

Université Sultan Moulay Slimane

Beni-Mellal



École nationale des sciences appliquées

Khouribga



COMPTE RENDU :

TP5 JAVA POO

Filière : Informatique et Ingénierie des données (iid1)

Réalisé par :

- LAHMAMA Fatima-Zahraa
- EL FATHI Zakaria

Encadré par :

- Mr. GHERABI Noredline

EXERCICE 1 :

1. Définissons une classe **Personne** avec des attributs :

```
// EX1
package tp5;
// Q1
public class Personne {
    int code;
    String nom, prenom;
}
```

2. Définissons les accesseurs (getters and setters) : a l'intérieur de la classe Personne on écrit :

```
//Q2
public int getCode() {
    return code;
}
public void setCode(int code) {
    this.code = code;
}
public String getNom() {
    return nom;
}
public void setNom(String nom) {
    this.nom = nom;
}
public String getPrenom() {
    return prenom;
}
public void setPrenom(String prenom) {
    this.prenom = prenom;
}
```

TRAVAUX PRATIQUES 5

– JAVA POO

3. Un constructeur qui initialise les attributs avec des valeurs saisies par l'utilisateur :

```
//Q3
public Personne(int code,String nom, String prenom) {
    this.code=code;
    this.prenom=prenom;
    this.nom=nom;
}
```

4. Redéfinition de la méthode **toString()** qui affiche les informations d'une **Personne** :

Dans la même classe on écrit :

```
//Q4
@Override
public String toString() {
    return "Personne [code=" + code + ", nom=" + nom + ", prenom=" +
prenom + "] \n";
}
```

5. Créons l'interface IPerson avec l'ensemble des méthodes (save, delete, update, findById, etc..) :

```
package tp5;

import java.util.List;

//Q5
public interface IPerson {

    public void save(Personne P);

    public void delete(int code);

    public void update(Personne P);

    public Personne findById(int code);

    public List <Personne> findAll() ;
}
```

TRAVAUX PRATIQUES 5

– JAVA POO

6. Créons la classe `PersonneService` qui implémente l'interface `IPerson`, telles que les données seront stockées dans une collection de type `ArrayList<>` :

```
package tp5;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class PersonneService implements IPerson {
    public ArrayList<Personne> personne =new ArrayList<Personne>();
    static Scanner sc=new Scanner(System.in);

    public void save(Personne P) {
        personne.add(P);
    }

    @Override
    public void delete(int Code) {
        for(int i=0;i< personne.size();i++) {
            if ((personne.get(i)).code==Code) personne.remove(i);
        }
    }

    @Override
    public void update(Personne P) {

        System.out.println("code: ");
        P.setCode(sc.nextInt());
        System.out.println("Nom:");
        P.setNom(sc.next());
        System.out.println("Prenom: ");
        P.setPrenom(sc.next());
    }

    @Override
```

TRAVAUX PRATIQUES 5
– JAVA POO

```
public Personne findById(int Code) {  
    for(int i=0;i<personne.size();i++) {  
        if ((personne.get(i)).code==Code) return personne.get(i);  
    }  
    return null;  
}  
  
@Override  
public String toString() {  
    return "PersonneService [personne=" + personne + " ]";  
}  
  
@Override  
public List<Personne> findAll() {  
  
    return personne;  
}  
  
}  
7.
```

8. On crée une fonction principale de test qui permet la création, recherche, modification, affichages de cinq **Personnes** :

```
package tp5;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
//import java.util.ArrayList;  
  
public class mainTest {  
    //Q7  
  
    public static void main(String[] args) {  
        Scanner sc=new Scanner(System.in);
```

