OFPPT

مكتب التكوين المهني وإنعاش الشغل

*Office de la Formation Professionnelle*
*et de la Promotion du Travail*

*Direction Recherche et Ingénierie de Formation*

**Examen de passage**

**Session Juin 2009**

**Filière : TDI**

*Epreuve : Pratique*

**Niveau : Technicien Spécialisé**

**Durée : 4 h 30**

**Barème : 20 Pts**

***Important : assurez-vous que tous les éléments de vos projets sont sauvegardés dans un dossier portant votre nom et prénom et le numéro de la variante de l'examen comme suit "NOM PRENOM Variante".***

## Variante n° 1

### PARTIE I : PROGRAMMATION STRUCTUREE EN C OU C++ (4 PTS)

**1.** Ecrivez un programme qui lit N nombres entiers et qui affiche leur somme, leur produit et leur moyenne. Choisissez un type approprié pour les valeurs à afficher. Le nombre N est à entrer au clavier. **(1 Pt)**

**2. m** personnes sont assises autour d'une table circulaire et entament le jeu suivant. Elles se fixent entre-elles un entier **k>0** et une personne **P1** par laquelle elles commenceront le jeu. On commence à compter de **1** à **k** en partant de **P1** et en tournant à droite. La **k$^{ième}$** personne quitte la table et on continue à compter de **1** à **k** à partir de la personne suivante encore à table. La personne gagnante est la dernière qui reste à table.

Analyser ce problème et faire un programme permettant de réaliser ce jeu. **(3 Pts)**

### PARTIE II : PROGRAMMATION ORIENTEE OBJET (8 PTS)

Soit les classes suivantes :

| Classes : | | | |
|---|---|---|---|
| **Personnel** | **Enseignant** | **Etudiant** | **Module** |
| Code<br>Nom<br>Prénom<br>Grade | Salaire_brut<br>NbreHeuresSup<br>Taux_horaire<br>Taux_IGR | CodeE<br>NomE<br>Spécialité<br>DateNaiss | CodeM<br>TitreM<br>MasseHoraire |



www.madness-madman.hooxs.com

V1

***Travail à faire :***

1. Créer les classes : **Personnel** et **Enseignant**, sachant que la classe Enseignant hérite de la classe Personnel. Ajouter des constructeurs et des propriétés sachant que le champ **Grade** doit être un entier compris entre 16 et 22 et que le **Taux_IGR** ne peut prendre que la valeur **0,35** ou **0,42 (1,5 Pts)**

2. Créer les classes **Etudiant** et **Module** avec éventuellement des constructeur et des propriétés **(1 Pt)**

3. Nous souhaitons mettre à jour les données relatives aux enseignants en utilisant le fichier **Enseignant.txt :**

   a. Créer la méthode **Ajouter**(...) qui permet d'ajouter un Enseignant **(0,75 Pt)**
   b. Créer la méthode **Modifier**(...) qui permet de modifier un Enseignant **(0,75 Pt)**
   c. Créer la méthode **Supprimer**(...) qui permet de supprimer un Enseignant **(0,75 Pt)**
   d. Créer la méthode **Affichage**(...) qui permet d'afficher tous les Enseignant **(0,75 Pt)**

4. Ajouter une méthode permettant de calculer le salaire Net à payer pour un Enseignant donnée, sachant que : **(1,5 Pts)**

   ***SalaireNet = [SalaireBrut + (NbreheuresSup * Taux_horaire)] * (1 - Taux_IGR)***

5. Ajouter une méthode permettant de calculer la masse Salariale brute et Net de tous les enseignants **(1 Pt)**

## *PARTIE III : PROGRAMMATION EVENEMENTIELLE (8 PTS)*

Dans cette partie, nous allons utiliser les classes développées dans la Partie 2 en prévoyant ***l'enregistrement des données en mémoire***.

1. Créer un formulaire pour mettre à jour les données relatives à un enseignant
   - *a.* Ajout avec contrôle de saisie **(1 Pt)**
   - *b.* Suppression avec confirmation **(1 Pt)**
   - *c.* Modification avec confirmation **(1 Pt)**
   - *d.* Recherche par code ou par Nom de l'enseignant **(1 Pt)**
2. Ajouter dans le même formulaire, **une grille ou une Liste** permettant de lister tous les enseignants triés par ordre croissant **(1 Pt)**
3. Ajouter un **Label** permettant d'afficher le Salaire Net à payer à un enseignant choisi depuis la grille ou la Liste **(1 Pt)**
4. Ajouter un **Label** permettant d'afficher la masse salariale net **(1 Pt)**
5. Ajouter un bouton **Exporter**, permettant d'enregistrer les données de la grille ou la liste dans un fichier texte **(1 Pt)**



V/

www.madness-madman.hooxs.com