



FEDERATION EUROPEENNE DES ECOLES
EUROPEAN FEDERATION OF SCHOOLS

Organisation non gouvernementale dotée du statut participatif auprès du Conseil de l'Europe
NGO enjoying participatory status with the Council of Europe

UE D - TECHNIQUES PROFESSIONNELLES

Master Européen d'Informatique

UC D41.2 - Développement

Tous supports papiers et matériels sont interdits

Les réponses au QCM doivent être reportées sur la fiche optique fournie

Type d'épreuve : QCM et Exercice pratique

Durée : 3 heures

Session : Juin 2012

UC D41.2 - DEVELOPPEMENT

BAREME DE NOTATION QCM

Bonne réponse	3 points
Absence de réponse	0 point
Réponse erronée	-1 point

Nombre de points possible : **150 points**

Pour chacune des questions une seule réponse est possible

BAREME DE NOTATION EXERCICE PRATIQUE

Dossier 1 - Langages & Développement	100 points
Dossier 2 - Modélisation, Conceptualisation & Bases de données relationnelles	200 points
Total	300 points

1/ QCM : 45 minutes

Les différentes questions de ce QCM portent sur l'ensemble du cours de Développement d'application et bases de données, veuillez donc à bien lire chaque question attentivement.

Partie 1 : Langages & Développement

1. Pour transformer un code lisible en code compréhensible par la machine, on utilise un :
 - a. compilateur
 - b. éditeur de lien
 - c. assembleur
 - d. débogueur
2. Java est un langage :
 - a. compilé
 - b. interprété
 - c. compilé et interprété
 - d. ni compilé ni interprété
3. Pour avoir un commentaire sur une ligne d'un code Java, on place en début de ligne :
 - a. /**/
 - b. //
 - c. #
 - d. %
4. Dans le langage java, lequel de ces identificateurs est incorrect ?
 - a. CpChar
 - b. nb_premier
 - c. loop\$
 - d. 5_line
5. Quelle proposition n'a pas sa place parmi ces choix en langage Java ?
 - a. instanceof
 - b. while
 - c. select
 - d. new
6. En langage Java, pour déclarer une variable de type booléen de valeur « Vrai » on écrit :
 - a. boolean bool = new boolean(true) ;
 - b. enum boolean = {true, false} ; boolean bool = true ;
 - c. int bool = 1 ;
 - d. boolean bool = true;
7. Laquelle de ces affirmations est correcte ?
 - a. une classe doit implémenter plusieurs classes mais ne peut étendre qu'une seule interface
 - b. une classe peut implémenter plusieurs interfaces mais n'étend qu'une seule classe
 - c. une classe doit implémenter une seule interface et étendre plusieurs autres classes
 - d. une classe peut implémenter une seule classe et étendre plusieurs autres classes

8. Quel mot est réservé dans Java ?

- a. struct
- b. union
- c. template
- d. static

9. Quel mot clé utilise-t-on dans la déclaration d'une méthode pour l'empêcher d'être redéfinie ?

- a. finally
- b. const
- c. private
- d. final

10. De quel type primitif est le littéral 2.25 en Java ?

- a. short
- b. double
- c. long
- d. int

11. Dans Java la déclaration d'une variable locale se fait dans :

- a. une classe
- b. une méthode
- c. un objet
- d. une interface

12. Une évolution de la notion de module, permettant de réunir au sein d'un classifieur la structure et le comportement d'un type donnée, serait :

- a. héritage
- b. interface
- c. polymorphisme
- d. encapsulation

13. Le mot réservé « native » utilisé dans une déclaration d'une méthode en java permet de :

- a. spécifier que le corps de la méthode se trouve dans un autre fichier java
- b. indiquer que l'accès à la méthode n'est permis qu'à la classe dans laquelle elle est définie
- c. indiquer que cette méthode n'est pas codée en Java
- d. spécifier que la méthode est à l'origine codée en Java

14. `public class true{}` marque une erreur car :

- a. il manque un point-virgule à la fin
- b. « true » est un mot réservé
- c. la classe « true » doit contenir au moins une méthode
- d. il n'y a aucune erreur

