

CONTRÔLE FINAL

BASES DE DONNÉES ORIENTÉES OBJETS

Questions de cours (5 pts)

1. Qu'est-ce qu'une contrainte inverse ?
2. Est-ce que l'agrégation est un type de relation entre les classes des objets ?
3. Un lien de composition relie une classe C_1 à une classe C_2 si :
 - 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. L'héritage multiple est la cause d'un conflit dans les SGBDO. Quelles solutions proposent les SGBDO pour résoudre ce conflit ?
5. Un objet composant peut être, ou non, partagé entre plusieurs objets de la même classe composite ou bien partagé entre les classes composites ?

Modélisation (15 pts)

Un garage décide d'informatiser la gestion des véhicules qu'il met en vente. Pour cela il utilise une base de données orientée objets pour conserver les informations suivantes :

- Un véhicule est une instance d'un modèle de véhicule. Un modèle est caractérisé par son nom, l'année de sortie, le numéro d'immatriculation, son prix de base, le nombre de portes, le nombre de places et le type du moteur.
- Le type du moteur est caractérisé par la cylindrée et la puissance du moteur.

1. Ecrivez en java les classes suivantes avec les attributs cités ci-dessus :
 - a. MODELE
 - b. MOTEUR
 - c. GARAGE (classe principale)
2. Ecrivez la méthode **getLowCostModel()** de la classe **GARAGE** qui retourne le modèle de véhicule le moins cher.
3. Ecrivez la méthode **getModelByEngine()** de la classe **GARAGE** qui prend en argument un type de moteur et qui affiche les noms des de véhicules qui ont ce type de moteur.
4. Supposant que la base de données du garage est remplie. Ecrivez la méthode **main()** de la classe **GARAGE** qui permet de :
 - a. créer un type de moteur avec la cylindrée 3 et la puissance 6 chevaux ;
 - b. créer un modèle de véhicule avec le nom « MODEL1 », année de sortie « 2010 », numéro d'immatriculation 123456 ; le prix 1000000,00, le nombre de portes 5, le nombre de place 5 et un moteur de 3 cylindres et 6 chevaux de puissance ;

- c. créer un modèle de véhicule avec le nom « MODEL2 », année de sortie « 2011 », numéro d'immatriculation 7890123 ; le prix 900000,00, le nombre de portes 5, le nombre de place 5 et un moteur de 3 cylindres et 6 chevaux de puissance ;
- d. Afficher les noms des modèles de véhicules enregistrés dans la base qui ont un moteur de 3 cylindres et 6 chevaux de puissance ;
- e. Chercher le véhicule le moins cher on affichant son nom et son prix ;

Bonne chance...

NB: Le corrigé type vous le trouverez sur le site :
<http://www.larbiguezouli.com>

CORRECTION DU CONTRÔLE FINAL

BASES DE DONNÉES ORIENTÉES OBJETS

Questions de cours (5 pts)

1. Qu'est-ce qu'une contrainte inverse ?
L'objet composite dépend de ses objets composants.
2. Est-ce que l'agrégation est un type de relation entre les classes des objets ?
Non, l'agrégation est un type de relation entre les objets
3. Un lien de composition relie une classe C_1 à une classe C_2 si :
 - 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 .**Les objets de C_1 sont composés d'objets de C_2**
4. L'héritage multiple est la cause d'un conflit dans les SGBDO. Quelles solutions proposent les SGBDO pour résoudre ce conflit ?
 - **soit ils refusent la définition d'un tel héritage;**
 - **soit ils demandent au concepteur de choisir la sur-classe « dominante » dont la sous-classe héritera;**
 - **soit ils appliquent une règle (par exemple la première classe citée dans la clause `is-a`).**
5. Un objet composant peut être, ou non, partagé entre plusieurs objets de la même classe composite ou bien partagé entre les classes composites ?
Un objet composant peut être, ou non, partagé entre plusieurs objets de la même classe composite

Modélisation (15 pts)

Un garage décide d'informatiser la gestion des véhicules qu'il met en vente. Pour cela il utilise une base de données orientée objets pour conserver les informations suivantes :

- Un véhicule est une instance d'un modèle de véhicule. Un modèle est caractérisé par son nom, l'année de sortie, le numéro d'immatriculation, son prix de base, le nombre de portes, le nombre de places et le type du moteur.
- Le type du moteur est caractérisé par la cylindrée et la puissance du moteur.

