www.madness-madman.hooxs.com

OFPPT

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Examen de Passage
Session Juin 2008

Filière : TDI

Epreuve : Pratique

Niveau : Technicien Spécialisé

Durée : 4 h 30

Barème : 40 Pts

Variante 3

Important :

- ***Vous avez le choix d'utiliser un des langages suivants (C++, Pascal, JAVA, Vb en mode consol)***
- ***Le choix da la technologie utilisée dépend de votre spécialisation (JAVA ou .NET)***
- ***Le SGBDR utilisé est SQLSERVER ou ORACLE***
- ***Organiser vos travaux dans des dossiers portant le nom de chaque partie***
- ***choisir les structures de stockage de données***

Partie 1 : Programmation structurée : **(5 pts)**

Ecrivez un programme permettant, à l'utilisateur de saisir les notes d'une classe. Le programme, une fois la saisie terminée, affiche la liste des étudiants ayant une note supérieure à 10. La liste des étudiants doit être triée par ordre alphabétique

NB : Ne pas oublier des structures de contrôle pour la saisie des notes :
La note doit être numérique, comprise entre 0 et 20

Partie 2 : Programmation orientée objet : **(16 pts)**

Soit à développer une application de gestion des membres inscrits dans un Club sportif. Lorsqu'une personne est inscrite au Club, on lui affecte automatiquement un numéro et on fixe sa cotisation. La cotisation d'un membre peut changer en fonction de la situation de la personne.

1) Créer la classe ***Membre*** **(2 pts)**

2) Ajouter à la classe ***Membre*** les méthodes :

   - **toString()** : affichage des attributs de la classe ***Membre*** sous forme de chaîne de caractères **(1 pt)**
   - **modifie(double cotisation)** : modification de la cotisation **(1 pt)**

3) Ajouter un constructeur par défaut qui permet de créer un objet ***Membre*** dont le nom est "anonyme" **(1 pt)**

4) Ajouter un constructeur qui permet de créer un objet ***Membre*** en générant un numéro aléatoire **(1 pt)**

5) Ecrire le code permettant de saisir un **Membre** et prévoir les cas d'exception. **(2 pts)**

6) Ajouter une méthode de modification d'un **Membre (2 pts)**

7) Ajouter une méthode de recherche et d'affichage d'un **Membre (2 pts)**

8) Ajouter une méthode de suppression d'un **Membre (2 pts)**

9) Ajouter une méthode d'affichage de tous les **Membres**. **(2 pts)**

Partie 3 : Programmation Evénementielle : **(13 pts)**

Afin de faciliter la gestion de notre Club **(*Partie 2)***, on souhaite développer une petite application qui répond aux fonctionnalités désirées :

1. Créer une **Interface** de mise à jours des Membres
   a. Ajouter un Membre (Le contrôle de saisie est obligatoire, le membre ne doit pas être enregistré deux fois et le code Membre est généré automatiquement) **(2 pts)**
   b. Modifier un Membre **(2 pts)**
   c. Rechercher un Membre **(2 pts)**
   d. Supprimer un Membre **(2 pts)**
   e. Modifier la cotisation d'un Membre **(1 pt)**
   f. Afficher dans une liste tous les Membres **(2 pts)**
   g. Ajouter une méthode qui permet de trier la liste des Membres par code ou par nom **(2 pts)**

Partie 4 : SGBDR : **(6 pts)**

Soit le modèle relationnel suivant :

VEHICULE (**Immatriculation**, Catégorie)
CLIENT (**ClientCode**, ClientNom)
LOCATION (**LocationRef**, Immatriculation, ClientCode, MontantLocation)
BAREME (**Tranche**, MontantTranche, Points)

1 . Créer la base de données ***(2 Pts)***
2 . Créer les requêtes SQL :
   a. Le montant total des locations réalisées par le client ayant le code 12874. ***(1 Pt)***
   b. Le code et le nom du ou des client(s) ayant réalisé la location dont le montant est le plus élevé. ***(1 Pt)***
   c. Le code et le nom ainsi que le nombre de locations des clients ayant effectué au moins dix locations. ***(1 Pt)***
   d. Le code et le nom des clients qui n'ont pas loué de véhicules de catégorie 'C'. ***(1 Pt)***