



www.dirasats.com

هذا الغلاف لا يعبر عن حقوق الملكية او فحوى الكتاب, فهو مجرد واجهة للموقع المحمل منه



شكرا لك على ثقتك بنا وعلى اختيار موقعنا

www.dirasats.com



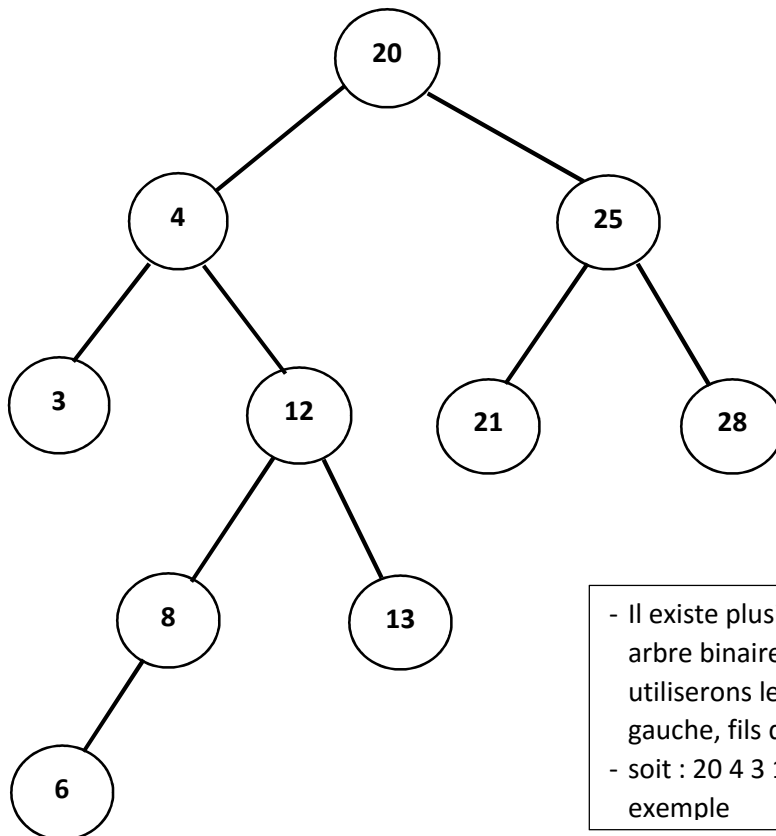
من اجل تواصل معنا المرجو زيارة الموقع ستجد جميع المعلومات

www.dirasats.com

Examen Java SMI S6
Prof. Abdessamad Belangour

Problème :

Un arbre binaire est un arbre qui a au plus deux fils : un fils droit et un fils gauche.



Dans l'exemple à gauche :

- Le nœud 20 est appelé racine de l'arbre
- Les nœuds 3, 6, 13, 21, 28 sont appelés feuilles.
- La taille de l'arbre correspond au nombre de ces nœuds soit 10
- La hauteur de l'arbre correspond à sa profondeur soit 5
- Si un nœud dispose d'un seul fils alors celui-là est gauche.

- Il existe plusieurs manières de parcourir un arbre binaire. Parmi ces manières nous utiliserons le Parcours en Préfixe : (père, fils gauche, fils droit)
- soit : 20 4 3 12 8 6 13 25 21 28 dans notre exemple

Nous décidons de représenter cet arbre par une classe Arbre comme suit :

```
public class Arbre {  
    private int contenu;  
    private Arbre filsGauche;  
    private Arbre filsDroit;  
  
    public Arbre(int contenu) { // nœud sans fils  
        this.contenu = contenu;  
        this.filsGauche = null;  
        this.filsDroit = null;  
    }  
  
    public Arbre(int contenu, Arbre filsGauche) { //nœud avec un seul fils  
        this.contenu = contenu;  
        this.filsGauche = filsGauche;  
        this.filsDroit = null;  
    }  
}
```

<pre> public Arbre(int contenu, Arbre filsGauche, Arbre filsDroit) { //avec 2 fils this.contenu = contenu; this.filsGauche = filsGauche; this.filsDroit = filsDroit; } </pre>
<pre> // on suppose que les getters et setters sont fournis </pre>
<pre> // fonction calculant la taille de l'arbre public int calculerTaille(){.....} </pre>
<pre> // fonction calculant la hauteur de l'arbre public int calculerHauteur(){.....} </pre>
<pre> // parcours en préfixe par la fonction toString @Override public String toString() {.....} </pre>
<pre> // test d'égalité @Override public boolean equals(Object o) {.....} </pre>

Questions :

N.B. : les réponses doivent être fournies sans réécrire toute la classe !

Partie 1 :

- 1) Donner le code de la fonction calculerTaille()
- 2) Donner le code de la fonction calculerHauteur()
- 3) Donner le code de la fonction toString() qui transforme un arbre en une chaîne de caractère selon le parcours en préfixe.
- 4) Donner le code de la fonction equals() qui permet l'égalité de deux objets arbres.

Partie 2 :

Nous décidons d'ajouter deux attributs à la classe Arbre qui sont taille et hauteur

- 5) Donner la déclaration de ces deux nouveaux attributs
- 6) Donner les fonctions Getters et/ou Setters pour l'attribut taille
- 7) Modifier le troisième constructeur de la classe Arbre pour prendre en considération l'initialisation de ces deux nouveaux attributs.