

Module : Conception orientée objet et programmation Java		Documents autorisés : Non
Enseignants : Sofiene G,Rochdi R,Adel KDIDI,Ibrahim B, Saif B, Bassem H ,Ibtihel S,Imen B,Sana BF,Emna BC		Nombre de pages :
Date : 12/07/2012	Heure : 9h	Durée : 1h30
Classes : 3 info A1,2,4,5,6,7,8,9,10,11,12,13 4 info B1,2,3,4,5 4 Tel B1,2 4INFNI		

GESTION D'UN HOPITAL

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 25. 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 **TriNomPatient** qui permet de comparer entre deux Patients en utilisant leurs noms (ordre croissant).
- 3) Complétez la classe **ListPatients** regroupant une liste de patients dans un **ArrayList**. Complétez les méthodes de cette classe sachant qu'elle implémente l'interface **InterfacePatient**.
- 4) Complétez la classe **Medecin**.
- 5) Complétez la classe **SetMedecins** qui regroupe un ensemble de médecins dans un **HashSet**. Faites le nécessaire pour que le HashSet n'accepte pas deux médecins qui ont les mêmes valeurs.
- 6) Complétez la classe **Hopital** qui regroupe pour chaque médecin la liste de ses patients dans un **TreeMap** (trié dans l'ordre croissant selon le cin médecin).

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 class TriNomPatient          /*à completer*/ (3) (0.5 point)
{
    public int compare(Patient p1, Patient p2) {
        /*à completer*/ (4) (0.5 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 compléter*/ (5) { (0.5 point)

    private List<Patient> listP;

    public ListPatients(){
        /*à compléter*/ (6) (0.5 point)
    }

    public void ajouterPatient(Patient p) {
        /*à compléter*/ (7) (0.5 point)
    }

    public void supprimerPatient(Patient p) {
        /*à compléter*/ (8) (0.5 point)
    }

    public boolean rechercherPatient(Patient p){
        /*à compléter*/ (9) (0.5 point)
    }

    public boolean rechercherPatient(int cin) {
        /*à compléter*/ (10) } (1 point)

    public void afficherPatients() {
        /*à compléter*/ (11) (1 point)
    }

    public void trierPatientsParNom() {
        /*à compléter*/ (12) (1 point)
    }
}

```

```

public class Medecin /*à compléter*/ (13) (0.5 point)

    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*/ (14) (1 point)
}

public boolean equals(Object obj) {
    /*à completer*/ (15) (1 point)
}

/*à completer*/ (16)
(2 point)

}

```

```

public class SetMedecins {

    private Set<Medecin> setM;

    public SetMedecins(){
        /*à completer*/ (17) (0.5 point)
    }

    public void ajouterMedecin(Medecin m) {
        /*à completer*/ (18) (0.5 point)
    }

    public boolean rechercherMedecin(int cin) {
        /*à completer*/ (19) (1 point)
    }

    public void afficherMedecins() {
        /*à completer*/ (20) (1 point)
    }

}

```

```

public class Hopital {

    public Map<Medecin,ListPatients> medecinPatients;

    public Hopital(){
        /*à completer*/ (21) (0.5 point)
    }

    public void ajouterMedecin(Medecin m){
        /*à completer*/ (22) (0.5 point)
    }

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

            /*à completer*/ (23) (1 point)
        }else {
            /*à completer*/ (24) (1 point)
        }
    }

    public void afficherMap(){
        /*à completer*/ (25) (1 point)
    }

}

```