



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 2010

<http://ofppt-examen.blogspot.com>

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° 4

PARTIE I : PROGRAMMATION STRUCTUREE EN C OU C++ (5 PTS)

1. Ecrire un programme qui permet de comparer et de calculer la somme de deux nombres. Pour ce faire ce menu doit être affiché. **(1 Pt)**

Entrez le nombre correspondant à votre choix

1. Comparer deux entiers
2. Somme de deux entiers
3. Quitter

2. On modélise un nombre entier de la base octale par un tableau d'entiers de dimension 10. Chaque élément du tableau contient un chiffre de 0 à 7. Le chiffre du poids faible occupera la case d'indice 9 du tableau. **(2 Pts)**

Exemple : 3406

0	0	0	0	0	0	3	4	0	6
---	---	---	---	---	---	---	---	---	---

Le nombre est saisi en entrant successivement les chiffres du poids plus faible au poids plus fort, on marque la fin par un chiffre qui n'est pas compris entre 0 et 7.

3. Soit un tableau de 100 valeurs ordonnées de manière croissante. Ecrire un programme permettant de créer deux nouveaux tableaux ordonnés de manière croissante, contenant l'un les valeurs paires, l'autre les valeurs impaires. **(2 Pts)**

Exemple :

Tableau initial

2	4	10	13	15	21	28	55	44	77
---	---	----	----	----	----	----	----	----	----

Tableau des valeurs paires

2	4	10	28	44
---	---	----	----	----

Tableau des valeurs impaires

13	15	21	55	77
----	----	----	----	----

PARTIE II : PROGRAMMATION ORIENTEE OBJET EN VB.NET, C# OU JAVA (10 Pts)

On désire programmer une application de gestion de petit commerce. Pour cela, on va écrire une classe **Produit** et une classe **Magasin**.

Dans notre commerce, chaque produit est caractérisé par son nom, son prix d'achat, son prix de vente, le nombre d'exemplaires en stock, et sa description.

A la création du produit, on fixe son nom, son prix d'achat et son prix de vente, c'est-à-dire que ces données sont fournies en argument au constructeur de la classe. Par défaut, la description du produit est « Pas de description », et la quantité de produits en stock est nulle.

La classe Produit dispose d'un certain nombre de méthodes, qui lui permettent d'afficher la description du produit, de l'éditer, d'augmenter ou de diminuer le nombre d'exemplaires en stock, ainsi que d'obtenir les valeurs des différents attributs.

Ecrivez la classe Produit (1.5 Pts)

Un magasin se caractérise par son solde et par son stock de produits. Le stock de produit est représenté par un tableau ou un vecteur d'objets « Produit ».

Avant de pouvoir acheter ou vendre un produit, il faut l'avoir ajouté dans le stock. Pour cela, la classe Magasin doit disposer d'une méthode AjouterProduit

L'indice de la case du tableau (ou du vecteur) dans laquelle on a stocké le produit devient alors la référence de ce produit. Pour acheter ou vendre ce produit, on utilise alors la référence comme argument des méthodes:

```
acheterProduit(int referenceProduit,int nombreExemplaires)  
vendreProduit(int referenceProduit,int nombreExemplaires)
```

La classe Magasin dispose également des méthodes habituelles d'accès à ses attributs, et d'une méthode Bilan qui permet d'afficher un bilan du magasin.

Ecrivez la classe Magasin. Testez-la. (1.5 Pts)

Différents types de produits

Jusque là, les produits disposent d'une description qui est une simple chaîne de caractère. Cela peut être insuffisant dans certains cas.

Par exemple, on aimerait avoir une classe Livre qui ait le même comportement que la classe Produit, mais qui dispose d'un attribut auteur et d'un attribut éditeur.

De même, un cd se caractérise par son auteur, son interprète, et l'ensemble des titres de ses pistes.

Ecrivez la classe Livre qui hérite de la classe Produit. Certaines méthodes doivent être surchargées (c'est-à-dire réécrites) **(1 Pt)**

Ecrivez la classe Cd qui hérite de la classe Produit. **(1 Pt)**

Il va maintenant falloir faire les modifications nécessaires dans la classe Magasin, mais elles sont minimales. En effet, Livre étant une sous classe de Produit, les objets de type Livre sont également de type Produit, donc le tableau (ou le vecteur) de produits peut également contenir des livres ou des cds.

C'est à la création des objets que les choses changent. En plus de la méthode ajouterProduit, la classe Magasin doit également contenir une méthode ajouterLivre et une méthode ajouterCd.

Modifier la classe Magasin pour tenir compte des nouveaux objets. **(1.5 Pts)**

Pour utiliser pleinement votre classe Magasin, vous avez maintenant besoin, d'une méthode d'interaction permettant de proposer à l'utilisateur d'ajouter un livre, un cd ou un produit standard, d'acheter, de vendre, d'obtenir ou d'éditer la description d'un produit déjà existant, d'afficher un bilan général.

Ecrivez la méthode interaction de la classe Magasin **(1.5 Pts)**

Recherche d'un produit

Dans l'exercice précédent, pour acheter, vendre, éditer ou afficher la description d'un produit existant, il fallait le spécifier par sa référence, c'est-à-dire par son indice dans le tableau. Ce n'est pas un moyen très pratique d'aller chercher les produits, surtout si il y en a un très grand nombre.

Nous allons écrire les deux méthodes suivantes :

La méthode public Produit rechercherProduit(String nom) qui renvoie un objet Produit dont le nom est spécifié en argument.

la méthode public Produit rechercherProduitParMot(String mot) qui renvoie un objet Produit dont la description contient le mot spécifié en argument.

Pour écrire ces méthodes, nous allons avoir besoin d'un outil capable de comparer deux chaînes de caractères, et d'un outil capable de dire si une chaîne est incluse dans une autre. C'est la classe String elle-même qui est capable de faire cela. En effet, String est une classe à part entière qui a ses attributs et ses méthodes.

Ecrivez la méthode rechercherProduit **(2Pts)**

PARTIE III : PROGRAMMATION EVENEMENTIELLE EN VB.NET, C# OU JAVA (5 Pts)

Nous allons maintenant nous intéresser à la création d'interfaces qui vont nous permettre d'interagir et de communiquer avec notre application en mode graphique.

1. Créer une interface permettant la mise à jour d'un produit.
 - a. Ajout d'un produit **(0.5 Pt)**
 - b. Suppression d'un produit **(0.5 Pt)**
 - c. Modification d'un produit **(0.5 Pt)**
 - d. Rechercher un produit **(0.5 Pt)**
 - e. Afficher la liste des produits **(0.5 Pt)**
 - f. Afficher le nombre de produits dans le stock **(0.5 Pt)**
2. Construisez une interface permettant de créer un magasin et de lui affecter un stock de produit. Lors de l'affectation, le système propose à l'utilisateur les différentes possibilités (produit standard, livre, cd) **(1 Pt)**
3. Créer une interface permettant de rechercher un produit **(1 Pt)**