

CONTRÔLE FINAL MASTER II - SRI

Questions de cours (5 pts)

Choisissez une réponse parmi les trois réponses (a, b, c) :

1. Les SGBDOO doivent supporter : (1 pt)
 - a. L'héritage simple ;
 - b. L'héritage multiple ;
 - c. Les deux.
2. Les SGBDOO : (1 pt)
 - a. Possèdent tous le même modèle de données orienté objet ;
 - b. Chacun possède son propre modèle de données orienté objet ;
 - c. N'ont pas de modèle de données orienté objet.
3. Un lien de composition relie une classe C_1 à une classe C_2 si : (1 pt)
 - a. Les objets de C_1 sont composés d'objets de C_2 et les objets de C_2 sont composés d'objets de C_1 .
 - b. Les objets de C_1 sont composés d'objets de C_2 ;
 - c. Les objets de C_2 sont composés d'objets de C_1 .
4. Un objet x , composant d'un objet y , est dit dépendant de y si : (1 pt)
 - a. La destruction de y entraîne automatiquement celle de x ;
 - b. La destruction de x entraîne automatiquement celle de y ;
 - c. Chaque objet se détruit à part.
5. Caractéristiques de l'identifiant d'un objet « oid » dans une base : (1 pt)
 - a. Dans quelques cas, deux objets peuvent partager le même identifiant « oid ».
 - b. Un objet dans une base peut prendre l'identifiant « oid » d'un autre objet s'il est détruit (supprimé) ;
 - c. Deux objets différents dans une base n'auront jamais le même identifiant, même s'ils n'existent pas en même temps ;

Modélisation (15 pts)

Un système de gestion d'une clinique conserve les informations suivantes concernant les médecins, les patients et leur historique de traitement.

- Chaque médecin est caractérisé par un identifiant, un nom, un prénom, une adresse (qui est composée du nom de la rue, du numéro dans la rue, du code postal et de la ville), sa spécialité et une liste de patients en attente.
- Un patient est caractérisé par un identifiant, un nom, un prénom, une adresse, ainsi qu'un numéro de sécurité sociale.
- La clinique utilise un système de codification des maladies. Une maladie est caractérisée par un code, un nom scientifique et une appellation commune.
- On désire conserver un historique des soins effectués au sein de la clinique. Ainsi, pour chaque hospitalisation, on enregistre la date de début et de fin, l'identifiant du patient et du médecin le traitant, ainsi que la maladie pour laquelle le patient a été soigné.
- Une clinique possède une base de données où sont enregistrés l'ensemble des médecins, l'ensemble des patients, l'ensemble des maladies connues et l'ensemble des hospitalisations.

1. Ecrivez en java les classes suivantes avec les attributs cités ci-dessus :
 - a. PERSONNE
 - b. MEDECIN
 - c. PATIENT
 - d. ADRESSE
 - e. MALADIE
 - f. HOSPITALISATION
 - g. CLINIQUE (classe principale)
2. Ecrivez la méthode **getMedecinLibre()** de la classe **CLINIQUE** qui retourne le médecin qui à le minimum de patients dans sa fille d'attente.
3. Ecrivez la méthode **arrivee_patient()** de la classe **HOSPITALISATION** qui permet d'enregistrer un patient dans la base de donnée de la clinique. Elle prend comme paramètres la date début de l'hospitalisation, le patient, le médecin libre et la maladie. Elle ajoute le patient arrivé à la liste des patients du médecin libre.
4. Ecrivez la méthode **sortie_patient()** de la classe **HOSPITALISATION** qui permet de mettre à jour la date de la fin de l'hospitalisation. Elle prend comme paramètres la date de sortie du patient. Elle supprime le patient de l'hospitalisation de la liste du médecin de cette hospitalisation puis elle enregistre l'hospitalisation dans la base de données de la clinique.
5. Supposant que la base de données de la clinique est remplie. Ecrivez la méthode **main()** de la classe **CLINIQUE** qui permet de créer un patient avec l'identifiant 1, le nom « toto », prénom « titi », numéro de sécurité sociale 1111 et l'adresse « 1, rue de la république, 05000 Batna ». Puis on cherche le médecin libre, on crée une maladie (code : 1, nom_scientifique : "A", appellation_commune : "B"), et enfin en crée un objet hospitalisation pour ce patient qui entre le 20/02/2011 et sort le 24/02/2011.

Bonne chance...