TRAVAUX PRATIQUES 5

– JAVA POO

```
// on cree cinq Personnes(instanciacion de cinq objets de type Personnes)
Personne p1 =new Personne (1,"El fathi","zakaria");
Personne p2 =new Personne (2,"Lahmama","fz");
Personne p3 =new Personne (3,"saidi","said");
Personne p4 =new Personne (4,"gnoumi","simo");
Personne p5 =new Personne (5,"rachdi","halima");

// ajout des elements dans la liste
PersonneService personneServ = new PersonneService();
personneServ.save(p1);
personneServ.save(p2);
personneServ.save(p3);
personneServ.save(p4);
personneServ.save(p5);

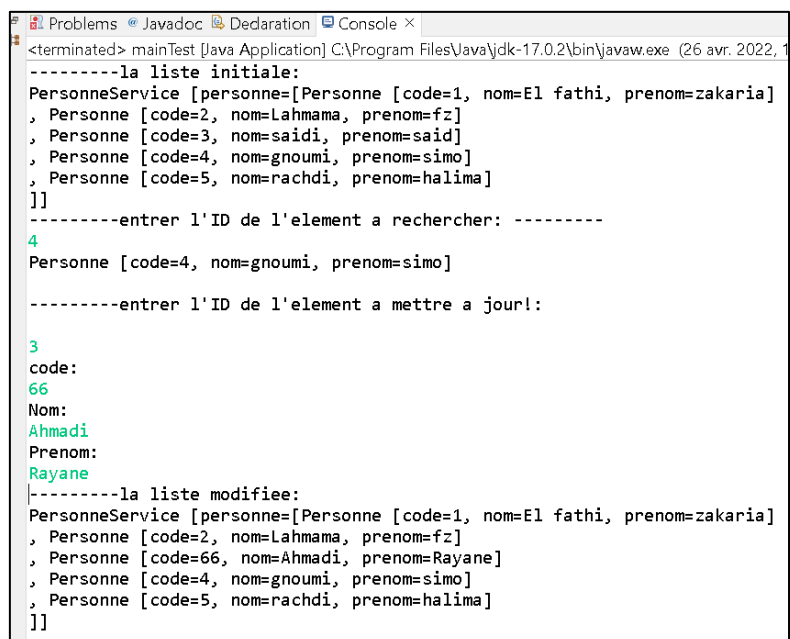
//affichage de la Liste des Personnes
System.out.println("-----la liste initiale: \n"+personneServ);

// recherche d'un element avec son code ID
System.out.println("-----entrer l'ID de l'element a rechercher: -----");
System.out.println(personneServ.findById(sc.nextInt()));

//modification des informations d'une Personne
System.out.println("-----entrer l'ID de l'element a mettre a jour!: \n ");
personneServ.update(personneServ.findById(sc.nextInt()));

// affichage de la Liste finale des Personnes
System.out.println("-----la liste modifiee: \n"+personneServ);}}
```

Apres execution on aura :



```
<terminated> mainTest [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (26 avr. 2022, 1
-----la liste initiale:
PersonneService [personne=[Personne [code=1, nom=El fathi, prenom=zakaria]
, Personne [code=2, nom=Lahmama, prenom=fz]
, Personne [code=3, nom=saidi, prenom=said]
, Personne [code=4, nom=gnoumi, prenom=simo]
, Personne [code=5, nom=rachdi, prenom=halima]
]]
-----entrer l'ID de l'element a rechercher: -----
4
Personne [code=4, nom=gnoumi, prenom=simo]
-----entrer l'ID de l'element a mettre a jour!:
3
code:
66
Nom:
Ahmadi
Prenom:
Rayane
-----la liste modifiee:
PersonneService [personne=[Personne [code=1, nom=El fathi, prenom=zakaria]
, Personne [code=2, nom=Lahmama, prenom=fz]
, Personne [code=66, nom=Ahmadi, prenom=Rayane]
, Personne [code=4, nom=gnoumi, prenom=simo]
, Personne [code=5, nom=rachdi, prenom=halima]
]]
```

EXERCICE 2 :

Cette exercice vise a mieux comprendre les collections précisément le type **LinkedList** et l'usage des **Iterators** :