15. Que signifie XSL ?

- a. Xml Stylesheet Language
- b. Extensible Stylesheet Language
- c. Executive Symbole Language
- d. Extensible System Language

16. Que signifie « `simplexml_load_file` » ?

- a. une instruction XML qui permet de charger tout fichier
- b. une fonction PHP qui permet de parser ou convertir un fichier XML
- c. une instruction XML pour charger un fichier PHP
- d. aucune signification

17. A quoi sert la déclaration : `<!DOCTYPE biblio SYSTEM "biblio.dtd">` ?

- a. une déclaration de feuille de style
- b. une déclaration de type de document interne
- c. une déclaration de type de document externe
- d. un commentaire

18. Quelle est la ligne correcte ?

- a. `<Tableau> <souligne><gras>Titre</gras></souligne></tableau>`
- b. `<tableau> <gras><souligne>Titre</gras></souligne></tableau>`
- c. `<tableau> <souligne><gras>Titre</gras><souligne></tableau>`
- d. `<tableau> <souligne><gras>Titre</gras></souligne></tableau>`

19. Que définit l'élément « `sequence` » ?

- a. un groupe d'éléments
- b. une structure de choix entre plusieurs éléments possibles
- c. un à plusieurs éléments devant obligatoirement apparaître dans un ordre prédéfini
- d. une liste par dérivation de types de données atomiques existants

20. Un document XML a obligatoirement un :

- a. ou plusieurs éléments racine
- b. seul élément racine
- c. élément racine sans attributs
- d. ou plusieurs éléments racines avec attributs

21. Perl est un :

- a. langage de programmation interprété adapté au traitement des fichiers textes et tâches d'administration des systèmes et des applications
- b. langage générique qui permet de définir différents espaces noms
- c. langage basé sur le concept d'appel procédural uniquement, permettant la réutilisation du même code à différents emplacements
- d. environnement de développement de logiciel

22. Que sera le résultat de l'exécution de ce segment de code en Perl ?

```
$msg = « 'zeus a été a  
suez' ! » ;  
$msg =~  
s/('w+)/reverse($1)/e ;  
Print $msg  
Exit 0
```

- a. ! zeus a été a suez
- b. erreur
- c. zeus a été a suez !
- d. ! 'zeus a été a suez'

23. Quelle expression est vraie ?

- a. '0'
- b. 0
- c. '00'
- d. ''0''

24. Soit $@s = (10,1,2,"age",20)$, que résulte de cette affectation : « (\$a, @t, @u, \$b) = @s; » ?

- a. $a = 10$, $@t = (1,2,"age",20)$, $@u = ()$, $\$b = \text{undef}$.
- b. $\$a = \text{undef}$, $@t = (10,1,2,"age")$, $@u = ()$, $\$b = 20$.
- c. $\$a = \text{undef}$, $@t = (10,1,2)$, $@u = ("age",20)$, $\$b = \text{undef}$.
- d. $\$a = 10$, $@t = (1)$, $@u = (2,"age")$, $\$b = 20$.

25. Soit $@t=(1,7,8,4,3,0)$ quelle est la valeur de @t après l'exécution de l'instruction : « unshift(@t,5,6); » ?

- a. $@t=(1,7,8,5,6)$.
- b. $@t=(1,7,8,4,6,3,0)$.
- c. $@t=(1,7,8,4)$.
- d. $@t=(5,6,1,7,8,4,3,0)$.

26. Le logo Linux est un :

- a. manchot
- b. remard
- c. gnou
- d. spirale

27. Le compilateur C du projet GNU est :

- a. gcc
- b. gdb
- c. gdc
- d. gnc

28. Sachant que : $\text{attributs} == 130$ et $\text{vrai} == 0$, que vaut « nombre » après ces deux itérations :

$\text{nombre} = (\text{vrai} || \text{attributs} >= 100) ? 20 : 30;$

$\text{nombre} = (\text{nombre} == 20 \ \&\& (\text{vrai} \ \&\& \text{attributs} >= 100)) ? 40 : 50;$

