



Mini-projet 1

10



Présentation de mini-projet 1

Nom de l'atelier	Mini-projet1: l'arbre généalogique
Objectifs de l'atelier	A la fin de ce mini-projet, vous devez être capable de : <ul style="list-style-type: none">comprendre les concepts de base orientés objets : classe, objet, surcharge, membres statiques, composition et agrégationappliquer ces concepts dans la réalisation d'une petite application
Mots clés	classe, objet, membres statiques, composition, agrégation
Durée	libre mais ne dépasse pas un mois
Outils et environnements	Eclipse ou équivalent



Mini-projet 1 : L'arbre généalogique

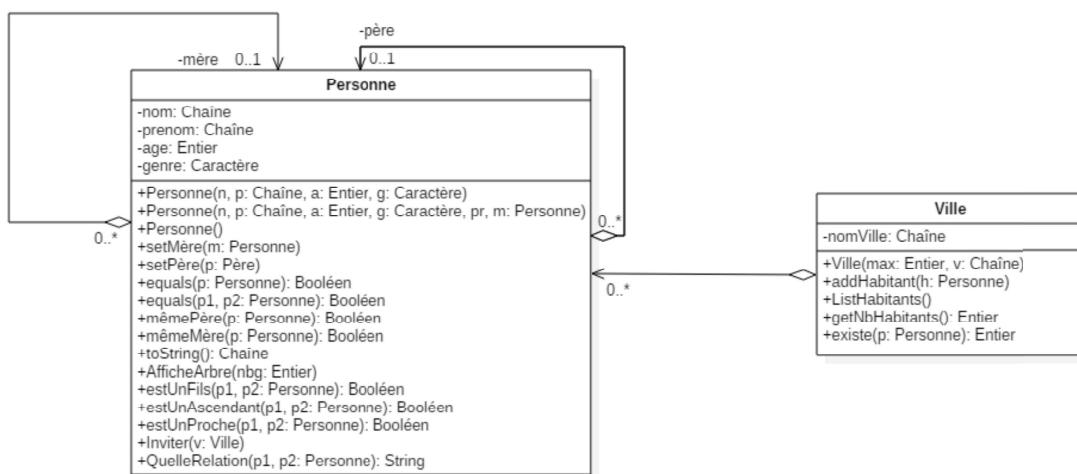
Énoncé

Dans ce mini-projet, vous allez exploiter vos connaissances autour des concepts orientés objet pour réaliser une application qui gère les **arbres généalogiques** des personnes d'une **ville**.

A partir de ces arbres, on peut déduire **les relations** entre les personnes (les frères, les parents, les ascendants, les descendants et les proches en général).

Une **ville** est un ensemble d'habitants (personnes) qui peuvent avoir des relations entre eux. Un habitant est ajouté à la ville s'il n'existe pas déjà. Une personne peut inviter les personnes qui sont **proches** et on peut qualifier la relation entre deux personnes.

Nous nous basons sur ce **diagramme de classes** qui contient une relation **réflexive** qui relie une personne à un autre.



Remarques

- On peut affecter le père et la mère d'une personne à travers le **constructeur** ou à l'aide des méthodes **setPère** et **setMère**.
- Les méthodes **mêmePère** et **mêmeMère** vérifient si deux personnes ont le même père ou la même mère.
- La méthode **sontFrères** permet de vérifier 2 personnes sont deux frères.
- La méthode **estUnFils** permet de vérifier si la première personne est un fils d'une autre personne.
- La méthode **estUnAscendant** permet de vérifier si la première personne est un descendant de la deuxième personne.

- La méthode **estUnDescendant** permet de vérifier si la première personne est un descendant de la deuxième personne.
- La méthode **estUnAscendant** permet de vérifier si la première personne est un ascendant de la deuxième personne.
- La méthode **estUnProche** permet de vérifier si la première personne a une relation quelconque avec la deuxième personne.
- La méthode **QuelleRelation** permet de vérifier le type de la relation entre deux personnes (Même, Frère, Parent, Fils, Ascendant, Descendant, Proche, Aucune).
- La méthode **equals** permet de vérifier si deux personnes sont identiques.
- La méthode **toString** permet de retourner le nom et le prénom d'une personne.
- La méthode **AfficheArbre** permet d'afficher l'arbre généalogique d'une personne.
- La méthode **Inviter** permet d'afficher la liste des invités (les proches) d'une personne habitant dans une ville.



