

TP2 – Arbre, récursivité, stream et sérialisation. (10 %)

Modalités :

- Le travail doit être fait **OBLIGATOIREMENT** en équipe de 2 ou 3 personnes.
- Remise LÉA une archive zip comprenant. Le projet complet (TP2_XXYY où XX et YY sont vos initiales).
- Date de remise : 29 octobre 2017 à 20h.

Arbre généalogique

Vous avez le mandat de programmer une application qui permet de gérer un arbre généalogique. Vous aurez en fait à compléter le programme déjà en place d'une version simplifiée d'un arbre généalogique. En fait seulement une personne à un 0 ou plusieurs enfants. Les enfants peuvent aussi avoir 0 à plusieurs enfants.

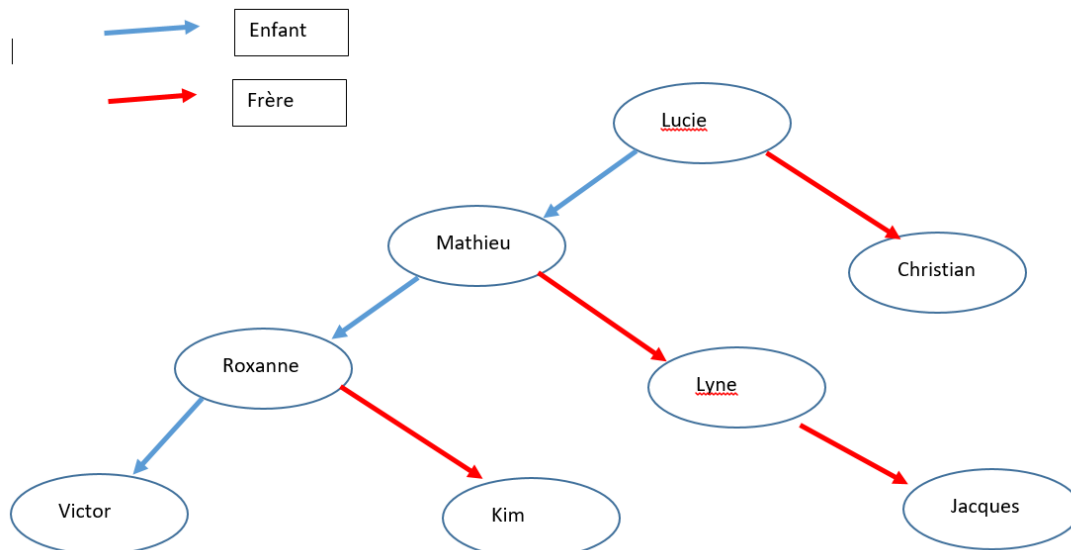
Affichage de l'Arbre :

```

Lucie
  Mathieu
    Roxanne
      Victor
        Kim
          Lyne
            Jacques
              Christian

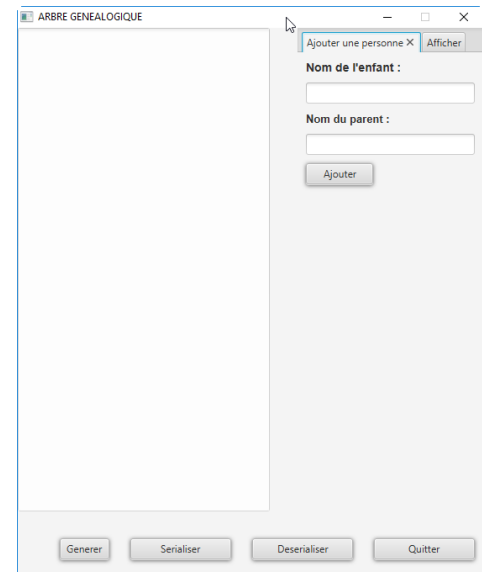
```

- Lucie est la mère de Mathieu, Lyne et Jacques.
- Mathieu est le père de Roxanne et Kim.
- Roxanne est la mère de Victor.



Description de l'interface :

- 1) Bouton **Générer** : permet de générer l'arbre avec plusieurs ajouts prédéfinis. Vous allez sauver bien du temps. Vous devez cliquer sur ce bouton pour activer les boutons de l'onglet Afficher.
- 2) Bouton **Sérialiser** : Permet de sérialiser tout l'arbre dans un fichier.
- 3) Bouton **Désérialiser** : Permet de désérialiser le fichier et mettre à jour l'arbre généalogique.
- 4) Onglet Afficher, permet d'afficher :
 - a. Arbre
 - b. Liste
 - c. Ensemble String
 - d. Liste String moins de 40 ans.



Votre mandant :

- Classe *Personne* :
 - o Ajouter un numéro d'assurance social (NAS) à la classe *Personne* et assurez-vous que 2 personnes soit uniques dans l'arborescence.
 - o Modifiez l'interface pour la saisie NAS autant pour l'enfant que pour le parent.
 - o Vous devrez modifier *ArbreGenealogique* pour ajouter des Personnes et non des String.
- Classe *ArbreBinaireDerive*, programmer les méthodes :
 - o **ajouter** : Vous devez ajouter un nœud dans l'Arbre. Vous pouvez utiliser la méthode *rechercherNoeud* de la même classe.
 - o **assistantRecuratif** : Voir la javadoc. Elle sera utilisée pour afficher l'arbre. Les enfants doivent être indentés d'un espace par rapport au parent.
 - o **getListeElement** : Cette méthode retourne l'arbre sous forme de liste de *Personne*. Vous devez **OBLIGATOIREMENT** faire de la récursivité dans cette méthode.
- Classe *ArbreGenealogique*
 - o **transformeSetString** : Cette méthode transforme la liste retournée par **getListeElement** en un ensemble de String. Vous devez obligatoirement utiliser les streams. Conserver cet ensemble dans un attribut de la classe *ArbreGenealogique* qui sera utilisé dans la méthode **getListePersonne40ansEtMoins**.
 - o **getListeString40ansEtMoins** : Cette méthode retourne une liste de nom String de moins de 40 ans à partir de l'ensemble créé dans la méthode **transformeSetPersonne**. Vous devez obligatoirement utiliser les streams.
 - o **serialize** : vous devez sérialiser l'arbre au complet en dehors du jar.
 - o **deserialize** : Vous devez désérialiser l'arbre au complet qui mettra à jour l'arbre.
- Classe *VueArbreGenealogique* et *Controleur* vous sont fournis et appelle déjà les méthodes publiques de la classe *ArbreGenealogique*.
- Rendre les classes suivantes génériques :
 - o *NoeudArbreDerive*
 - o *ArbreBinaireDerive*
- Vous devez aussi ajouter une fonctionnalité qui permet d'afficher la liste des ancêtres d'une personne.