- a. 20
- b. 30
- c. 40
- d. 50

29. Laquelle de ces boucles n'existe pas en C ?

- a. for
- b. repeat
- c. do... while
- d. while

30. Combien de fois le message "Bonjour" sera-t-il affiché ?

```
int compteur = 15;
do
{
    printf("Bonjour\n");
    compteur++;
} while (compteur < 15);
```

- a. 0
- b. 14
- c. 1
- d. 15

Partie 2 : Modélisation & Conceptualisation

31. En UML, le diagramme qui présente les instances de classes utilisées dans le système est le diagramme :

- a. de classes
- b. de composants
- c. d'objets
- d. de déploiement

32. Un diagramme de séquence :

- a. est une représentation simplifiée d'un enchaînement de séquence se concentrant sur les échanges de messages entre les objets
- b. permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés
- c. est une représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs
- d. permet de décrire les variations séquentielles d'une donnée au cours du temps

33. Quel est l'intrus ne rentrant pas dans la modélisation d'un diagramme UML ?

- a. objet
- b. entité
- c. nœud
- d. paquetage

34. Quel est le nombre des diagrammes UML ?

- a. 13
- b. 15
- c. 10
- d. 4

35. Lequel de ces diagrammes n'est ni structurel ni statique ?

- a. diagramme de composants
- b. diagramme d'objets
- c. diagramme de structure composite
- d. diagramme d'activités

36. Le diagramme d'interaction :

- a. est une variante du diagramme structurel
- b. est une variante du diagramme d'activité
- c. englobe le diagramme d'interaction
- d. est un cas particulier du diagramme de séquence

37. Quelle affirmation est fausse ?

- a. le Stéréotype est une généralisation qui montre que l'objet est une variété d'un modèle
- b. le classeur permet de regrouper des unités ayant le même comportement ou structure
- c. chaque classe ou objet se définit précisément avec le signe « : »
- d. un paquetage regroupe des diagrammes ou des unités

38. Laquelle de ces vue ne fait pas partie d'UML ?

- a. vue de déploiement
- b. vue des fonctions
- c. vue logique
- d. vue des cas d'utilisation

39. Lequel de ces éléments de modélisation n'est pas de type relation ?

- a. association
- b. généralisation
- c. réalisation
- d. initialisation

40. Quel est le souci avec les outils logiciels de modélisation UML ?

- a. aucun d'entre eux ne respecte strictement aucune des versions d'UML
- b. il n'y a aucun outil logiciel de modélisation UML
- c. il n'y a aucun souci, plusieurs outils existent et sont conformes à la version UML 2
- d. leurs pictogrammes ne sont pas normalisés

Partie 3 : Bases de données relationnelles

41. Qu'est-ce qu'une base de données ?

- a. un logiciel
- b. une collection d'informations structurées
- c. une collection de documents reliés entre eux par des liens hypertextes
- d. un système assemblant des tables qui contiennent des cellules reliées par des formules

42. L'utilisation des systèmes de gestion de bases de données ne permet pas :

- a. l'utilisation d'un langage de requête déclaratif
- b. une description globale et unique des données dans une base de données
- c. l'indépendance totale par rapport au système d'exploitation car il en inclut un
- d. une sécurité d'accès aux données et résistance aux pannes en limitant les dégâts

43. Une contrainte d'intégrité est :

- a. une propriété qui permet de spécifier qu'un utilisateur est intègre
- b. une propriété à respecter par les données pour que la base de données soit cohérente
- c. une instruction qui permet d'intégrer des données dans une table de la base de données
- d. une condition qui permet d'effectuer une transaction

44. Une clé étrangère est un :

- a. champ de table spécial qui est présent dans toutes les tables d'une base de données
- b. ou plusieurs champs de table qui font référence à un ou plusieurs champs, clé primaire dans une autre table
- c. champ de table appartenant à une table d'une autre base de données
- d. champ de table ne portant pas le même nom que les champs des autres tables

45. Lequel de ces SGBD n'existe pas ?

- a. PostgreSQL
- b. MySQL
- c. MyTinySQL
- d. Microsoft SQL Server

46. Une transaction est :

- a. une unité d'exécution qui effectue une lecture/écriture sur une base de données en respectant certaines propriétés
- b. une commande de validation « commit »
- c. une opération commerciale ou boursière
- d. un accord conclu sur la base de concessions réciproques des parties

