

Feuille 4 - ADTs

1 Classe Set – Listes

Rappel : on a défini la classe abstraite suivante :

```

1 abstract class Set {
2     public abstract Object [] toArray();
3     public abstract void add(Object o);
4     public abstract boolean isMember(Object o);
5     public abstract String toString();
6 }
```

Pour cet exercice on va programmer une sous-classe de la classe `Set` qui donne une implémentation concrète de sa fonctionnalité, en utilisant des listes chaînées. Une partie de la signature de notre classe sera :

```

1 class ListSet extends Set {
2     ListSet next;
3     Object o;
4     int taille;
5 }
```

1. Programmer les méthodes qui manquent.
2. Considérez le programme suivant. Est-il correcte? Que va-t-il afficher?

```

1 Set s = new ListSet();
2 s.add(1);
3 s.add(5);
4 s.add(7);
5 s.add(3);
6 s.add(7);
7 s.add("hi");
8 s.add("hi");
9 System.out.println(s);
```

2 Classe Set – Equals

Rappel : la méthode `equals` est défini dans la classe `Object`. Son objectif est de faire des comparaisons entre des objets (l'opérateur `==` fait des comparaisons des références). Pour cet exercice vous devez redéfinir cette méthode pour la classe `Set`.

1. Peut-on implémenter cette méthode dans la classe abstraite?
2. Donner une implémentation.