



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° 5

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

1. Dans un magasin les caissières saisissent pour chaque produit :

- Le prix unitaire
- La quantité achetée
- La catégorie du produit

Il y a une taxe pour chaque catégorie du produit

Catégorie	Taxe
1	6%
2	9%
4	15%
5	20%

Ecrire un programme qui permet d'afficher le prix à payer pour un client **(2 Pts)**

2. Ecrire un programme qui permet d'afficher tous les nombres abondants inférieurs à 50. Un nombre abondant est un nombre entier naturel n qui est strictement inférieur à la somme de ses diviseurs stricts. **(2 Pts)**

Exemple :

12 est abondant car $1+2+3+4+6 > 12$

3. Un palindrome est une chaîne de caractères qui se lit de gauche à droite ou de droite à gauche et donne la même résultat.

Exemple : « radar », « Laval ».

Ecrire un programme qui permet de tester si un mot est un palindrome ou non. **(1 Pt)**

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

Nous allons développer une application pour gérer les différents types de produits dans un magasin. Pour cela, on va écrire une classe **Article** et une classe **Magasin**.

Dans notre commerce, chaque Article 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 d'un article, 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 de l'article est « Pas de description », et la quantité de Articles en stock est nulle.

La classe Article dispose d'un certain nombre de méthodes, qui lui permettent d'afficher la description du Article, 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 Article (1.5 Pts)

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

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

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

```
acheterArticle(int referenceArticle,int nombreExemplaires)
vendreArticle(int referenceArticle,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 Articles

Jusque là, les Articles 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 Article, 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 Article. Certaines méthodes doivent être surchargées (c'est-à-dire réécrites) **(1 Pt)**

Ecrivez la classe Cd qui hérite de la classe Article. **(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 Article, les objets de type Livre sont également de type Article, donc le tableau (ou le vecteur) de Articles peut également contenir des livres ou des cds.

C'est à la création des objets que les choses changent. En plus de la méthode ajouterArticle, 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 Article standard, d'acheter, de vendre, d'obtenir ou d'éditer la description d'un Article déjà existant, d'afficher un bilan général.

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

Recherche d'un Article

Dans l'exercice précédent, pour acheter, vendre, éditer ou afficher la description d'un Article 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 Articles, surtout si il y en a un très grand nombre.

Nous allons écrire les deux méthodes suivantes :

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

la méthode public Article rechercherArticleParMot(String mot) qui renvoie un objet Article 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 rechercherArticle **(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 Article.
 - a. Ajout d'un Article **(0.5 Pt)**
 - b. Suppression d'un Article **(0.5 Pt)**
 - c. Modification d'un Article **(0.5 Pt)**
 - d. Rechercher un Article **(0.5 Pt)**
 - e. Afficher la liste des Articles **(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 Article. Lors de l'affectation, le système propose à l'utilisateur les différentes possibilités (Article standard, livre, cd) **(1 Pt)**

3. Créer une interface permettant de rechercher un Article **(1 Pt)**

- 4.