47. Quelle est le modèle de données inexistant ?

- a. hiérarchique
- b. relationnel
- c. parallèle
- d. réseau

48. Quelle étape ne fait partie du processus de conception d'une base de données ?

- a. le passage du schéma logique vers le schéma physique
- b. analyse des données et construction du dictionnaire
- c. mise en relation des entités et déduction du schéma
- d. structuration des données et détermination des entités et associations

49. Quel mot clé doit-on utiliser pour récupérer des enregistrements sur plusieurs tables ?

- a. multiset
- b. union
- c. join
- d. ALL

50. L'acronyme ACID associé aux transactions fait référence à :

- a. Atomicity, Consistency, Isolation, Durability
- b. Action, Commit, Integration, Data
- c. Association Composée et Intégrité de Données
- d. Aaron Coleman and Ivan Davis

2/ Exercice pratique : 2 heures 15

⇒ Dossier 1 - Langages & Développement

Question 1

Soit un salon de coiffure avec un coiffeur, un fauteuil de coiffure et un nombre de chaises Nbr_Chaise pour les clients qui attendent. En l'absence de clients, le coiffeur s'assoit sur le fauteuil et s'endort. Le premier client qui arrive réveille le coiffeur. Les clients suivants doivent s'asseoir sur une chaise (s'il y en a des libres) ou partir (si toutes les chaises sont occupées).

La solution au problème du coiffeur endormi suivante présente un problème.

Lequel ?

Expliquez clairement.

Déclarations :

SClient, SCoiffeur : sémaphores initialisés à 0.

mutex : sémaphore initialisé à 1.

Attente : entier initialisé à 0

<i>Processus Coiffeur</i>	<i>Processus Client</i>
Début	Début
Cycle	Wait(mutex)
Wait(SClient)	Si (Attente < Nbr_Chaise) alors
Wait(mutex)	Attente := Attente+1
Attente := Attente-1	Signal(mutex)
Signal(SCoiffeur)	Signal(SClient)
Signal(mutex)	Wait(SCoiffeur)
Couper_Cheveux	Obtenir_Coupe
FinCycle	Sinon Signal(mutex)
Fin	FinSi
	Fin

Question 2

On souhaite rechercher l'existence d'un mot (qu'on passera en argument au programme) dans tous les fichiers du répertoire courant.

Ecrivez en Perl le programme correspondant, qui affiche à la fin la liste des noms des fichiers qui contiennent le mot recherché ainsi que le nombre d'occurrences de ce mot et les numéros de lignes où il apparaît.

Le résultat est présenté par ordre décroissant du nombre d'occurrences.

Directives : utilisez un tableau associatif dont la clef est le nom du fichier et la valeur est une référence sur une liste contenant le nombre d'occurrences ainsi qu'une référence sur une liste contenant les numéros de lignes où le mot est présent.

Question 3

Que fait chacun des programmes suivants ?

```
1)    int main( )
      {
        int p=1 ;
        while(p>0) p=fork() ;
        execlp("prog", "prog", NULL) ;
        return 0 ;
      }
```

```

2)   int   i=2 ;
      int main ( )
      {
        j=10;
        int p ;
        while(i-- && p = fork())
          if(p<0) exit(1) ;
        j += 2;
        if (p == 0)
        {
          i *= 3;
          j *= 3;
        }
        else
        {
          i *= 2;
          j *= 2;
        }
        {
          printf(« i=%d, j=%d », i,j) ;
          return 0 ;
        }
      }

```

```

3) #include <stdio.h>
    #include <unistd.h>
    int main ( )
    {
      int fd[2], i=2;
      char ch[100];
      while ( i)
      {
        pipe(fd);
        if( fork())
        {
          close(fd[0]);
          dup2(fd[1],1);
          close(fd[1]);
          break;
        } else
        {
          close(fd[1]);
          dup2(fd[0],0);
          close(fd[0]);
        }
        i--;
      }
      scanf("%s", ch);
      printf("%s\n",ch);
      exit(0);
    }