1. On crée une liste **liste1** de type LinkedList contenant des entiers :

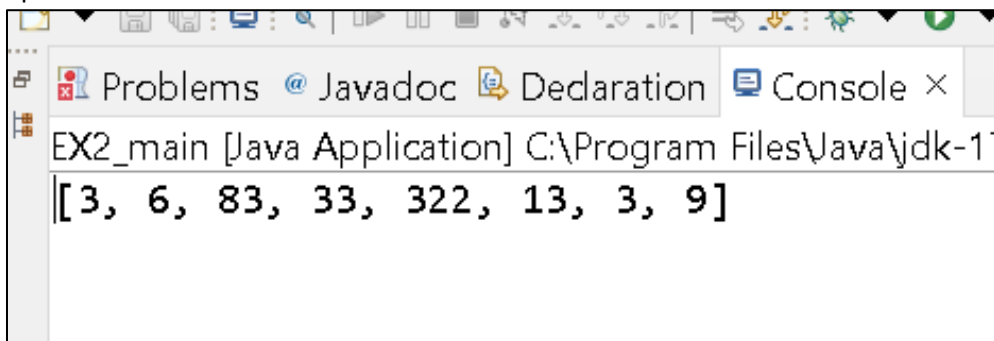
```
package tp5;

import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Scanner;

public class EX2_main {

    public static void main(String[] args) {
        //Q1
        LinkedList liste1 = new LinkedList();// instanciation de liste1 de type LinkedList
        liste1.add(3);
        liste1.add(6);
        liste1.add(83);
        liste1.add(33);
        liste1.add(322);
        liste1.add(13);
        liste1.add(3);
        liste1.add(9);
    }
}
```

Après exécution on aura :



2. Insertion dans la liste un élément a une position bien spécifiée :

```
//Q2
System.out.println("inserer 1 position et la valeur");
Scanner sc=new Scanner(System.in);
int i=sc.nextInt();
```

TRAVAUX PRATIQUES 5

– JAVA POO

```
Integer n=sc.nextInt();
while(i>liste1.size()) {
    System.out.println("position invalide!\n entrer la position a nouveau:");
    i=sc.nextInt();
}
liste1.add(i,n);
System.out.println("la nouvelle liste est :"+liste1);
```

Après exécution on aura :

```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
insérer 1 position et la valeur
4
19
la nouvelle liste est :[3, 6, 83, 33, 19, 322, 13, 3, 9]
```

3. Insertion au début et à la queue de la liste
Dans la même classe principale :

```
//Q3
System.out.println("entrer l'element du debut: ");
liste1.addFirst(sc.nextInt());
System.out.println("entrer l'element a la fin: ");
liste1.addLast(sc.nextInt());
```

Après exécution on aura :

```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
entrer l'element du debut:
111
entrer l'element a la fin:
111
la nouvelle liste est :[111, 3, 6, 83, 33, 322, 13, 3, 9, 111]
```

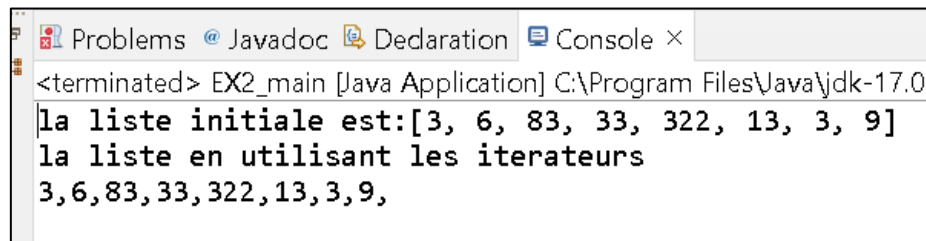

TRAVAUX PRATIQUES 5

– JAVA POO

4. Affichage des éléments de la liste en utilisant l'itérateur de l'interface Iterator :

```
//Q4
System.out.println("la liste en utilisant les iterateurs");
Iterator <Integer> it= liste1.iterator();
while(it.hasNext()) {
    System.out.print(it.next()+",");
}
```

Après exécution on aura :



