



CONCOURS DE PROMOTION INTERNE AU TITRE DE L'ANNEE 2015		
Spécialité	:	DEVELOPPEMENT INFORMATIQUE
Catégorie	:	16-18
Durée	:	2 heures

Exercice 1 : (4 points)

- 1.1 Ecrire (en JavaScript) une fonction $f1()$ qui va procéder à une permutation circulaire des url des liens contenus sur la page en cours. (2 points)
- 1.2. Ecrire (en JavaScript) une fonction $f2()$ qui déclenche automatiquement $f1$ toutes les 20 secondes, dès le chargement de la page. (1 point)
- 1.3. Etant donnée le fragment de code HTML suivant (1 point)

```
<div class="poésie">
<h1>La légende des siècles </h1>
<p>un recueil de Victor Hugo.</p>
<p>Paru en 1883</p>
</div>
<div>
<h1>Le langage HTML</h1>
<p>une spécification du W3C</p>
</div>
```

Proposez les règles CSS pour que :

- Le paragraphe qui suit les titres soit en gras.
- Les titres des livres de poésie soient en italique.

Exercice 2 : (4 points)

Soit le schéma relationnel suivant :

```
Salle (NomS, Horaire, capacité, Titre*)
Film (Titre, réalisateur, durée, acteur)
Producteur (Producteur, Titre*)
Vu (Spectateur, Titre*)
Aime (Amateur, Titre*)
```

N.B : Les champs soulignés sont des clés primaires, ceux marqués par * sont des clés étrangères.

Sachant que :

- Un film peut être produit par plusieurs producteurs.
- La relation Vu décrit les titres des films vus par les spectateurs. Evidemment un spectateur peut voir plusieurs films.
- La relation Aime décrit les films appréciés par les spectateurs (amateurs). Evidemment, un spectateur peut aimer plusieurs films.

- 2.1. Quels sont les acteurs qui jouent dans tous les films ? (1 point)
- 2.2. Quels sont les spectateurs qui aiment tous les films qu'ils voient ? (1 point)
- 2.3. Ecrire un bloc (PL/SQL ou Transact-SQL) permettant d'afficher le nombre de films réalisés par chaque réalisateur sous la forme (Le réalisateur : Youssef CHAHINE a réalisé 35 films). Traiter le cas où la table Film est vide comme une Exception. Dans ce cas afficher un message de la forme « Pas de films disponibles !! » (2 points)

Exercice 3 : (6 points)

3.1. Dans un établissement scolaire, on désire gérer la réservation des salles de cours ainsi que du matériel pédagogique (ordinateur portable ou/et Vidéo projecteur). Seuls les enseignants sont habilités à effectuer des réservations (sous réserve de disponibilité de la salle ou du matériel). Le planning des salles peut quant à lui être consulté par tout le monde (enseignants et étudiants). Par contre, le récapitulatif horaire par enseignant (calculé à partir du planning des salles) ne peut être consulté que par les enseignants. Enfin, il existe pour chaque formation un enseignant responsable qui seul peut éditer le récapitulatif horaire pour l'ensemble de la formation. (3 points)

Question : Donner le diagramme de cas d'utilisation correspondant.

3.2. Le déroulement normal d'utilisation d'une caisse de supermarché est le suivant :

- Un client arrive à la caisse avec ses articles à payer
- Le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si elle est supérieure à 1.
- La caisse affiche le prix de chaque article et son libellé.
- Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente.
- La caisse affiche le total des achats.
- Le caissier annonce au client le montant total à payer.
- Le client choisit son mode de paiement
 - liquide : le caissier encaisse l'argent, la caisse indique le montant à rendre au client
 - chèque : le caissier note le numéro de pièce d'identité du client
 - carte de crédit : la demande d'autorisation est envoyée avant la saisie
- La caisse enregistre la vente et l'imprime
- Le caissier donne le ticket de caisse au client

Question : Modéliser cette situation à l'aide d'un diagramme de séquence en ne prenant en compte que le cas du paiement en liquide. (3 points)

Exercice 4 : (6 points)

Vous pouvez Coder en Java ou C#

Le but de cet exercice est d'écrire une classe permettant de gérer les scores de joueurs pour un jeu donné. Les informations que l'on souhaite mémoriser pour un joueur, désigné par son nom, sont son meilleur score sur l'ensemble des parties, le nombre de parties jouées et la moyenne des scores de toutes les parties jouées.

Les jeux dont il est question se jouent seul et le joueur se voit délivrer en fin de partie un score (entier). Plus celui-ci est faible meilleur est le résultat du joueur. Le démineur est un exemple.

4.1. Les jeux vont être définis par une interface **Jeu**, qui ne possédera qu'une méthode appelée **joue** qui prend en paramètre un nom de joueur (une chaîne de caractères) et retourne le score du joueur pour une partie du jeu. Définissez le code de cette interface. (1 point)

4.2. La classe **Resultat** permet de gérer les informations pour un joueur. (2 points)

- Définir les attributs nécessaires de la classe **Resultat**.
- Définir un constructeur qui prend comme paramètre le premier score effectué par le joueur au jeu concerné et initialise les attributs.
- Définir la méthode **getMeilleur** qui retourne le meilleur score du joueur.
- Définir la méthode **public void nouveauScore(int score)** qui permet d'ajouter un nouveau score aux résultats d'un joueur.

4.3. La classe **GererResultats** permet de gérer les résultats pour l'ensemble des joueurs pour un jeu donné. Elle mémorise pour chaque joueur, désigné par son nom (une chaîne de caractères) un objet de la classe **Resultat**.

- Chaque instance de cette classe est définie par le jeu concerné et les résultats pour celui-ci. Faites les déclarations nécessaires ainsi qu'un constructeur, prenant comme paramètre un jeu, et initialisant les attributs définis. (1 point)
- Définissez une méthode **joueurJoue** qui prend en paramètre un nom de joueur, fait jouer une partie à celui-ci et met à jour ses résultats. (1 point)
- Définissez le code de la méthode dont la signature est :
public void meilleurMeilleur() et qui affiche le nom du (ou de l'un des) joueur ayant le meilleur de tous les scores pour le jeu. (1 point)