```

```

4)   int i=4, j=10;
      int main ( )
      {
        int p ;
        p = fork();
        if(p<0) exit(1) ;
        j += 2;
        if (p == 0)
        {
          i *= 3;
          j *= 3;
        }
        else

```

```

    {
        i *= 2;
        j *= 2;
    }
    printf("i=%d, j=%d", i, j);
    return 0;
}

```

5) `int main ()`
`{ int p=1 ;`
`for(int i=0 ; i<=4 ; i++)`
`if (p>0) p=fork() ;`
`if(p !=-1) execlp("prog", "prog", NULL) ;`
`else exit(1) ;`
`while(wait(NULL) !=-1) ;`
`return 0 ;`
`}`

Question 4

Soit le fichier XML « famille_atome.xml » suivant :

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<classification_atomique>
  <famille type="gaz rare">
    <atome>
      <nom>hélium</nom>
      <symbole>He</symbole>
      <numero>2</numero>
      <masse>4</masse>
    </atome>
    <atome>
      <nom>néon</nom>
      <symbole>Ne</symbole>
      <numero>10</numero>
      <masse>20</masse>
    </atome>
    <atome>
      <nom>argon</nom>
      <symbole>Ar</symbole>
      <numero>18</numero>
      <masse>40</masse>
    </atome>
  </famille>
  <famille type="métal alcalin">
    <atome>
      <nom>lithium</nom>
      <symbole>Li</symbole>
      <numero>3</numero>
      <masse>7</masse>
    </atome>
    <atome>
      <nom>sodium</nom>
      <symbole>Na</symbole>
      <numero>11</numero>
      <masse>23</masse>
    </atome>
  </famille>
</classification_atomique>

```

```

</famille>
<famille type="métal alcalino-terreux">
  <atome>
    <nom>béryllium</nom>
    <symbole>Be</symbole>
    <numero>4</numero>
    <masse>9</masse>
  </atome>
  <atome>
    <nom>magnésium</nom>
    <symbole>Mg</symbole>
    <numero>12</numero>
    <masse>24</masse>
  </atome>
</famille>
<famille type="non métal">
  <atome>
    <nom>bore</nom>
    <symbole>B</symbole>
    <numero>5</numero>
    <masse>11</masse>
  </atome>
  <atome>
    <nom>carbone</nom>
    <symbole>C</symbole>
    <numero>6</numero>
    <masse>12</masse>
  </atome>
  <atome>
    <nom>azote</nom>
    <symbole>N</symbole>
    <numero>7</numero>
    <masse>14</masse>
  </atome>
  <atome>
    <nom>oxygène</nom>
    <symbole>O</symbole>
    <numero>8</numero>
    <masse>16</masse>
  </atome>
  <atome>
    <nom>fluor</nom>
    <symbole>F</symbole>
    <numero>9</numero>
    <masse>19</masse>
  </atome>
  <atome>
    <nom>silicium</nom>
    <symbole>Si</symbole>
    <numero>14</numero>
    <masse>28</masse>
  </atome>
</famille>
</classification atomique>

```

Que donne l'exécution du code PHP suivant ?

