

Feuille d'exercices 22

listes

Exercice 22.1 dessins

On reprend ici la classe `Liste`. Pour chacun des points marqués par un commentaire dans la méthode `main`, dessinez la liste 1.

```
public class ElementListe {
    private int valeur;
    private ElementListe suivant;

    public ElementListe(int valeur, ElementListe suivant) {
        this.valeur = valeur; this.suivant = suivant;
    }

    public ElementListe(int v) {
        this.valeur = v; this.suivant = null;
    }

    public int getValeur() {
        return valeur;
    }

    public void setValeur(int valeur) {
        this.valeur = valeur;
    }

    public ElementListe getSuivant() {
        return suivant;
    }

    public void setSuivant(ElementListe suivant) {
        this.suivant = suivant;
    }
}

public class ListeIter {
    ElementListe premier;

    public boolean estVide() {
        return premier == null;
    }

    public ElementListe getPremier() {
        return premier;
    }

    public void ajouterAuDebut(int v) {
        ElementListe ancienPremier = premier;
        premier = new ElementListe(v, ancienPremier);
    }

    public void ajouterALaFin(int v) {
        if (estVide()) {
            premier = new ElementListe(v);
        } else {
            ElementListe dernier = getDernierElement();
            dernier.setSuivant(new ElementListe(v));
        }
    }

    private ElementListe getDernierElement() {
        ElementListe dernier = premier;
        while (dernier.getSuivant() != null) {
            dernier = dernier.getSuivant();
        }
        return dernier;
    }
}
```

```

public int getLongueur() {
    int longueur= 0;
    ElementListe elt= getPremier();
    while (elt != null) {
        longueur++;
        elt= elt.getSuivant();}
    return longueur;}

public boolean contient(int v) {
    boolean trouve= false;
    ElementListe elt= getPremier();
    while (! trouve && elt != null) {
        if (elt.getValeur() ==v ) {
            trouve= true;
        } else {
            elt= elt.getSuivant();}
    }
    return trouve;}

public void retirerPremiereOccurrence(int v) {
    if (estVide())
        return;
    if (premier.getValeur() == v) {
        premier= premier.getSuivant();
    } else {
        ElementListe precedent= premier;
        ElementListe elt= premier.getSuivant();
        while (elt != null && elt.getValeur() != v) {
            precedent= elt;
            elt= elt.getSuivant();}
        if (elt != null) {
            precedent.setSuivant(elt.getSuivant());}
    }
}

public void concatener(ListeIter l) {
    if (this.estVide()) {
        this.premier= l.premier;
    } else {
        ElementListe dernier= getDernierElement();
        dernier.setSuivant(l.getPremier());}
}

}

class Exo21_1{

    public static void main(String[] args){
        ListeIter l = new ListeIter();
        // dessin 1
        l.ajouterAuDebut(1);
        // dessin 2
        l.ajouterAuDebut(2);
        // dessin 3
        l.ajouterAuDebut(3);
        // dessin 4
        l.getPremier().setValeur(4);
        // dessin 5
        l.getPremier().setSuivant(new ElementListe(5,null));
        // dessin 6
        (l.getPremier().getSuivant()).setSuivant(new ElementListe(6));
        // dessin 7

    }
}

```

Exercice 22.2 - affichage

Question 22.2.1

Reprendre la classe `Liste` du cours et lui ajouter une méthode `ecrireListe()` qui affiche la liste.

Question 22.2.2

Réécrire la méthode `ecrireListe()` en utilisant la récursivité (**sans boucle**).

Question 22.2.3

Si ce n'est pas déjà fait, faire afficher une virgule entre chaque valeur de la liste.
Attention : il ne doit pas y avoir de virgule avant la première valeur ni après la dernière.

Exercice 22.3 - insertion entre chaque élément

Ajouter une méthode qui permette d'insérer entre chaque élément de la liste une nouvelle valeur.
Par exemple, avec la liste `[5,12,7]`, la méthode `insérerEntre(1)` donnerait :
`[5 , 1 , 12 , 1 , 7]`.