



Office de la Formation Professionnelle
et de la Promotion du Travail
Direction Recherche et Ingénierie de la Formation

Examen de Passage, session Juillet 2012

Filière : **Techniques de Développement Informatique**
Niveau : TS
Durée : 4h30mn

Epreuve : Pratique – V2-1
Barème : /20 pts

Variante 4

<http://www.ista-ntic.net/>

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**".

Dossier 1 : (1,5 pts)

- Ecrire un programme dans un langage structuré permettant d'afficher tous les nombres compris entre 100 et 2000 répondants au critère suivant : il y a au moins un chiffre de chacun de ces nombres qui est répété **exactement deux fois**. Exemples de tels nombres : 101 (le chiffre 1 est répété exactement deux fois), 1505 (le chiffre 5 est répété deux fois), 1771 (le chiffre 1 est répété deux fois ou le chiffre 7 est répété deux fois). Mais 1101 ne fait pas partie de ces nombres car le chiffre 1 est répété trois fois et le chiffre 0 une fois, et ainsi aucun des chiffres n'est répété exactement deux fois. A l'exécution, les 5 premiers nombres affichés sont : 100, 101, 110, 112, 113... (1 pt)
- Modifier le programme précédent pour avoir les effets suivants :
Afficher un seul nombre par ligne et chaque fois que 20 de ces nombres sont affichés, le programme affichera le message suivant « Tapez ENTREE pour continuer ». Quand l'utilisateur tape la touche « ENTREE », l'écran s'efface et on obtient une autre partie de ces nombres (s'il en reste). Le dit message s'affiche encore quand tous ces nombres ont été affichés. (0,5 pt)

Dossier 2 : (15,75 pts)

L'objet de cette application est la gestion informatisée des commandes de clients, Elle doit permettre de mettre à jour la quantité stock de chaque produit commandé.

Recommandation : utiliser des listes génériques (List<T> : listes typées) de préférence. Mais vous pouvez utiliser des ArrayList ou des Collections.

1- Classe Client.

- Définir une classe Client dont les caractéristiques sont : Code client, Nom, Adresse, Tél, E-mail. (0,5 pt)
- Ecrire un constructeur à deux paramètres : Code client, Nom. (0,5 pt)
- Ecrire un deuxième constructeur avec tous les paramètres. (0,5 pt)
- Ecrire les accesseurs des champs et la méthode toString() qui renverra le Nom et l'adresse du client séparés par un retour à la ligne. (0,75 pt)

<http://www.ista-ntic.net/>

2- Classe Produit.

- Définir une classe Produit dont les caractéristiques sont Référence, Désignation, Prix unitaire, Quantité stock. (0,5 pt)
- Choisir des types adaptés aux champs. (0,25 pt)
- Écrire un constructeur avec tous les paramètres. (0,5 pt)
- Écrire les accesseurs des champs. (0,5 pt)
- Écrire la méthode **toString()** qui renverra les informations de ce produit séparées par tabulation mais sans la quantité stock. (0,5 pt)

<http://www.ista-ntic.net/>

3- Classe Commande.

- Définir une classe Commande dont les caractéristiques sont : (1 pt)
 - Numéro commande : le numéro de la 1^{ère} commande créée est 1001 et à chaque création d'une nouvelle commande, ce numéro doit être automatiquement incrémenté de 1. Faire le nécessaire (données + code) pour y arriver.
 - Cclient : c'est le client qui a passé cette commande.
 - Date Commande : date où Cclient a passé cette commande.
 - une liste des produits commandés et une liste des quantités commandées : ces deux listes doivent toujours avoir le même nombre d'éléments : à chaque produit commandé correspond une quantité commandée dans la 2^{ème} liste (même indice).
- Ecrire un constructeur sans aucun paramètre permettant de définir la valeur du numéro commande et d'affecter la date système à Date commande. (0,75 pt)
- Écrire les accesseurs des champs et la méthode **toString()** qui renverra le numéro commande, la date commande et le nom client séparés par tabulation. (0,5 pt)
- Ecrire la méthode **TotalCommande()** qui retourne le coût global de la commande = Somme des (Prix unitaire * Quantité commandée) pour l'instance en cours. (0,5 pt)
- Ecrire la méthode **AjouterProduit()** ayant comme paramètres le produit à commander et la quantité commandée correspondante. Elle doit permettre la mise à jour de la liste des produits commandés, la liste des quantités commandées et la quantité stock de ce produit ajouté (Nouvelle quantité stock = Ancienne quantité stock - quantité commandée). (1 pt)
- Ecrire une méthode **AfficherCommande()** qui permet d'imprimer sur l'écran les détails d'une commande de la manière suivante : (1 pt)

Date :/../....			
N° Commande :			
Nom client :			
Adresse :			
Réf.	Désignation	Prix unitaire	Qté. Commandée
.....			
Montant			
.....			

