

22053



Génie Logiciel (DSI310)

Examen Final

Date : 02/02/2024

Durée : 1h30

Filière : DSI

***** Tous les documents sont interdits *****

Exercice 1 : Modélisation

Il s'agit d'établir le schéma conceptuel d'une base de données pour la gestion des formations payantes d'un établissement :

- Un cours est caractérisé par un numéro unique, un libellé et une durée en minutes (par exemple 120mn pour un cours de 2h).
- Les cours sont organisés sous forme de sessions. Un cours peut faire l'objet de plusieurs sessions et chaque session correspond à un seul cours.
- Une session est caractérisée par un numéro unique, une date de début et un prix.
- Une session est le plus souvent assurée par plusieurs enseignants et est placée sous la responsabilité de l'un des enseignants (le coordinateur). Il n'est pas impossible que plusieurs sessions aient un même coordinateur.
- Un enseignant peut intervenir dans plusieurs sessions au cours de l'année.
- On désire mémoriser le nombre d'heures effectué par un enseignant pour chaque session.
- Un enseignant est caractérisé par un matricule, un nom et un numéro de téléphone.
- Chaque session est suivie par un certain nombre d'étudiants. Un étudiant peut-être un étudiant interne à l'établissement ou un étudiant externe c'est-à-dire qui vient d'un autre établissement.
- Un étudiant est caractérisé par un matricule un nom et un email.
- Dans le cas des étudiants externes, on enregistre leurs numéros de téléphone ainsi que le nom de leur établissement d'origine.
- On désire pouvoir gérer d'une manière séparée les étudiants internes d'une part, et les étudiants externes d'autre part.

Remarque : On considère les situations suivantes à prendre en considération :

- A partir d'une session donnée on doit pouvoir lister tous ses enseignants y compris le coordinateur.
- A partir d'un enseignant donné, on peut connaître toutes les sessions qu'il enseigne ou coordonne.
- Si on connaît un cours, alors on connaît toutes ses sessions programmées. De même, si un cours est supprimé, pour une raison ou une autre, alors toutes ses sessions doivent être supprimées.

Travail demandé :

1. Établir un diagramme de classes pour cette base de données en optimisant le maximum possible.
2. Donner brièvement le squelette du code java des classes session et enseignant.



Exercice 2 : Reverse Engineering

El utilisant les techniques de reverse engineering (retro-conception), donner un diagramme de classes correspondant aux classes A, B, C et D suivantes.

```

public class A {
    private String a1, a2;
    public List<C> c = new ArrayList<C> ();
    public List<D> d = new ArrayList<D> ();
}
  
```

```

public class C {
    private int c1;
    public List<A> a = new ArrayList<A> ();
}
  
```

```

public class B extends C {
    private String b1;
    private int b2;
}
  
```

```

public class D {
    public String d1;
    public List<B> b = new ArrayList<B> ();
}
  
```

Exercice 3 : Cas d'utilisation

On se propose de réaliser une application mobile destinée aux banques mauritaniennes afin de constituer une interface entre le client et sa banque et offrir un ensemble de services pour la clientèle.

Un client, avec son smartphone, peut se connecter au système pour vérifier son solde, effectuer un virement entre comptes et faire des paiements. Les paiements peuvent se faire en ouguiya ou en devise. Certains clients VIP peuvent en plus commander un chéquier ou demander un crédit via l'application. Toute transaction doit être sécurisée et nécessitée par conséquent une authentification.

Donner un diagramme de cas d'utilisation pour modéliser cette description.

Exercice 4 : Junit

On définit une classe **Division** permettant de d'effectuer une opération de division.

Proposer une classe de test pour tester la fonction *divide*.
 Proposer un cas de test qui permet de détecter une erreur.

```

Public class Division{
    public int divide(int a, int b) {
        if(b==0)
            boolean signe=false;
        int res=0;
        if(a<0) {
            signe=true;
        }
        if(a<b) {
            res=0;
        }
        if(b<0) {
            signe=!signe;
            b=-b;
        }
        while(a>0) {
            a=subtract(a,b);
            res++;
        }
        if(signe)
            res=-res;
        return res;
    }
}
  
```

- Bonne Chance -