```
<?php
$document = simplexml_load_file('famille_atome.xml');
foreach ($document->famille as $famille)
    foreach ($famille->atome as $atome)
        echo "symbole : {"$atome->symbole} nom : {"$atome->nom}<br>";
?>
```

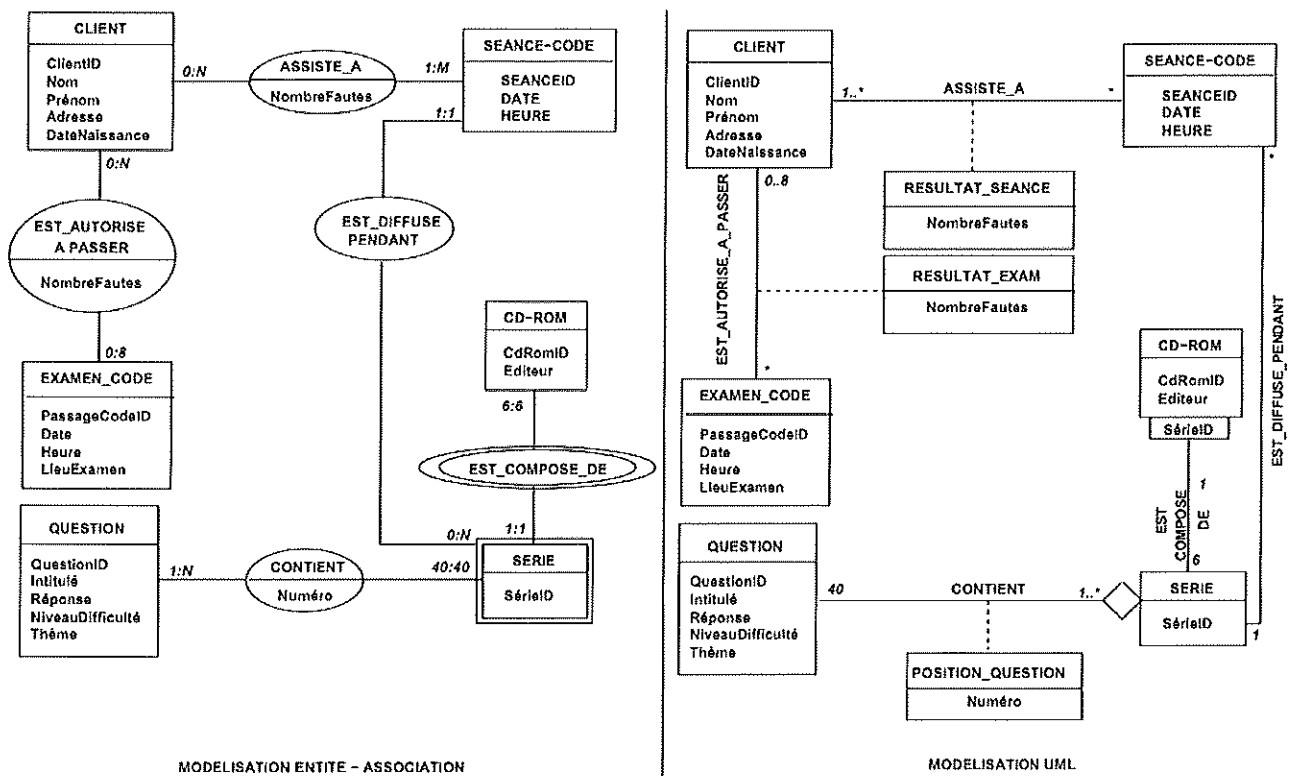
Ecrivez un code PHP qui affiche les Types de familles existant dans le fichier XML « famille_atome.xml ».

Ecrivez un code PHP qui apporte certaines modifications sur le fichier « famille_atome.xml » :

- supprimer l'attribut « type » de l'élément « famille »,
- modifier le contenu de l'élément « atome » de façon à ce que la valeur de l'élément « Nom » contienne : la valeur de l'élément « nom » + la valeur de l'élément « symbole » entre parenthèse : <nom> Nom_atome (symbole_atome) </nom> ,
- supprimer les éléments « symbole », « numero » et « masse » et leurs contenus.

⇒ Dossier 2 - Modélisation, Conceptualisation & Bases de données relationnelles

Question 1



Une auto-école souhaite construire une base de données pour gérer les examens théoriques du code de la route de ses élèves. Chaque élève est identifié par un numéro unique et est caractérisé par un nom, un prénom, une adresse et une date de naissance. Chaque élève assiste à plusieurs séances de code (autant qu'il le souhaite). Chaque séance est caractérisée par une date et une heure.

A chaque séance de code, le directeur de l'auto-école choisit une série de questions sur un CD-ROM. Chaque CD-ROM est identifié par un numéro et caractérisé par un nom d'éditeur. Chaque CD-ROM est composé de 6 séries, numérotées de 1 à 6. Chaque série est composée de 40 questions.

Chaque question est identifiée par un intitulé et caractérisée par une réponse, un niveau de difficulté et un thème. Une même question peut apparaître dans plusieurs séries avec un numéro d'ordre pour chaque série. Par exemple une même question peut apparaître comme question n°2 dans la série 5 du CD-ROM 15 et comme question n°12 dans la série 3 du CD-ROM 4. Une même série peut être projetée plusieurs fois à des séances différentes. Lorsqu'un élève assiste à une séance, il obtient le nombre de fautes (une note sur 40) qu'il a fait pour la série passée pendant la séance.

Lorsqu'un élève a obtenu, au cours des quatre dernières séances auxquelles il a assistées, un nombre de fautes inférieur ou égal à 5, le directeur de l'auto-école l'autorise à passer l'examen théorique du code de la route à une date donnée (un seul examen pour une date donnée). L'auto-école ne peut présenter que 8 élèves maximum à chaque date d'examen. Les élèves ayant obtenu plus de 5 fautes à l'examen sont recalés et doivent assister de nouveau à des séances de code avant de pouvoir se représenter à l'examen.

La base de données doit permettre de répondre à des requêtes telles que "Quel est le nombre moyen de fautes pour la série 5 du CD-ROM 14?", "Quels élèves peuvent se présenter au prochain examen du code de la route ?", "Quels élèves ont échoué au moins une fois à l'examen ? "...

