esprit Ecole Supérieure Privée d'Ingénierie et de Technologies	**Année Universitaire : 2015-2016** **Examen de Révision**	
Module : Conception orientée objet et programmation Java		**Documents autorisés** : Non
Enseignants : Equipe Java		**Nombre de pages** : 5
Date :	**Heure :**	**Durée** : 1h30
Classes : 3INFO A, 4INFINI		

GESTION D'UN HOPITAL : REVISION

Nous souhaitons développer une application simplifiée de gestion d'un hôpital. Pour réaliser notre application, nous vous demandons de compléter les classes présentées ci-dessous.

Les parties à compléter sont numérotées de 1 à 26. Vous devez *mettre dans vos copies le numéro correspondant à chaque code* ajouté. Vous *mettez juste la partie à compléter* sans réécrire aucune ligne de code déjà donnée dans l'énoncé.

Travail demandé :

En précisant le numéro de chaque partie à ajouter :

1) Complétez la classe **Patient.** Deux patients sont égaux en cas d'égalité de leurs cin ainsi que leurs numéros de sécurité sociale.
2) Complétez la classe **ListPatients** regroupant un ensemble de patients dans un **ArrayList.** Complétez les méthodes de cette classe sachant qu'elle implémente l'interface InterfacePatient.
3) Complétez la classe **Medecin.**
4) Complétez la classe **SetMedecins** qui regroupe un ensemble de Medecin dans un **HashSet**. Faites le nécessaire pour que le HashSet n'accepte pas les doublons.
5) Complétez la classe **Hopital** qui regroupe pour chaque Medecin l'ensemble de ses patients dans un **hashMap**.

   Remarque : **Toutes les classes sont présentées dans l'ANNEXE ci-dessous, vous êtes amenés à compléter juste le code incomplet.**

ANNEXE :

```
public class Patient {

    private int cin;
    private String nom;
    private String prenom;
    private int numSecuriteSociale;

  public Patient() {
    }

  public Patient(int cin, String nom, String prenom, int numSecuriteSociale) {
        this.cin = cin;
        this.nom = nom;
        this.prenom = prenom;
        this.numSecuriteSociale = numSecuriteSociale;
```

```
    }

    public int getCin() {
        return cin;
    }

    public void setCin(int cin) {
        this.cin = cin;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public int getNumSecuriteSociale() {
        return numSecuriteSociale;
    }

    public void setNumSecuriteSociale(int numSecuriteSociale) {
        this.numSecuriteSociale = numSecuriteSociale;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public String toString() {
        /*à completer*/ (1) (1 point)
    }

    public boolean equals(Object obj) {
        /*à completer*/ (2) (1 point)
    }
}
```

```
public interface InterfacePatient {
    public void ajouterPatient(Patient p);
    public void supprimerPatient(Patient p);
    public boolean rechercherPatient(Patient p);
    public boolean rechercherPatient(int cin);
    public void afficherPatients();
    public void trierPatientsParNom();
}
```

```
public class ListPatients         /*à completer*/ (3) { (0.5 point)

    private List<Patient> listP;

    public ListPatients(){
        /*à completer*/ (4) (0.5 point)
```

```
    }

    public void ajouterPatient(Patient p) {
        /*à completer*/ (5) (0.5 point)
    }

    public void supprimerPatient(Patient p) {
        /*à completer*/ (6) (0.5 point)
    }
     /* Avec l'api Stream */
    public boolean rechercherPatient(Patient p){
        /*à completer*/ (7)  (0.5 point)
    }
    /* Avec l'api Stream */
    public boolean rechercherPatient(int cin) {
        /*à completer*/ (8)     } (1 point)
    /* Avec lambda expression*/
    public void afficherPatients() {
        /*à completer*/ (9)   (1 point)
    }
    /* Avec l'api Stream */
    public void trierPatientsParNom() {
        /*à completer */ (10) (1 point)
    }
    /* Avec l'api stream */
    public void PatientSansRedondance(){
        /*à completer : Afficher la liste des patients sans redondance*/
    }
}
```

```
public class Medecin

    private int cin;
    private String nom;
    private String prenom;
    private int numOrdre;

    public Medecin(int cin, String nom, String prenom, int numOrdre) {
        this.cin = cin;
        this.nom = nom;
        this.prenom = prenom;
        this.numOrdre = numOrdre;
    }

    public int getCin() {
        return cin;
    }

    public void setCin(int cin) {
        this.cin = cin;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public int getNumOrdre() {
        return numOrdre;
    }
    public String toString() {
```

```
        /*à completer*/ (11) (1 point)
    }

    public boolean equals(Object obj) {
        /*à completer*/ (12) (1 point)
    }

    /*à completer*/ (13) indication : le HashSet n'accepte pas les doublons
    (1 point)
}
```

```
public class SetMedecins {

    private Set<Medecin> setM;

    public SetMedecins(){
        /*à completer*/ (14) (1 point)
    }

    public void ajouterMedecin(Medecin m) {
        /*à completer*/ (15) (0.5 point)
    }
     /* Avec l'api Stream */
    public boolean rechercherMedecin(int cin) {
        /*à completer*/ (16) (1 point)
    }
     /* Avec lambda expression */
    public void afficherMedecins() {
        /*à completer*/ (17)    } (1 point)
     /* Avec l'api Stream */
    public long nombreMedecins(){
        /*à completer*/
    }
     /* Avec l'api Stream */
    public TreeSet<Medecin> trierMedecins(){
        /*à completer*/
    }

}
```

```
public class Hopital {

    public Map<Medecin,ListPatients> medecinPatients;

    public Hopital(){
        /*à completer*/ (18) (0.5 point)

    }

    public void ajouterMedecin(Medecin m){

        /*à completer*/ (19) (0.5 point)
    }

    public void ajouterPatient(Medecin m,Patient p){
        if(medecinPatients.containsKey(m)){

            /*à completer*/ (20) (1 point)
        }else {
            /*à completer*/ (21) (1 point)       }
    }
  /*Avec lambda expression */
    public void afficherMap(){
```

```
        /*à completer*/
 }

    /* Afficher les patients d'un medecin dont le nom est "mohamed" */
    public void afficherMedcinPatients (Medecin m){
        /*à completer*/
    }
    /* Retourner les noms des patients dont le " numSecuriteSociale = 1" */
    public List<String> RetournerNomPatients (){
        /*à completer*/
    }
}
```