Exercices Java

HEIG VD

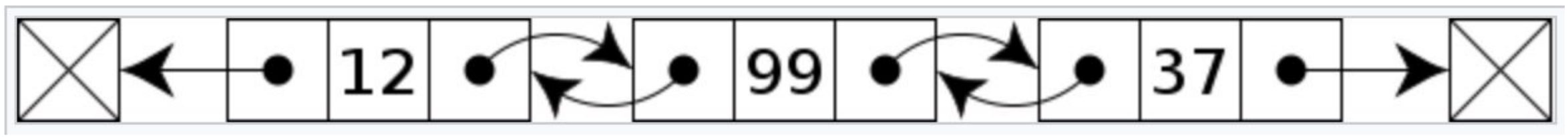
Hes·so
Haute Ecole Spécialisée
de Suisse occidentale

Introduction aux LinkedList<T>

Java possède plusieurs types de Collections dont les “Tableaux” et les “LinkedList<T>”. Quelques prioriétés de LinkedList<T>

- Une liste d’éléments doublement chainée, les éléments sont de type <T. T représente un type classe (***ne peut pas représenter un type primitif***).
- Insertion et suppression d’élément en O(1)
- Accès à un noeud par index en O(n)
- Recherche d’un noeud en O(n)
- Doublement chainée, e.g. navigable dans les deux sens

Contrairement aux tableaux, il est facilement possible d’insérer ou de supprimer des éléments.



Introduction aux LinkedList<T>

Exemple:

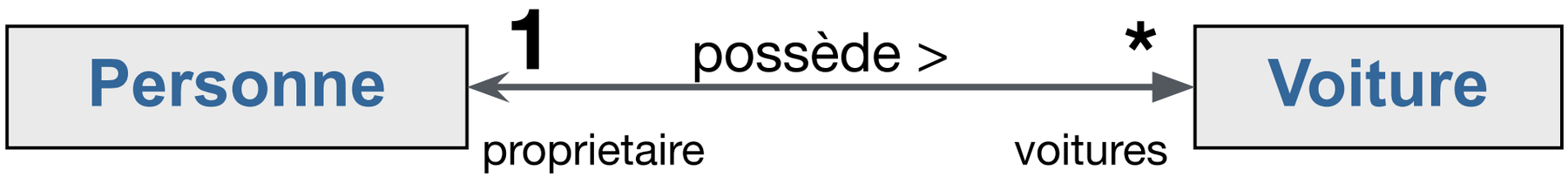
```
Voiture v = new Voiture(“bmw”);

// Crée une nouvelle liste de Voitures.
LinkedList<Voiture> voitures = new LinkedList<Voiture>();

// ajouter bmw à voitures:
voitures.add(v)

// v est-il dans voitures?
bool hasVoiture = voitures.contains(v);

// enlever v à voitures
voitures.remove(v);

// voir: https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html
```

Introduction aux LinkedList<T>

Exemple:

```
Voiture v = new Voiture(“bmw”);

// Crée une nouvelle liste de Voitures.
LinkedList<Voiture> voitures = new LinkedList<Voiture>();

// ajouter bmw à voitures:
voitures.add(v)

// v est-il dans voitures?
bool hasVoiture = voitures.contains(v);

// enlever v à voitures
voitures.remove(v);

// voir: https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html
```

Exercice: Voiture



Compléter 1 (Voiture)

```
public class Voiture {
    private String marque;
    private Personne proprietaire; // initialisé par défaut à null

    public Voiture(String marque) { /* … */ }
    public String getMarque() { /* … */ }
    public Personne getProprietaire() { /* … */ }

    public String toString() {
        return "Voiture " + marque
                + (proprietaire != null ? " (propriétaire: " + proprietaire.getNom() + ")"
                                        : " (sans propriétaire)");
    }
}
```

Compléter 2 (Personne)

```
import java.util.LinkedList;


class Personne {
    private String nom;
    private LinkedList<Voiture> voitures = new LinkedList<Voiture>();

    public Personne(String nom) { /* … */ }
    public String getNom() { /* … */ }

    public String toString(){
        String str = nom + " possède: ";
        for (Voiture v : voitures) str += v.getMarque() +"; ";
        return str;
    }
}
```

Compléter 3 (maintient de la cohérence)

```
public class Voiture {
    // setProprietaire doit maintinir la cohérence
    public void setProprietaire(Personne p)  { /* … */ } // maintient la cohérence
}
```

```
public class Personne {
    // ajouterVoiture maintient la cohérence
    public void ajouterVoiture(Voiture v) { /* … */ }
    // enleverVoiture maintient la cohérence
    void enleverVoiture(Voiture v) { /* … */ }
}
```