La figure précédente présente la modélisation Entité/Association (format Merise) et la modélisation UML de l'énoncé. Les explications ne sont données que pour le schéma E/A mais peuvent être adaptées au schéma UML. Pour une compréhension de la différence entre une modélisation E/A ou UML et le passage au relationnel.

Les schémas de modélisation sont sémantiquement clairs. Néanmoins, quelques points nécessitent d'être précisés :

- l'ensemble d'entités Série est un ensemble d'entités faibles de CD-ROM, au format Merise (ou une association qualifiée en UML). En effet, ce choix de modélisation a été fait pour représenter le fait que le numéro d'une série est relatif au CD-ROM auquel la série appartient,
- les cardinalités de l'association entre les ensembles d'entités Série et CD-ROM sont 1:1-6:6, car une série appartient à un CD-ROM unique et un CD-ROM contient exactement 6 séries de questions. Le principe est le même pour les cardinalités de l'association entre Série et Question : une série contient exactement 40 questions (cardinalité 40 : 40). En revanche, une même question peut apparaître dans plusieurs séries avec un numéro d'ordre différent à chaque fois, d'où la cardinalité 1 : N et l'attribut Numéro qui caractérise l'association contient,
- l'attribut NombreFautes est un attribut de l'association entre les ensembles d'entités Client et Examen_Code et de l'association entre les ensembles d'entités Client et Séance_Code. En effet, cet attribut caractérise l'association et non pas un client, une séance de code ou encore un examen de code. Il caractérise le lien entre deux entités de ces ensembles.

Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence.

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante : $Nom(att_1, \dots, att_n)$: où « Nom » est le nom de la relation et « att_1, \dots, att_n » sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation dans la figure précédente.

Vous donnerez des explications claires et concises du passage au relationnel.

Vous préciserez notamment pourquoi et comment vous créez ou modifiez certaines relations (1 ligne maximum par relation).

Question 2

Soit le schéma de base de données relationnel suivant :

AGENCE (Num_Agence, Nom, Ville, Actif)

CLIENT (Num_Client, Nom, Ville)

COMPTE (Num_Compte, Num_Agence, Num_Client, Solde)

EMPRUNT (Num_Emprunt, Num_Agence, Num_Client, Montant)

Ecrivez les requêtes suivantes en SQL :

- 1) Clients ayant un compte et un emprunt à "La Rochelle"
- 2) Clients ayant un compte à l'agence "Paris-Etoile" et nom de la ville où ils habitent
- 3) Emprunteurs de l'agence "Paris-Rambuteau" classés par ordre alphabétique
- 4) Nombre de clients de l'agence "Paris-Bastille" n'ayant pas leur adresse dans la relation CLIENT
- 5) Supprimez de la relation AGENCE toutes les agences sans client
- 6) Clients ayant un compte dans une agence où "Claude" a un compte
- 7) Clients ayant un compte dans au-moins une agence d'"Orsay"