

Exercises 22

lists

Exercise 1.22. Drawing the List

Here we take the class `Liste`. For each of the points scored by a comment in the main method, draw the list.

```
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
    }
}

```

Exercise 22.2 - Display

Question 22.2.1

Take the class `List` and add a method `ecrireListe()` which displays the list.

Question 22.2.2

Rewrite the method `EcrireListe()` using recursion (no loops).

Question 22.2.3

If it is not already done, display a comma between each value in the list.

Warning: There should be no comma before the first or after the last value.

Exercise 22.3 - Insertion between each element

Add a method that inserts between each element of the list a new value. For example, with the `List[5,12,7]`, the method `insererEntr(1)` would give: `[5,1,12,1,7]`.