

Problème Java  
Prof. Abdessamad Belangour  
SMI / S6

**Problème :**

Soit **Sexe** une énumération, **IPersonne** une interface, **Personne** une classe qui l'implémente et **Groupe** une classe qui s'en sert et dont les codes sont :

```
public enum Sexe { M, F }

public interface IPersonne {
    void marier(Personne conjoint);
    void divorcer(Personne conjoint);
    void afficherStatut();
}

public class Personne implements IPersonne {
    private String cin;
    private String nom;
    private Sexe sexe;
    private Personne conjoint;

    // On suppose que les getters/setters sont fournies en cas de besoin

    //constructeur paramétré 1 : pour une personne non encore mariée

    public Personne(String cin, String nom, Sexe sexe) {..... }

    // constructeur paramétré 2 : pour une personne mariée

    public Personne(String cin, String nom, Sexe sexe, Personne conjoint){
        this.cin = cin; this.nom = nom; this.sexe = sexe;
        this.conjoint = conjoint; //ici nous voulions garder référence au conjoint
    }

    // constructeur paramétré 3

    public Personne(String cin, String nom, Sexe sexe,String cinConjoint, String
    nomConjoint, Sexe sexeConjoint) { ..... }

    //méthodes
    public void marier(Personne conjoint) throws DéjàMariéException, MemeSexeException {
        ..... }
    public void divorcer() throws NonMariéException { ..... }
    public void afficherStatut() { .....}
}
```

```

class Groupe {
    private ArrayList liste;
    public Groupe() {    liste = new ArrayList() ;    remplirListe();    }
    private void remplirListe() { .....}
    public IPersonne getPersonne( String cne){.....}
    public int taille() { ..... }
    public void afficherToutLesStatuts(){.....}
    public void MarierCelibataires(){ .....}
}

```

### Questions :

**N.B. :** *Ne réécrivez pas toute la classe. Donnez juste le code des méthodes demandées !!!*

- 1) Donner le code du premier constructeur paramétré de la classe Personne.
- 2) Donner le code du troisième constructeur paramétré de la classe Personne.
- 3) Donner le code de la méthode « marier » de la classe Personne.  
**Attention** : vérifier que les deux personnes à marier sont de sexes opposés et qu'ils ne sont pas déjà mariées.
- 4) Donner le code de la méthode « divorcer » de la classe Personne.  
**Attention** : vérifier que les personnes à divorcer sont déjà mariées.
- 5) Donner le code des classes Exceptions DéjàMariéException, MemeSexeException et NonMariéException.
- 6) Donner le code de la méthode « afficherStatut » de la classe Personne qui affiche une phrase de type « *Ali est marié(e) à Sara* » ou « *Ali est célibataire* »
- 7) Donner le code de la méthode privée « remplirListe » de la classe Groupe qui permet d'initialiser notre liste à partir de la base de données « EtatCivil ».
- 8) Donner le code de la méthode getPersonne() de la classe Groupe ;
- 9) Donner le code de la méthode taille() de la classe Groupe ;
- 10) Donner le code de la méthode « afficherToutLesStatuts » de la classe Groupe qui parcourt la liste et affiche le statut marital de chacun des objets qu'elle contient.
- 11) Donner le code la méthode « MarierCelibataires » de la classe Groupe qui parcourt la liste des personnes et les marie hommes et femmes.

**Remarque :** la classe ArrayList stocke les objets qu'on lui passe sous forme d'objets de type object. Et voici quelques méthodes :

- void Add(object o) // ajout
- void Remove(object o) //suppression
- int indexOf(object o) //recherche
- ...