ISIMM
Département informatique

A.U. 2021/2022

TP n°1 de Programmation Java

Enseignant : Dr ABBASSI Imed **Groupe : CPI 2**

Exercice 1 :

Un compte bancaire est identifié par un numéro de compte. Ce numéro de compte est un entier positif permettant de désigner et distinguer sans ambiguïté possible chaque compte géré. Lorsqu'un nouveau compte est créé, le numéro qui lui est attribué est n+1. Un compte est associé à une personne (civile ou morale) titulaire du compte. Le titulaire du compte est décrit par son nom. Il peut ajouter ou retirer un montant au solde du compte. Le solde d'un compte est un entier positif ou nul. Lors de l'opération de retrait, un compte ne peut être débité d'un montant supérieur à une valeur désignée sous le terme de débit maximal autorisé. Le solde initial d'un compte ne peut être inférieur à 20 TND.

1) Elaborez une spécification d'une classe Java **Compte** modélisant un compte bancaire.
2) Écrire la méthode **créditer** qui permet d'ajouter un montant positif au solde du compte.
3) Écrire la méthode **débiter** permettant de retirer un montant positif au solde du compte.
4) Écrire la méthode **virer** permettant de transférer un montant d'un compte vers un autre.
5) Écrirez un programme de test permettant de :

- Créer un compte c1, au nom de J. DUPONT avec un solde initial de 1000 TND et un montant maximal de retrait autorisé de 300 TND.
- Créer un compte c2, au nom de C. DURANT avec un solde initial de 10000 TND et un montant maximal de retrait autorisé de 500 TND.
- Retirer 300 TND du compte c1.
- Retirer 600 TND du compte c2.
- Déposer 500 TND sur le compte c1.
- D'afficher les caractéristiques des comptes c1 et c2.
- Virer 1000 TND du compte c2 vers le compte c1.
- D'afficher les caractéristiques des comptes c1 et c2.

Exercice 2 :

Réaliser une classe Vecteur permettant de manipuler des vecteurs ayant un nombre quelconque de composantes de type double. On y prévoira :

- un constructeur Vecteur (int n), n représentant le nombre de composantes qui seront alors initialisées à zéro,
- un constructeur Vecteur (int n, double x), n représentant le nombre de composantes qui seront alors toutes initialisées à la valeur x,
- un constructeur Vecteur (double [ ] v) qui créera un vecteur par recopie du tableau v,
- une méthode (non statique) prod_scal fournissant le produit scalaire de deux vecteurs (ici, si les deux vecteurs ne sont pas de même taille, on se contentera de fournir la valeur zéro),
- une méthode (statique) somme fournissant la somme de deux vecteurs ; s'ils n'ont pas la même taille, on renverra une référence "nulle",
- une méthode affiche affichant les composantes d'un vecteur.

Écrire un petit programme d'utilisation.

Exercice 3 :

Il existe une méthode de détermination de tous les nombres premiers compris entre 1 et n, connue sous le nom de "crible d'Eratosthène". Elle consiste à dresser une liste de tous les nombres entiers considérés et à y "rayer" tous les multiples d'autres entiers. Plus précisément, on procède ainsi :

- on raye le 1 (qui, par définition, n'est pas un nombre premier),
- on recherche, à partir du dernier nombre premier considéré (la première fois, on convient qu'il s'agit du 1), le premier nombre non rayé (on peut montrer qu'il est premier). Il devient, à son tour, le dernier nombre premier considéré et on raye tous ses multiples.
- On répète le traitement précédent jusqu'à ce que le nombre premier considéré soit supérieur à la racine carrée de n. On peut alors démontrer que tous les nombres non premiers ont été rayés de la liste.

Écrire un programme exploitant cette méthode pour rechercher tous les nombres premiers compris entre 1 et 100.

Exercice 4 :

Un répertoire téléphonique est caractérisé par sa capacité et un ensemble d'abonnés. La capacité est le nombre maximal d'abonnés que pourra contenir le répertoire. Un abonné dispose un nom et un numéro de téléphone.

1) Proposez une classe Java nommée Abonne pour gérer les abonnés tout en respectant la notion d'encapsulation de données.
2) Proposez une classe Java nommée Repertoire pour la manipulation des répertoires téléphoniques tout en respectant la notion d'encapsulation. Une telle classe devra disposer des fonctionnalités suivantes :
   - Un constructeur recevant un argument de type entier permettant la création d'un répertoire.
   - Une méthode addAbonne permettant d'ajouter un nouvel abonné.
   - Une méthode getNumero fournissant le numéro associé à un nom d'abonné fourni en argument.
   - Une méthode getNAbonnes qui fournit le nombre d'abonnés figurant dans le répertoire.
   - Une méthode getAbonne fournissant l'abonné dont le rang est fourni en argument.
   - Une méthode getAbonnesTries fournissant un tableau des références des différents abonnés, rangés par ordre alphabétique (pour simplifier, on supposera que les noms sont écrits en minuscules, sans caractères accentués). La méthode compareTo de la classe String peut être utilisée pour la comparaison des noms des abonnés. Par exemple, l'expression « *str1.compareTO(str2)>0* » exprime que la chaine str1 est plus grande que str2.
   - Une méthode deleteAbonne permettant de supprimer un abonné du répertoire.
   - Une méthode findAbonne permettant la recherche d'un abonné passé en paramètre dans le répertoire.
3) Proposer un programme principal montrant les possibilités d'utilisation des classes Repertoire.et Abonne. Ce programme doit créer au moins un objet de type Répertoire.