

Thème : Révisions

Exercice

On souhaite construire une classe de listes d'entiers triés dans l'ordre croissant. Coder les méthodes suivantes sous une forme récursive.

1. Ecrire une procédure `affiche` qui affiche la liste à l'écran
2. Ecrire une fonction `trouver` qui teste si un entier `x` appartient à la liste.
3. Ecrire une fonction `insereEnPlace` qui insère un entier `x` dans la liste, tout en la maintenant triée et renvoie la liste obtenue
4. Ecrire une fonction `supprimer` qui supprime la première occurrence trouvée d'un entier `x` dans la liste.

Corrigé

On souhaite construire une classe de listes d'entiers triés dans l'ordre croissant.

```
class ListeTrie {
    int valeur ;
    ListeTrie suivant ;
    public void affiche(int premier, ListeTrie reste){
        valeur=premier ;
        suivant=reste ;
    }
}
```

1. Ecrire une procédure `affiche` qui affiche la liste à l'écran

```
public void affiche () {
    ListeTrie p=this;
    System.out.println(p.valeur);
    if ( p.suivant != null) p.suivant.affiche();
}
```

2. Ecrire une fonction `trouver` qui teste si un entier `x` appartient a la liste.

```
public boolean trouver (int x){
    ListeTrie p=this;
    if (p.valeur == x)
        return(true);
    if (p.valeur >x)
        return false; //l'élément n'a pas été retrouvé
    if (p.suivant ==null) return(false);
    return(p.suivant.trouver(x));
}
```

3. Ecrire une fonction `insereEnPlace` qui insère un entier `x` dans la liste, tout en la maintenant triée et renvoie la liste obtenue

```
public ListeTrie insereEnPlace(int x){
    //insertion recursive, ordre croissant
    if (this.valeur >=x){
        //cas ou on insere en tete
        ListeTrie L =new ListeTrie(x, this);
        return L;
    }
    ListeTrie L=this;
    ListeTrie Lbis;
    if (L.suivant==null){
        Lbis =new ListeTrie(x, null);
    }else{
        Lbis=(L.suivant).insereEnPlace(x);
    }
    L.suivant=Lbis;
    return L;
}
```

4. Ecrire une fonction `supprimer` qui supprime la première occurrence trouvée d'un entier `x` dans la liste.

```
public ListeTrie supprimer (int x){
    ListeTrie p=this;
    if (p.valeur == x) return (p.suivant);
    if (p.valeur >x)
```

```

        return p;    //l'élément n'a pas été retrouvé
    if (p.suivant ==null) return p;
    //dans tous les autres cas
    ListeTrie q = p.suivant.supprimer(x);
    p.suivant=q;
    return p;
}

```

Corrigé une version plus synthétique

On souhaite construire une classe de listes d'entiers triés dans l'ordre croissant.

```

Class ListeTrie {
    Int valeur ;
    ListeTrie suivant ;
    public ListeTrie(int premier ; suivant reste) {
        this.valeur=premier ;
        this.suivant=reste ;
    }
    // les méthodes
}

```

5. Ecrire une procédure affiche qui affiche la liste à l'écran

```

public void affiche () {
    System.out.println(this.valeur);
    if ( this.suivant != null) this.suivant.affiche();
}

```

6. Ecrire une fonction trouver qui teste si un entier x appartient a la liste.

```

public boolean trouver (int x){
    if (this.valeur == x)
        return(true);
    if (this.valeur >x)
        return false;    //l'élément n'a pas été trouvé
    if (this.suivant ==null) return(false);
    return(this.suivant.trouver(x));
}

```

7. Ecrire une fonction insereEnPlace qui insère un entier x dans la liste, tout en la maintenant triée et renvoie la liste obtenue

```

public ListeTrie insereEnPlace(int x){
    //insertion recursive, ordre croissant
    if (this.valeur >=x){
        //cas ou on insere en tete
        return (new ListeTrie(x, this));
    }
    if (this.suivant==null){
        return (new ListeTrie(x, null));
    }
    return (new ListeTrie(this.valeur,
    this.suivant.insereEnPlace(x))) ;
}

```

8. Ecrire une fonction supprimer qui supprime la première occurrence trouvée

d'un entier x dans la liste.

```
public ListeTrie supprmer (int x){
    if (this.valeur == x) return (this.suivant);
    if (this.valeur >x)
        return this;    //l'élément n'a pas été retrouvé
    if (this.suivant ==null) return this;
    //dans tous les autres cas
    return(new ListeTrie(this.valeur,this.suivant.supprmer(x)) ) ;
}
```