Méthode

Écrire en Java les deux classes **Personne** et **Ville** et les tester à travers la classe de test **TestPersonne** suivante :

```

1 package genealogie;
2 public class TestPersonne {
3     public static void main(String[] args) {
4         Personne p1= new Personne ("Khelifa", "Afifa", 35, 'F');
5         Personne p2 = new Personne ("Khelifa", "Bechir", 60,'H');
6         Personne p3 = new Personne ("Helaoui", "Fatma", 55,'F');
7         p1.setPère(p2);
8         p1.setMère(p3);
9         Personne p4 = new Personne ("Zguerni", "Helmi", 8,'H');
10        p4.setMère(p1);
11        Personne p5= new Personne ("Khelifa", "Afifa", 35, 'F');
12        p5.setPère(p2);
13        p5.setMère(p3);
14        p1.AfficheArbre(1);
15        p2.AfficheArbre(1);
16        p3.AfficheArbre(1);
17        p4.AfficheArbre(1);
18        if (Personne.equals(p5,p3))
19            System.out.println ("Les 2 personnes sont les mêmes");
20        else
21            System.out.println ("Les 2 personnes ne sont pas les mêmes");
22        Personne p6= new Personne ("Khelifa", "Hanene", 37, 'F');
23        p6.setPère(p2);
24        p6.setMère(p3);
25        if (Personne.sontFrères(p1, p6))
26            System.out.println (p1 + " et " + p6 + " sont frères!");
27        else
28            System.out.println (p1 + " et " + p6 + " sont frères!");
29        if (Personne.estUnProche(p4, p6))
30            System.out.println (p4 + " est un proche de " + p6);
31        else
32            System.out.println (p4 + " n'est pas un proche de " + p6);
33        Ville v = new Ville (100, "Mahdia");
34        v.addHabitant(p1);
35        v.addHabitant(p2);v.addHabitant(p3);v.addHabitant(p4);v.addHabitant(p5);v.addHabitant(p6);
36        Personne p7 =new Personne ("Nahla", "Mejri", 38, 'F');
37        v.addHabitant(p7);
38        p1.Inviter(v);
39        String relation;
40        relation = Personne.QuelleRelation(p1, p2);
41        System.out.println ( p1 + " et " + p2 + " : " + relation);
42        relation = Personne.QuelleRelation(p1, p3);

```

```

42 System.out.println ( p1 + " et " + p3 + " : " + relation);
43 relation = Personne.QuelleRelation(p2, p1);
44 System.out.println ( p2 + " et " + p1 + " : " + relation);
45 relation = Personne.QuelleRelation(p3, p1);
46 System.out.println ( p3 + " et " + p1 + " : " + relation);
47 relation = Personne.QuelleRelation(p1, p4);
48 System.out.println ( p1 + " et " + p4 + " : " + relation);
49 relation = Personne.QuelleRelation(p1, p6);
50 System.out.println ( p1 + " et " + p6 + " : " + relation);
51 relation = Personne.QuelleRelation(p1, p7);
52 System.out.println ( p1 + " et " + p7 + " : " + relation);
53 relation = Personne.QuelleRelation(p2, p7);
54 System.out.println ( p2 + " et " + p7 + " : " + relation);
55 relation = Personne.QuelleRelation(p1, p5);
56 System.out.println ( p1 + " et " + p5 + " : " + relation);
57 relation = Personne.QuelleRelation(p3, p4);
58 System.out.println ( p3 + " et " + p4 + " : " + relation);
59 relation = Personne.QuelleRelation(p4, p3);
60 System.out.println ( p4 + " et " + p3 + " : " + relation);
61 }
62 }

