

Corrigé du TD 6 : arbre binaire et récursivité

```
class Noeud
{
    String valeur;
    Noeud gauche;
    Noeud droite;
    Noeud(String _valeur)
    {
        valeur=_valeur;
    }
}

class Arbre
{
    private Noeud tete;
    void insererValeur(String valeur)
    {
        if(tete==null){tete=new Noeud(valeur);return;}
        Noeud lieu=trouver(valeur);
        if(lieu.valeur==valeur){return;}
        Noeud nouveau=new Noeud(valeur);
        if(valeur.compareTo(lieu.valeur)<0){lieu.gauche=nouveau;}
        if(valeur.compareTo(lieu.valeur)>0){lieu.droite=nouveau;}
    }

    Noeud trouver(String valeur)
    {
        if(tete==null){return null;}// arbre vide
        return trouverRecurusif(valeur,tete);
    }
    private Noeud trouverRecurusif(String valeur,Noeud courant)
    {
        if(valeur.compareTo(courant.valeur)<0 && courant.gauche!=null)
        {
            return trouverRecurusif(valeur,courant.gauche);
        }
        if(valeur.compareTo(courant.valeur)>0 && courant.droite!=null)
        {
            return trouverRecurusif(valeur,courant.droite);
        }
        return courant;
    }
    void afficher()
    {
        if(tete==null){return;}
        afficherRecursif(tete);
    }
}
```

```
private void afficherRecursif(Noeud courant)
{
    if(courant.gauche!=null){afficherRecursif(courant.gauche);}
    System.out.println(courant.valeur);
    if(courant.droite!=null){afficherRecursif(courant.droite);}
}
}
```