Durée : 1h30

Examen du module POO

Exercice 1 (15 pts)

Partie I

Dans cet exercice, on s'intéresse à la gestion des lignes et des moyens de transport en commun dans une ville donnée. Un moyen de transport peut être de trois types : *Bus*, *Tramway* ou *Métro*.

1. Décrire la classe **TypeTransp** en utilisant une énumération.

Sur une ligne de transport, on trouve plusieurs stations (les arrêts). Une station est décrite par un *numéro* entier, un *nom* de type chaine de caractères, un *type* (station principale ou secondaire), le *nombre* de types de transport (Bus, Tramway, Métro) disponibles dans la station et un *tableau de TypeTransp*.

2. Donner l'implémentation de la classe **Station** comportant un **constructeur**, une méthode Lire pour la saisie au clavier, une méthode **Afficher** et une méthode **Modifier** qui modifie le type de la station.

Une ligne de transport est décrite par un *numéro* entier (à déclarer *private*) , un *point de départ* et un *point d'arrivée* de type chaine de caractères (par exemple : Alger - DarElBeida), un *seul type de transport*, un *nombre* de stations et l'*ensemble des stations*.

3. Donner l'implémentation d'une classe **Ligne** comportant un **constructeur**, une méthode **Afficher,** une méthode **Lire**, une méthode **Ajouter** pour ajouter une station **St** sur la ligne après une station de numéro ***n*** donné et les **accesseurs** nécessaires.

4. Ecrire un programme qui
   - Crée un ensemble de lignes (le nombre n'est pas connu), l'arrêt de la saisie se fera dès qu'on arrive à un numéro **numLigne** donné.
   - Affiche les numéros de lignes desservies par le même moyen de transport M donné.

Partie II

On voudrait améliorer la qualité de service aux passagers en affichant des horaires de passage des bus, tramway ou métro à chaque station, pour chaque ligne. Donc, on doit compléter la classe **Ligne** par un tableau comportant les horaires de passage du Bus, Tram ou Métro.

5. Donner l'implémentation d'une classe Horaire décrite par (*heure : minute*) comportant un **constructeur**, une méthode **Lire**, une méthode **Afficher**, une méthode **ChangerHoraire** qui change un horaire (h1, m1) par (h2, m2) donnés et une méthode **Egal** qui vérifie l'égalité entre deux horaires.

6. A partir de la classe Station, Implémenter une classe **StationIndiquée** comportant un *tableau d'horaires de* taille 20. Pour simplifier, on considère les stations desservies par un seul type

de transport, donc un seul tableau d'horaires. La classe **StationIndiquée** doit implémenter un **constructeur**, une méthode **Lire**, une méthode **Afficher** et une méthode **ChangerHoraire (H1, H2)** sur le tableau d'horaires.

7. Ecrire un programme qui
   - Crée un vecteur de n objets de type **StationIndiquée** (appartenant à la même ligne).
   - Modifie l'horaire (h1 :m1) donné par un horaire (h2 :m2) donné dans les tableaux d'horaires.
   - Afficher les numéros de stations ainsi que les tableaux d'horaires modifiés.

Exercice 2 (5pts)

1. Définir une classe abstraite **Liste** qui est une *liste chainée* d'entiers comportant une méthode **Afficher**, une méthode abstraite **Ajouter** pour ajouter un élément et une méthode **Construire** (int n) pour construire une liste chainée de n éléments.
2. En utilisant la classe Liste, donner l'implémentation d'une classe **ListeFiFO** et d'une classe **ListeLIFO** d'entiers.

Indications

1. **Java.util.Scanner** : classe pour la lecture de données au clavier
2. **Java.util.Vector** : classe pour la création d'un vecteur de taille variable et illimitée
3. **add(object obj) :** pour ajouter un élément obj à une structure de type Vector
4. **elementAt (i)** : donne l'élément d'indice i dans l'objet de type Vector
5. **size()** : donne le nombre d'éléments d'un objet de type Vector

Bon Courage