et sont tels que :

```
a.contenu=30;      b.contenu=50;      c.contenu=90;  
d.contenu=70;      e.contenu=60;      f.contenu=200;
```

Compléter :

```
b.gauche=?;      a.droit.contenu=?;  
f.droit=?;      a.gauche.droit.contenu=?
```