NB: Le corrigé type vous le trouverez sur les sites :
<http://www.larbiquezouli.com>

CORRECTION DU CONTRÔLE FINAL

MASTER II - SRI

Questions de cours (5 pts)

Choisissez une réponse parmi les trois réponses (a, b, c) :

1. Les SGBDOO doivent supporter : (1 pt)
a. L'héritage simple ;
2. Les SGBDOO : (1 pt)
b. Chacun possède son propre modèle de données orienté objet ;
3. Un lien de composition relie une classe C_1 à une classe C_2 si : (1 pt)
b. Les objets de C_1 sont composés d'objets de C_2 ;
4. Un objet x , composant d'un objet y , est dit dépendant de y si : (1 pt)
a. La destruction de y entraîne automatiquement celle de x ;
5. Caractéristiques de l'identifiant d'un objet « oid » dans une base : (1 pt)
c. Deux objets différents dans une base n'auront jamais le même identifiant, même s'ils n'existent pas en même temps ;

Modélisation (15 pts)

Le code JAVA :

La classe PERSONNE : (1 pt)

```
public class PERSONNE {  
    public int id;  
    public String nom;  
    public String prenom;  
    public ADRESSE adresse;  
}
```

La classe PATIENT : (1 pt)

```
public class PATIENT extends PERSONNE {  
    public int nss;  
}
```

La classe MEDECIN : (1 pt)

```
import java.util.LinkedList;  
public class MEDECIN extends PERSONNE {  
    public String specialite;  
    public LinkedList<PATIENT> listPatients;  
}
```

La classe MALADIE (1 pt)

```
public class MALADIE {  
    public int code;  
    public String nom_scientifique;  
    public String appellation_commune;  
}
```

La classe ADRESSE : (1 pt)

```
public class ADRESSE {  
    public int num;  
    public String rue;  
    public int codePostal;  
}
```

```

        public String ville;
    }

```

La classe HOSPITALISATION : (1 pt)

```

import java.util.Date;
import com.db4o.ObjectContainer;
public class HOSPITALISATION {
    public Date date_debut;
    public Date date_fin;
    public PATIENT patient;
    public MEDECIN medecin;
    public MALADIE maladie;
    public void arrivee_patient(ObjectContainer db, Date d, PATIENT p,
                                MEDECIN medecinLibre, MALADIE m) (1.5 pts)
    {
        db.set(p);
        patient = p;
        medecin = medecinLibre;
        medecin.listPatients.add(p);
        maladie = m;
        date_debut = d;
    }
    public void sortie_patient(ObjectContainer db, Date f) (1.5 pts)
    {
        date_fin = f;
        medecin.listPatients.remove(patient);
        db.set(this);
    }
}

```

La classe CLINIQUE : (1 pt)

```

import java.io.File;
import java.util.Date;
import java.util.LinkedList;
import com.db4o.Db4o;
import com.db4o.ObjectContainer;
import com.db4o.ObjectSet;
public class CLINIQUE {
    public static String DBOFILENAME = "c:/myClinique.dbo";
    public ObjectContainer db;
    CLINIQUE() (1 pt)
    {
        //Préparation de la base de données objet
        new File(DBOFILENAME).delete();
        ObjectContainer db = Db4o.openFile(DBOFILENAME);
    }
    public MEDECIN getMedecinLibre() (1.5 pts)
    {
        MEDECIN medecinLibre = null, medecin = null;
        int sizeMin = 1000;
        int idx = 0;
        ObjectSet result = db.get(MEDECIN.class);
        for (int i=1; i<=result.size(); i++) {
            medecin = (MEDECIN) result.next();
            int size = medecin.listPatients.size();
            if (size < sizeMin) {
                medecinLibre = medecin;
                sizeMin = size;
                idx = i-1;
            }
        }
    }
}

```

```

    }
    return medecinLibre;
}
public void main() (1.5 pts)
{
    try {
        PATIENT p = new PATIENT();
        p.id = 1;
        p.nom = "toto";
        p.prenom = "titi";
        p.nss = 1111;
        p.adresse = new ADRESSE();
        p.adresse.num = 1;
        p.adresse.rue = "république";
        p.adresse.codePostal = 05000;
        p.adresse.ville = "Batna";
        Date d = new Date(2011, 02, 20);
        Date f = new Date(2011, 02, 24);
        MEDECIN medecinLibre = getMedecinLibre();
        MALADIE m = new MALADIE();
        m.code = 1;
        m.nom_scientifique = "A";
        m.appellation_commune = "B";
        HOSPITALISATION hospitalisation = new HOSPITALISATION();
        hospitalisation.arrivee_patient(db, d, p, medecinLibre, m);
        hospitalisation.sortie_patient(db, f);
    } finally {
        db.close(); (0.5 pt)
    }
}
}

```