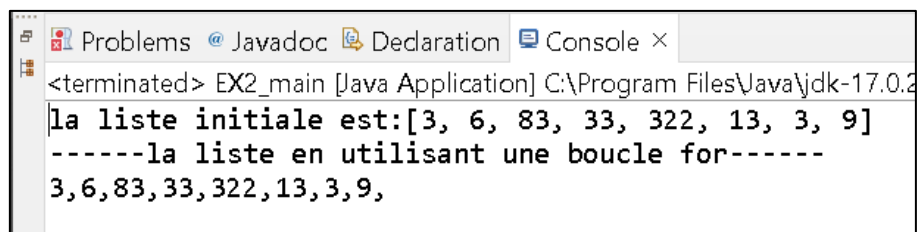
The screenshot shows an IDE console window with the following output:

```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
la liste en utilisant les iterateurs
3,6,83,33,322,13,3,9,
```

5. Affichage des éléments de la liste en utilisant une boucle 'for' :

```
//Q5
System.out.println("-----la liste en utilisant une boucle for-----");
for(int j=0; j<liste1.size(); j++)
{
    System.out.print(liste1.get(j)+",");
}
```

Après exécution on aura :



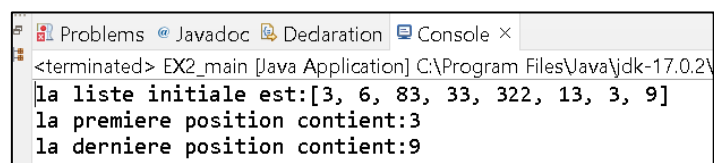
The screenshot shows an IDE console window with the following output:

```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0.2
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
-----la liste en utilisant une boucle for-----
3,6,83,33,322,13,3,9,
```

6. Affichage de la première et la deuxième position :

```
//Q6
System.out.println("la premiere position contient:"+liste1.getFirst());
System.out.println("la derniere position contient:"+liste1.getLast());
```

Après exécution on aura :



The screenshot shows an IDE console window with the following output:

```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0.2
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
la premiere position contient:3
la derniere position contient:9
```

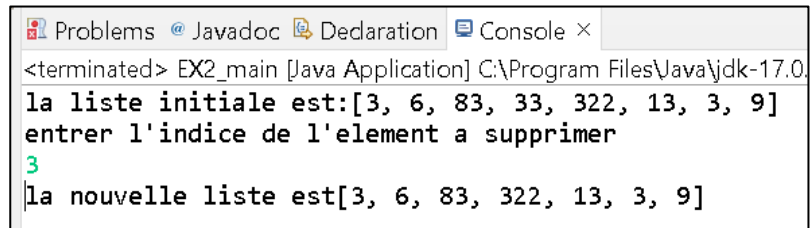
TRAVAUX PRATIQUES 5

– JAVA POO

7. Suppression d'un élément donne de la liste :

```
//Q7
System.out.println("entrer l'element a supprimer");
liste1.remove(sc.nextInt());
System.out.println("la nouvelle liste est"+liste1);
```

Après exécution on aura :



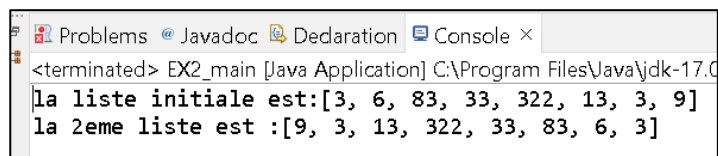
```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
entrer l'indice de l'element a supprimer
3
la nouvelle liste est[3, 6, 83, 322, 13, 3, 9]
```

8. Création d'une liste « liste2 » de type LinkedList qui contient les copies inversées de « liste1 » :

```
//Q8

LinkedList liste2= new LinkedList();
Collections.reverse(liste1);
liste2.addAll(liste1);
System.out.println("la 2eme liste est :"+ liste2);
```

Après exécution on aura :



```
<terminated> EX2_main [Java Application] C:\Program Files\Java\jdk-17.0
la liste initiale est:[3, 6, 83, 33, 322, 13, 3, 9]
la 2eme liste est :[9, 3, 13, 322, 33, 83, 6, 3]
```

FIN !