```

Le résultat de cette classe est le suivant :

```

Rattachement effectué: entre Khelifa Afifa et Khelifa Bechir
Rattachement effectué: entre Khelifa Afifa et Helaoui Fatma
Rattachement effectué: entre Zguerni Helmi et Khelifa Afifa
Rattachement effectué: entre Khelifa Afifa et Khelifa Bechir
Rattachement effectué: entre Khelifa Afifa et Helaoui Fatma
Khelifa Afifa 35 ans
    Père : Khelifa Bechir 60 ans
    Mère : Helaoui Fatma 55 ans
Khelifa Bechir 60 ans
Helaoui Fatma 55 ans
Zguerni Helmi 8 ans
    Mère : Khelifa Afifa 35 ans
        Père : Khelifa Bechir 60 ans
        Mère : Helaoui Fatma 55 ans
Les 2 personnes ne sont pas les mêmes
Rattachement effectué: entre Khelifa Hanene et Khelifa Bechir
Rattachement effectué: entre Khelifa Hanene et Helaoui Fatma
Khelifa Afifa et Khelifa Hanene sont frères!
Zguerni Helmi est un proche de Khelifa Hanene
Les proches de Khelifa Afifa sont :
    Khelifa Bechir
    Helaoui Fatma
    Zguerni Helmi
    Khelifa Hanene
Khelifa Afifa et Khelifa Bechir : Fils
Khelifa Afifa et Helaoui Fatma : Fils
Khelifa Bechir et Khelifa Afifa : Parent
Helaoui Fatma et Khelifa Afifa : Parent
Khelifa Afifa et Zguerni Helmi : Parent
Khelifa Afifa et Khelifa Hanene : Frère
Khelifa Afifa et Nahla Mejri : Aucune
Khelifa Bechir et Nahla Mejri : Aucune
Khelifa Afifa et Khelifa Afifa : Même
Helaoui Fatma et Zguerni Helmi : Ascendant
Zguerni Helmi et Helaoui Fatma : Descendant

```



Complément

Ajouter un **constructeur** sans paramètre de la classe **Personne** qui permet d'initialiser une personne **à partir du clavier** tout en donnant toutes les informations disponibles concernant les **ascendants**. Le schéma suivant montre un exemple d'exécution de cette méthode

```
Nom: Riahi
Prenom: Karim
Femme? (O/N): 0
Age: 12
Avez-vous des informations sur la mère de Riahi Karim?1
Nom: Kilani
Prenom: Fatma
Femme? (O/N): 1
Age: 35
Avez-vous des informations sur la mère de Kilani Fatma?1
Nom: Bayaoui
Prenom: Hedia
Femme? (O/N): 1
Age: 65
Avez-vous des informations sur la mère de Bayaoui Hedia?0
Avez-vous des informations sur le père de Bayaoui Hedia?0
Rattachement effectué: entre Kilani Fatma et Bayaoui Hedia
Avez-vous des informations sur le père de Kilani Fatma?1
Nom: Kilani
Prenom: Saleh
Femme? (O/N): 0
Age: 67
Avez-vous des informations sur la mère de Kilani Saleh?0
Avez-vous des informations sur le père de Kilani Saleh?0
Rattachement effectué: entre Kilani Fatma et Kilani Saleh
Rattachement effectué: entre Riahi Karim et Kilani Fatma
Avez-vous des informations sur le père de Riahi Karim?1
Nom: Riahi
Prenom: Mabrouk
Femme? (O/N): 0
Age: 40
Avez-vous des informations sur la mère de Riahi Mabrouk?0
Avez-vous des informations sur le père de Riahi Mabrouk?0
Rattachement effectué: entre Riahi Karim et Riahi Mabrouk
```



Remarque

Vous pouvez enrichir votre projet en ajoutant d'autres méthodes qui recherche le nombre d'enfants d'une personnes déduit à partir d'une liste ou bien l'époux ou épouse d'une personne donnée.