

## UE ALGO5 — TD2 — Séance 9 : Implémentation d'un type abstrait « Arbre binaire »

On donne un exemple de spécification d'un type abstrait *Arbre* construit sur un type *Élément*.

*ArbreVide*

```
{ paramètres : aucun  
  valeur de retour : un Arbre  
  description : renvoie un Arbre vide  
  effets de bord : aucun }
```

*NouveauNœud*

```
{ paramètres : un Arbre G, un Élément x, un Arbre D (données)  
  valeur de retour : un Arbre  
  description : renvoie un Arbre constitué du nœud (G, x, D)  
  effets de bord : un nouveau nœud a été créé }
```

*InsérerGauche*

```
{ paramètres : un Arbre A (donnée-résultat), un Arbre G (donnée)  
  valeur de retour : aucune  
  description : A doit être non vide, supprime le fils gauche de A et le remplace par l'Arbre G.  
  effets de bord : l'Arbre A est modifié, son ancien fils gauche est «détruit» }
```

*InsérerDroit*

```
{ paramètres : un Arbre A (donnée-résultat), un Arbre D (donnée)  
  valeur de retour : aucune  
  description : A doit être non vide, supprime le fils droit de A et le remplace par l'Arbre D.  
  effets de bord : l'Arbre A est modifié, son ancien fils droit est «détruit» }
```

*EstVide*

```
{ paramètres : un Arbre A (donnée)  
  valeur de retour : un booléen  
  description : vaut vrai ssi A est un Arbre vide  
  effets de bord : aucun }
```

*Elem*

```
{ paramètres : un Arbre A (donnée)  
  valeur de retour : un Élément  
  description : A doit être non vide, renvoie l'élément associé à la racine de A  
  effets de bord : aucun }
```

*FDroit*

```
{ paramètres : un Arbre A (donnée)  
  valeur de retour : un Arbre  
  description : A doit être non vide, renvoie le fils droit associé à la racine de A  
  effets de bord : aucun }
```

*FGauche*

```
{ paramètres : un Arbre A (donnée)  
  valeur de retour : un Arbre  
  description : A doit être non vide, renvoie le fils gauche associé à la racine de A  
  effets de bord : aucun }
```

*Libérer*

```
{ paramètres : un Arbre A (donnée)  
  valeur de retour : aucune  
  description : A doit être non vide, libère la mémoire associée au nœud racine de A  
  effets de bord : La racine de A est détruite, les fils gauche et droit sont inchangés }
```

## Utilisation du type abstrait **Arbre**

1. Dessiner un arbre binaire de hauteur 3 comportant 5 feuilles

En n'utilisant que des primitives du type abstrait :

2. Écrivez une séquence d'instructions permettant de construire l'arbre binaire proposé à la question 1.

3. Écrivez une fonction qui prend en paramètre un arbre binaire et renvoie le nombre de feuilles de cet arbre.

4. Écrivez une fonction qui prend en paramètre un arbre binaire et renvoie la hauteur de cet arbre.

5. Écrivez une procédure permettant de «supprimer» un arbre entier, en libérant la mémoire de chacun de ses nœuds.

## Implémentation du type abstrait **Arbre**

6. Proposez une implémentation du type **Arbre** à l'aide d'un tableau de taille  $N$  alloué statiquement (on suppose donc ici que le nombre de nœuds de tout arbre binaire sera limité à  $N$ ). Écrivez le code de chaque primitive pour cette implémentation.

7. Proposez maintenant une implémentation du type **Arbre** à l'aide de cellules mémoires allouées dynamiquement et chaînées par pointeurs.