UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT INFORMATIQUE

DIC_1

Java

TP N° 4

Exercice 1

Il s'agit de représenter les personnes ainsi que leurs liens de parenté. Par exemple une ***Personne*** a un papa ou un père ainsi qu'une maman ou une mère.

```
public class Personne {// Objets représentant les personnes
        protected Nom name;
        protected char sex; // 'M' or 'F'
        protected String id; // e.g., Numéro CNI
        protected Personne mother;
        protected Personne father;
}
```

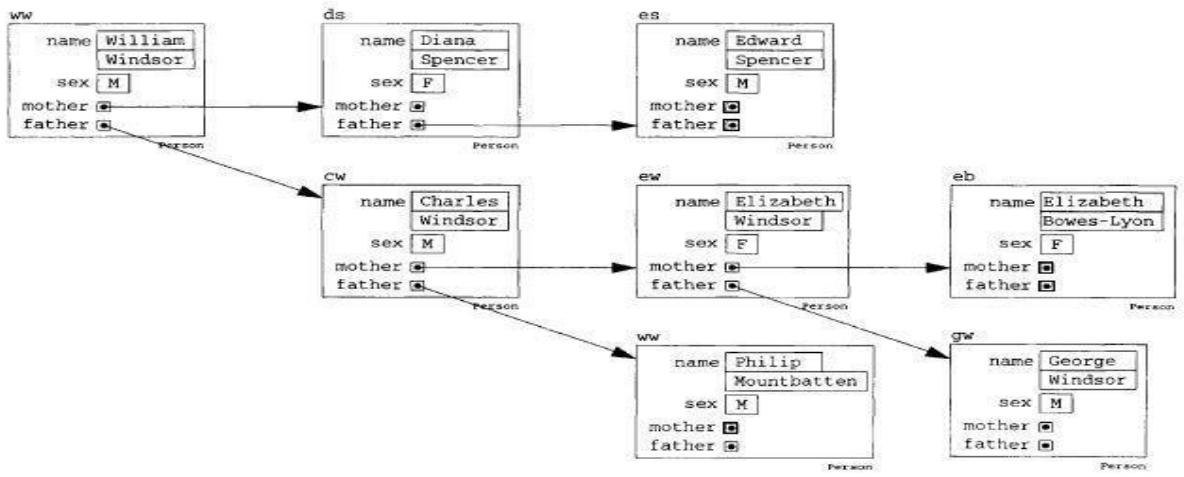
Avec la classe ***Nom***, ainsi définie :

```
public class Nom { // Objets représentant les noms de personne
private String first; // e.g., El Hadji "
private String middle; // e.g., "Bassirou"
private String last; // e.g., "TOURE"
}
```

1- Ajoutez les méthodes nécessaires

2- Puis implémentez la classe Personne

Ajoutez toutes les méthodes nécessaires à cette classe puis de créer une nouvelle classe permettant de tester la hiérarchie ci-dessous (exactement la même).



Formattez l'affichage de sorte à avoir l'affichage (de l'exemple précédent) ci-dessous:

```
William Windsor (M)
   mother: Diana Spencer (F)
      father: Edward Spencer (M)
   father: Charles Windsor (M)
      mother: Elizabeth Windsor (F)
         mother: Elizabeth Bowes-Lyon (F)
         father: George Windsor (M)
      father: Philip Mountbatten (M)
```

Exercice 2

Créez la classe ***Contacts*** qui permet de représenter une liste de contacts (répertoire téléphonique).

```
public class Contacts{
  // Objets représentant mes amis
Personne contact;

String telephone; // e.g., " 773587910"

Contats next; // prochain objet(ami) de la liste

static Contacts list;// Liste chainée d'amis

...
```

- Créez toutes les méthodes (constructeurs, accesseurs, modificateurs, etc.) dont vous avez besoin. MAIS AJOUTEZ les méthodes ci-dessous.
- ***static void print() ;*** Cette méthode permet de parcourir et d'afficher le répertoire si ce dernier n'est pas vide.
- ***static void add(Personne unePersonne, String sonTelephone);*** Cette méthode permet d'ajouter un nouveau contact dans la liste de contacts. Deux contacts ne doivent pas avoir le même numéro.
- ***static int find(Contacts);*** Cette méthode permet de rechercher un contact dans la liste de contacts.
- ***static void rm(Contacts);*** Cette méthode permet de supprimer un contact de la liste de contacts.
- Créez une classe ***Test Contacts*** pour tester la classe ***Contacts*** qui donne en sortie ce qui est en dessous.