Indication : Montant est une donnée calculée = Prix unitaire * Qté. Commandée

4- Programme principal

- Déclarer une liste de clients, une liste de produits et une liste de commandes accessibles par toutes les méthodes du programme principal. (0,5 pt)
- Ajouter une méthode **rechercherClient()** qui recherche un client par son code. Si trouvé, elle retourne le client correspondant dans la liste des clients. Sinon, elle retourne la valeur **null**. (0,5 pt)
- Ajouter une méthode **rechercherProduit()** qui recherche un produit par sa désignation. Si trouvée, elle retourne le produit correspondant dans la liste des produits. Sinon, elle retourne la valeur **null**. (0,5 pt)
- Ajouter une méthode **rechercherCommande()** qui recherche une commande par son numéro. Si trouvé, elle retourne l'indice correspondant dans la liste des commandes. Sinon, elle retourne la valeur -1. (0,5 pt)
- Ajouter par code (en mode conception) trois clients dans la liste des clients. (0,5 pt)

<http://www.ista-ntic.net/> 2/4

- f. Ajouter par code trois produits dans la liste des produits. (0,5 pt)
- g. Ajouter par code deux commandes dans la liste des commandes. La 1^{ère} contenant un produit et la 2^{ème} deux produits. Utiliser certains constructeurs et certaines méthodes déjà conçus. (0,5 pt)
- h. Concevoir le menu suivant : (en tenant compte des indications ci-dessous)
 - 1- Ajouter client. (0,5 pt)
 - 2- Ajouter produit. (0,25 pt)
 - 3- Ajouter commande. (0,75 pt)
 - 4- Rechercher commande par numéro commande. (0,5 pt)
 - 5- Rechercher les commandes passées à une date. (0,75 pt)
 - 6- Fin. (0,25 pt)

<http://www.ista-ntic.net/>

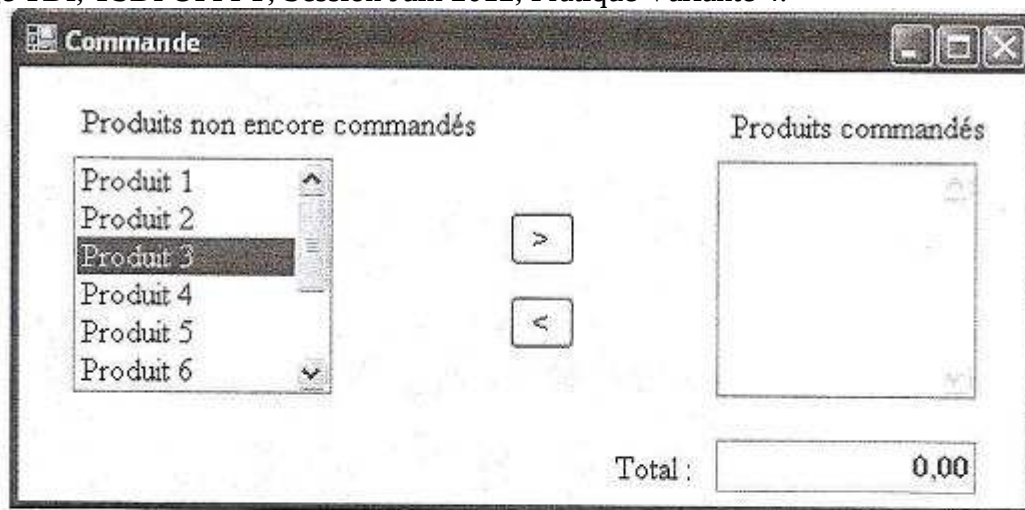
Indications concernant les traitements à réaliser dans le menu ci-dessus :

- Ajouter commande :
 - Le numéro de la nouvelle commande est automatiquement affecté par programme et la date commande correspond à la date système.
 - Saisir le code du client qui doit correspondre à un client de la liste des clients et si trouvé, le client ainsi trouvé correspond à Cclient de la commande à ajouter, sinon le programme vous demandera de saisir un autre code client.
 - Pour chaque produit à commander, l'utilisateur saisira sa désignation à chercher dans la liste des produits (même principe que celui du code client). Si trouvée, le produit ainsi trouvé sera ajouté à la liste des produits de la commande en cours.
 - Chaque fois que la désignation saisie est trouvée, saisir la quantité commandée correspondante au produit trouvé. Cette dernière sera soustraite de la quantité stock de ce produit.
 - Après la saisie de la quantité commandée, le programme demandera à l'utilisateur s'il souhaite saisir un autre produit pour la commande en cours.
- Rechercher commande par numéro commande : saisir le numéro de commande. Si trouvé, imprimer la commande comme précisé dans la question 3.f (méthode AfficherCommande()). Sinon, afficher un message adéquat pour avertir l'utilisateur de l'inexistence de cette commande.
- Rechercher les commandes passées à une date : Imprimer la liste des commandes (Numéro commande, Date commande, Nom client) dont la date commande est égale à la date saisie. Si aucune commande ne correspond à cette date saisie, afficher un message adéquat.

Dossier 3 : (2,75 pts)

Dans cette partie, vous ferez appel uniquement à la classe Produit du dossier 2 (ni Client ni Commande). Soit l'interface suivante de commande :

<http://www.ista-ntic.net/>



<http://www.ista-ntic.net/>

- 1- Réaliser l'interface Commande ci-dessus. (0,5 pt)
- 2- Ecrire une méthode possédant 2 paramètres de type ListBox, permettant d'ajouter l'élément sélectionné de la première liste à la fin de la deuxième et de supprimer cet élément de la première. (0,5pt)
- 3- Ecrire la procédure événementielle pour qu'au démarrage de l'application, la 1ère ListBox de l'interface Commande ci-dessus, soit chargée par les noms des produits contenus dans une liste de produits et dont les éléments sont eux aussi ajoutés automatiquement par code. (0,75 pt)
- 4- Ecrire le code du 1er bouton (>) : Si un produit de la 1ère ListBox est sélectionné, il sera ajouté à la 2ème et son prix unitaire sera ajouté au total. Sinon un message adéquat est affiché. Utiliser une liste de produits pour contenir les produits relatifs à la 2ème ListBox. (0,5 pt)
- 5- Ecrire le code du 2ème bouton (<) : Si un produit de la 2ème ListBox est sélectionné, il sera ajouté à la 1ère et son prix unitaire sera soustrait du total. Sinon un message adéquat est affiché. (0,5 pt)

<http://www.ista-ntic.net/>