1. Ecrivez en java les classes suivantes avec les attributs cités ci-dessus :
 - a. MODELE (2 pts)

Réponse :

```
public class Modele { (0.25 pts)
    public String nom; (0.25 pts)
    public int annee; (0.25 pts)
    public String immatriculation; (0.25 pts)
    public float prix; (0.25 pts)
    public int nbportes; (0.25 pts)
    public int nbplaces; (0.25 pts)
    public Moteur moteur; (0.25 pts)
}
```

b. MOTEUR (0.75 pts)

Réponse :

```
public class Moteur { (0.25 pts)
    int cylindree; (0.25 pts)
    int puissance; (0.25 pts)
}
```

c. GARAGE (classe principale) (1.25 pts)

Réponse :

```
public class Garage { (0.25 pts)

    public static String DBOFILENAME = "c:/myClinique.dbo"; (0.25 pts)
    public ObjectContainer db; (0.25 pts)

    public void main() {
        //Préparation de la base de données objet
        new File(DBOFILENAME).delete();
        ObjectContainer db = Db4o.openFile(DBOFILENAME); (0.25 pts)
        try {
            //Traitement des questions suivantes
            .....
        } finally {
            db.close();(0.25 pts)
        }
    }
}
```

2. Ecrivez la méthode **getLowCostModel()** de la classe **GARAGE** qui retourne le modèle de véhicule le moins cher. (2 pts)

Réponse :

```
public Modele getLowCostModel() (0.25 pts)
{
    Modele my_modele = null, lowCostModel = null;
    float min = 9999999F;
    ObjectSet result = db.query(Modele.class); (0.25 pts)
    for (int i=1; i<=result.size(); i++) { (0.25 pts)
        my_modele = (Modele) result.next(); (0.25 pts)
        if (my_modele.prix < min) { (0.25 pts)
            min = my_modele.prix; (0.25 pts)
            lowCostModel = my_modele; (0.25 pts)
        }
    }
    return lowCostModel; (0.25 pts)
}
```

3. Ecrivez la méthode **getModelByEngine()** de la classe **GARAGE** qui prend en argument un type de moteur et qui affiche les noms des de véhicules qui ont ce type de moteur. (1.75 pts)

Réponse :

```
Public void getModelByEngine(Moteur my_moteur) (0.25 pts)
{
    Modele my_modele = new Modele(); (0.25 pts)
    my_modele.moteur = my_moteur; (0.25 pts)
    ObjectSet result = db.queryByExample(my_modele); (0.25 pts)
    for (int i=1; i<=result.size(); i++) { (0.25 pts)
        Modele modele_found = (Modele) result.next(); (0.25 pts)
    }
}
```

```

        System.out.println("Modèle: " + modele_found.nom); (0.25 pts)
    }
}

```

4. Supposant que la base de données du garage est remplie. Ecrivez la méthode **main()** de la classe **GARAGE** qui permet de :

a. créer un type de moteur avec la cylindrée 3 et la puissance 6 chevaux ; (0.75 pts)

Réponse :

```

Moteur my_moteur = new Moteur(); (0.25 pts)
my_moteur.cylindree = 3; (0.25 pts)
my_moteur.puissance = 6; (0.25 pts)

```

b. créer un modèle de véhicule avec le nom « MODEL1 », année de sortie « 2010 », numéro d'immatriculation 123456 ; le prix 1000000,00, le nombre de portes 5, le nombre de place 5 et un moteur de 3 cylindres et 6 chevaux de puissance ; (2 pts)

Réponse :

```

//Création du modèle "MODELE1"
Modele my_modele1 = new Modele(); (0.25 pts)
my_modele1.nom = "MODELE1"; (0.25 pts)
my_modele1.annee = 2010; (0.25 pts)
my_modele1.immatriculation = "123456"; (0.25 pts)
my_modele1.prix = 1000000F; (0.25 pts)
my_modele1.nb_portes = 5; (0.25 pts)
my_modele1.nb_places = 5; (0.25 pts)
my_modele1.moteur = my_moteur; (0.25 pts)
db.store(my_modele1);

```

c. créer un modèle de véhicule avec le nom « MODEL2 », année de sortie « 2011 », numéro d'immatriculation 7890123 ; le prix 900000,00, le nombre de portes 5, le nombre de place 5 et un moteur de 3 cylindres et 6 chevaux de puissance ; (2 pts)

Réponse :

```

//Création du modèle "MODELE2"
Modele my_modele2 = new Modele(); (0.25 pts)
my_modele2.nom = "MODELE2"; (0.25 pts)
my_modele2.annee = 2011; (0.25 pts)
my_modele2.immatriculation = "7890123"; (0.25 pts)
my_modele2.prix = 900000F; (0.25 pts)
my_modele2.nb_portes = 5; (0.25 pts)
my_modele2.nb_places = 5; (0.25 pts)
my_modele2.moteur = my_moteur; (0.25 pts)
db.store(my_modele2);

```

d. Afficher les noms des modèles de véhicules enregistrés dans la base qui ont un moteur de 3 cylindres et 6 chevaux de puissance ; (0.5 pts)

Réponse :

```

// Affichage des noms des modèles de véhicules enregistrés dans la base
getModelByEngine(my_moteur); (0.5 pts)

```

e. Chercher le véhicule le moins cher on affichant son nom et son prix ; (1 pt)

Réponse :

```

//Recherche du véhicule le moins cher
Modele modele_moins_cher = getLowCostModel(); (0.5 pts)

```

```
System.out.println("Le véhicule le moins cher est: " +  
modele_moins_cher.nom + " qui a le prix: " +  
modele_moins_cher.prix); (0.5 pts)
```

(1 pt) sur la présentation de la feuille.