



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Examen de Passage

Session Juin 2007

Filière : TSDI

Epreuve : Pratique

Niveau : Technicien Spécialisé

Durée : 4 h 30

Barème : 40 Pts

Variante n° 10

Partie 1 – JAVA (20Pts)

En vue de la gestion d'une bibliothèque, on vous demande d'écrire une application pour traiter des *documents* de nature diverse : des *livres*, qui peuvent être des *romans* ou des *manuels*, des *revues*, des *dictionnaires*, etc.

Tous les documents ont un numéro d'enregistrement (un entier) et un titre (une chaîne de caractères). Les livres ont, en plus, un auteur (une chaîne) et un nombre de pages (un entier). Les romans ont éventuellement un prix littéraire (un entier, parmi : GONCOURT, MEDICIS, INTERALLIE, etc.), tandis que les manuels ont un niveau scolaire (entier). Les revues ont un mois et une année (des entiers) et les dictionnaires ont une langue (un entier, parmi ANGLAIS, ALLEMAND, ESPAGNOL, etc.).

Ces diverses sortes de choses doivent pouvoir être manipulées en tant que documents.

1. Définissez les classes **Document**, **Livre**, **Roman**, **Manuel**, **Revue** et **Dictionnaire**, entre lesquelles existeront les liens d'héritage que la description précédente suggère.

Dans chacune de ces classes définissez le constructeur qui prend autant d'arguments qu'il y a de variables d'instance et qui se limite à initialiser ces dernières avec les valeurs des arguments. Définissez également une méthode **toString** produisant une description sous forme de chaîne de caractères des instances de la classe. Enfin, si les variables d'instance ont été déclarées **private** (à

www.Dev-Informatique.Com

défaut d'autre information, c'est conseillé) définissez également des accesseurs publics permettant de consulter les valeurs de ces variables. (5 pts)

Écrivez une classe exécutable **TestDocuments** qui construit et affiche un assortiment de documents de classes différentes. (1 pt)

2. Une bibliothèque sera représentée par une collection *de documents* (travailler avec Vector ou hashtable). Définissez une classe **Bibliotheque**, avec les méthodes :

- **Bibliotheque(int capacité)** - constructeur qui crée une bibliothèque ayant la capacité (nombre maximum de documents) indiquée, (2 pts)
- **void afficherDocuments()** - affiche tous les ouvrages de la bibliothèque, (1 pt)
- **void afficherAuteurs()** - affiche la liste des auteurs de la bibliothèque (au besoin, utilisez l'opérateur **instanceof**), (2 pts)
- **boolean ajouter(Document doc)** - ajoute le document référencé par *doc* et renvoie **true** (**false** en cas d'échec), (2 pts)
- **Document document(int i)** - renvoie le *i*^{ème} document, (1 pt)
- **boolean supprimer(Document doc)** - supprime le document référencé par *doc* et renvoie **true** (**false** en cas d'échec) (2 pts)

3. Définissez, avec un effort minimal, une classe **Livrotheque** ayant les mêmes fonctionnalités que **Bibliotheque** mais entièrement constituée de livres. En optimisant dans la classe **Livrotheque** la méthode **afficherAuteurs**. (4 pts)

Partie 2 – VB.Net (12 Pts)

On souhaite maintenant re-développer quelques fonctionnalités de cette application (de la Partie JAVA) en Vb

1. Créer un formulaire permettant d'ajouter, modifier, supprimer et rechercher un document (1 Pt)
2. Programmer les fonctionnalités suivantes :
 - a. Ajout (1 Pt)
 - b. Modification (2 Pts)
 - c. Suppression (2 Pts)
 - d. Recherche (2 Pts)
3. Ajouter un bouton permettant d'afficher la liste des documents d'un auteur après avoir saisi le nom de l'auteur dans une zone de texte (2 Pts)
4. Ajouter dans un autre formulaire une liste affichant les auteurs et leurs documents ainsi que le nombre de livre pour chaque auteur (2 Pts)

Partie 3 – Sql Server (8 Pts)

Soit le modèle logique suivant :

Client(idClient, nom, adresse)

Commande(num_cmd, idClient, date_cmd)

Produit(num_prd, libelle, prix_unitaire, qte_stock)

Ligne commande(num_cmd, num_prd, qte_commandee)

- 1) - Créer la base de données avec le nom « **gestionProduit** » (2 pts)
 - Remplir les tables avec quelques données pour le test

- 2) Développer les requêtes suivantes :
 - a- afficher les informations suivantes (nom du client, date de commande) (1 pt)
 - b- afficher les information suivantes (libelle du produit, quantité totale commande) (1 pt)
 - c- afficher les clients qui ont commandé au moins un produit en avril 2006 (1 pt)
 - d- afficher pour chaque client le prix total de chaque commande (1 pt)
 - e- afficher le montant totale des ventes en 2006 (1 pt)
 - f- afficher le montant total des ventes en mois de Février 2006 (1 pt)