Série N 2 Module 11
Programmation Orienté Objet

FILIERE : TDI
NIVEAU : 1er année

Exercices 1 :

Chaque personne possède, dès sa naissance, un père et une mère, et ces liens ne changent pas.

Rédiger la classe Personne qui satisfait aux spécifications suivantes. Une personne possède un nom, un père, une mère et des enfants. Lors de la création d'une personne :

- On indique son nom, son père et sa mère,
- La liste de ses enfants est initialisée à vide,
- Cette personne est rajoutée dans la liste des enfants de son père et de sa mère.

Exercices 2 :

Définissez une classe Equation dans le but d'implémenter des objets représentant une équation du second degré et de donner la possibilité à chaque objet de calculer les racines de l'équation correspondante (l'équation ne traite que des réels).

La classe Equation sera définie par les propriétés :

- **a**, **b** et **c** de type double représentant les coefficients de l'équation
- $ax^2+bx+c=0$ nous supposons que le coefficient est déjà positif
- delta est une propriété de la classe qui représente la valeur du discriminant :
- delta = b^2 - 4*a*c
- x1 et x2, de type double, sont deux propriétés de la classe qui représentent les racines de l'équation lorsqu'elles existent.

1. Implémenter la classe Equation en respectant les règles d'encapsulation.
2. Implémenter la constructeur de cette classe : delta=0, x1=0, x2=0
3. Implémenter la méthode **calcul_Racines()** en respectant les critères suivants : le méthode ne renvoie aucune valeur, la méthode calcule delta, et la méthode calcule x1 et x2 selon la valeur du delta
4. Implémenter une méthode **donne_resultats()** sachant que cette méthode ne renvoie aucune valeur, et affiche les résultats x1 et x2 selon que le delta est positif ou nul ou delta est négatif
5. Tester votre classe : le test doit être effectué pour les 3 cas(delta positif, delta égal à zéro et delta négatif)
6. supplément : Redéfinir la méthode **Equals()**, Surchargé les opérateur **+** et *****

Exercices 3 :

On veut gérer les inscriptions dans une école, pour cela on veut modéliser un Etudiant comme objet. Un étudiant est caractérisé par un num_etud, nom_etud et une adresse.

Ecrire une classe avec le nom **Etudiant** qui comporte les éléments suivants :

1) les attributs *num_etud*, *nom_etud* et une *adresse*
2) les méthodes get et set pour recevoir et modifier les attributs de l'objet :
3) une méthode qui affiche les attributs.
4) Écrire une autre classe *Test* qui permet d'instancier des objets Etudiants par des occurrences telles que (11, « ALI », « 192, HAY NAZHA, BENI MALLEL »)…etc.
5) Ecrire une méthode qui trie les étudiants en fonction de leurs numéros.
6) Dans la même classe *Test*, trier les étudiants et les afficher ensuite.

Exercices 4 :

Il s'agit de modéliser un vecteur de Z^2 dont l'origine est en (0, 0) (un tel vecteur est donc caractérisé par deux nombres entiers relatifs). Les opérations que l'on souhaite faire sur ce segment sont :

- calculer sa longueur, par une méthode d'instance sans paramètre, nommée longueur, et qui retourne cette longueur sous forme d'un double
- savoir si le vecteur concerné est ou non plus petit qu'un autre un autre vecteur donné ; on écrira pour cela une méthode d'instance nommée plusPetitQue qui recevra en paramètre l'autre vecteur et qui retournera une variable de type boolean
- additionner au vecteur concerné un autre vecteur ; on écrira pour cela une méthode d'instance nommée addition qui recevra en paramètre l'autre vecteur et qui ne retournera rien
- additionner deux vecteurs donnés ; on écrira pour cela une méthode statique nommée aussi addition (en utilisant ainsi la possibilité de la surcharge) qui recevra en paramètres les deux vecteurs à additionner et qui retournera le résultat sous forme d'un objet de type Vecteur
- une méthode redéfinissant la méthode : **public** String toString(), Celle-ci décrira une instance de Vecteur sous la forme d'une chaîne de caractères (par exemple, le vecteur de composantes 1 et 2 pourra être décrit par la chaîne de caractères : "vecteur (1, 2)"). La méthode retournera cette chaîne.

On créera une classe EssaiVecteur contenant une méthode main pour tester la classe Vecteur