



Examen de Passage Session Juin 2008

Filière : TDI

Epreuve : Pratique

Niveau : Technicien Spécialisé

Durée : 4 h 30

Barème : 40 Pts

Variante 6

Important :

- Vous avez le choix d'utiliser un des langages suivants (C++, Pascal, JAVA, Vb en mode consol)
- Le choix de 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 : Excel : (6 pts)

Le : 12/06/2007

cte1= 50,00 Dh

Mat	Prénom	Nom	Date Embauche	S.Brut	Ass. Maladie	Retraite	S.Base	I.G.R	Prime anc.	S.Net
A12	B	B	01/10/1987	3 250,00 Dh						
D18	M	M	15/02/1991	5 000,00 Dh						
E19	F	F	16/03/1986	10 000,00 Dh						
C15	Y	Y	15/08/1988	3 000,00 Dh						
D15	Z	Z	23/09/1987	6 500,00 Dh						
B16	T	T	12/03/2000	2 000,00 Dh						
D16	Q	Q	24/02/1991	9 000,00 Dh						
E16	A	A	12/03/2003	15 000,00 Dh						
E17	S	S	26/03/2004	14 000,00 Dh						
D26	K	K	25/06/1980	7 500,00 Dh						
A16	G	G	04/11/2005	2 500,00 Dh						
B23	U	U	10/05/1987	2 000,00 Dh						

I. Réaliser le document :

- 1- Saisie et mise en forme **(1 Pt)**
 - a. Police: Book Antiqua et taille:11 , style gras pour les titres
 - b. Bordure- WortArt-forme automatique : respecter le modèle
 - c. Date courante

II. Etablir les fonctions suivantes:

- 1- Assurance maladie est égale à 6% du salaire brute si celui si est supérieur à 3000,00Dh; sinon elle égale à zéro **(0,5 Pt)**
- 2- Retraite est égale à 6% du salaire brute si celui si est supérieur à 7000,00Dh; sinon elle égale à 3% du salaire brute si celui si est supérieur à 3000,00Dh; sinon elle égale à la constante cte1 cellule J3 **(1 Pt)**
- 3- Salaire de base est égale à Salaire Brute - (l'assurance maladie + retraite) **(1 Pt)**
- 4- Impôt générale sur revenu est calculé par la règle suivante: **(1 Pt)**

	IGR
Salaire de base > 5000,00 Dh	Salaire de base*40%
5000,00 Dh > Salaire de base > 3000,00 Dh	Salaire de base*28%
3000,00 Dh > Salaire de base > 2000,00 Dh	Salaire de base*20%
2000,00 Dh > Salaire de base > 1500,00 Dh	Salaire de base*10%
1500,00 Dh > Salaire de base	0,00 Dh

- 5- Prime d'ancienneté est calculé par la règle suivante: **(0,5 Pt)**

Ancienneté	Prime ancienneté
15 ans et plus	1 300,00 Dh
de 10 ans à 14 ans	950,00 Dh
moins de 10 ans	500,00 Dh

NB: le nombre d'années d'ancienneté est égale à la différence entre l'année courante et l'année d'embauche

- 6- Salaire net est égale salaire de base - IGR + prime d'ancienneté **(1 Pt)**

Partie 2 : Programmation orientée objet : (13,5 pts)

1. Créez une classe **Professeur** dont les attributs sont les suivants : **(2 Pts)**

- numSomme
- nom
- age
- adresse
- NP : qui permet de compter le nombre d'objet créé

2. Créez les méthodes d'accès à ces attributs Modificateurs et Accesseurs **(2 Pts)**

3. Créer un constructeur d'initialisation (**1 Pt**)
4. Proposer une solution permettant de :
 - a. Ajouter un **Professeur (1,5 Pts)**
 - b. Supprimer un professeur (le compteur doit être mis à jour) (**2 Pts**)
 - c. Affichage de la liste des professeurs créés (**1,5 Pts**)
5. Créez une autre classe **ProfesseurVacataire** qui hérite de la classe **Etudiant** avec un attribut supplémentaire :

- honoraire

les attributs et méthodes de la classe parent ne doivent pas être répétées dans la classe **ProfesseurVacataire** : on se contente d'ajouter les attributs et méthodes nécessaires en utilisant les méthodes et constructeurs de la classe parente. (**2 Pts**)

6. Créer un menu qui permet à l'utilisateur de choisir l'opération à effectuer (**1,5 Pts**)

Partie 3 : Programmation Événementielle : (16,5 pts)

On vous demande de développer une solution pour la gestion des notes d'une classe de 24 stagiaires :

1. Gestion de modules :

- a. Créer un formulaire d'ajout (**1 Pt**)
- b. Pour chaque module créé, le professeur peut définir les coefficients des évaluations : contrôle continu 1, contrôle continu 2 et EFM. Proposer une solution permettant d'attribuer les coefficients aux évaluations (**2 Pts**)
- c. Afficher dans une liste l'ensemble des modules avec les coefficients de chaque évaluation (**2 Pts**)

2. Gestion des stagiaires

- a. Créer un formulaire d'ajout (**1 Pt**)
- b. Ajouter une fonction de suppression avec un message de confirmation (**1,5 Pts**)
- c. Créer un formulaire de recherche d'un stagiaire (**2 Pts**)

3. Gestion de notes

- a. Créer un formulaire pour la gestion des notes, le professeur choisi le stagiaire depuis une liste et affecte une note comprise entre 0 et 10 pour chaque évaluation relative à un module (**2 Pts**)
- b. Créer un formulaire affichant dans une liste les stagiaires et leurs notes pour chaque module. Le professeur choisi tout d'abord le module puis la liste se charge avec les données demandées (**2 Pts**)
- c. Ajouter dans le même formulaire un bouton qui permet d'afficher dans la liste les stagiaires ayant une moyenne des évaluations supérieure ou égale à 5 (**3 Pts**)

Partie 4 : SGBDR : (4 pts)

Soit le modèle suivant :

TYPÉMÉDICAMENT(**TypeMédicament**, **LibelléTypeMédicament**)

MÉDICAMENT(**NumMédicament**, **NomMédicament**, **TypeMédicament#**, ...)

ÉQUIVALENT(**NumMédicament#**, **NumMédicamentGénérique#**)

- 1- Créer la base de données **(2 Pts)**
- 2- Créer la requête qui permet de lister le nom et le libellé du type des médicaments qui n'ont pas d'équivalent générique **(1 Pt)**
- 3- Afficher la liste des médicaments dont le type est BOB **(1 Pt)**