Année universitaire 2021/2022

TP1 : POO – Classes et Constructeurs

Exercice 1 :

Partie 1 :

Soit Point une classe comportant :

- deux composantes x et y de type entier,

- quarte fonctions membres :

- Une fonction Affiche permettant d'afficher les coordonnées x et y,
- Une fonction Initialise qui permet d'initialiser ces coordonnées,
- Une fonction Deplace permettant de déplacer le point de coordonnées x et y par dx, dy,
- Une fonction Distance qui calcule la distance entre deux points.

a) Ecrire le programme qui permet l'instanciation d'un objet de la classe point et l'appel de ces fonctions membres.

b) Remplacer la fonction Initialise par un constructeur jouant le même rôle.

Partie 2 :

On peut créer un cercle de deux manières :

- Soit en précisant son centre et son rayon
- Soit en précisant son centre et un point du cercle

Les opérations que l'on souhaite exécuter sur un cercle sont :

- getPerimetre() : retourne le périmètre du cercle.
- getSurface() : retourne la surface du cercle.
- appartient(Point p) : retourne si le point p appartient ou non à l'intérieur du cercle.
- affiche() : permettant d'afficher les coordonnées du centre du cercle ainsi que son rayon.

2. Créer la classe Cercle

3. Créer une application qui permet de :

- Créer un cercle défini par le centre c(100,100) et un point p(200,200).
- Créer un cercle défini par le centre c(130,100) et de rayon r=40.
- Afficher le périmètre et le rayon des deux cercles.
- Afficher si le point p(120,100) appartient à l'intersection des deux cercles ou non.

Exercice 2 :

1. Ecrire une classe **Vecteur** comportant :
   - Trois composantes (privées) représentant respectivement l'abscisse, l'ordonnée et la hauteur,
   - Une fonction **affiche**,
   - Deux constructeurs qui affichent l'adresse de l'objet créé :
     - L'un, sans arguments, initialisant chaque composante à 0,
     - L'autre, avec 3 arguments, représentant les composantes, où par défaut, la hauteur est nulle.
   - Un destructeur où on affiche l'adresse de l'objet détruit.

2. Ajouter à la classe Vecteur précédente quatre fonctions membres nommées :
   - **prod_scal** fournissant en résultat le produit scalaire de deux vecteurs,
   - **somme** permettant de calculer la somme de deux vecteurs.

3. Modifier la classe Vecteur de manière que toutes les transmissions de valeurs de type vecteur aient lieu :
   a) par adresse
   b) par référence

Exercice 3 :

Soit **pile_entier** une classe permettant de gérer une pile d'entiers. Ces derniers sont conservés dans un tableau d'entiers alloués dynamiquement. La classe comporte les fonctions membres suivantes:

- **pile_entier(int)** : constructeur allouant dynamiquement un emplacement de n entiers,
- **pile_entier( )** : constructeur sans arguments allouant par défaut un emplacement de 20 entiers,
- **~pile_entier( ) :** destructeur,
- **void empile(int p) :** ajoute l'entier p sur la pile,
- **int depile( ) :** fournit la valeur de l'entier situé en haut de la pile, en le supprimant de la pile,
- **int pleine( ) :** fournit 1 si la pile est pleine, 0 sinon.
- **int vide( ) :** fournit 1 si la pile est vide, 0 sinon.

a) Ecrire la classe pile_entier et toutes ces fonctions membres.

b) Ecrire une fonction main utilisant des objets automatiques et dynamiques du type pile_entier défini précédemment.

c) Mettre en évidence les problèmes posés par des déclarations de la forme :

```
pile_entier
a(10) ;
```

```
pile_entier b
= a ;
```

d) Ajouter à la classe pile_entier le constructeur par recopie permettant de régler les problèmes précédents.

Exercice 4 : (Surcharge des opérateurs)

L'objectif de cet exercice est de définir les opérateurs arithmétiques et de comparaison d'une classe Fraction.

1. Créer la classe Fraction possédant deux données membres « num » et « den » qui correspondent respectivement au numérateur et au dénominateur de la fraction.
2. Définir deux constructeurs d'initialisation : un avec deux paramètres et un autre avec un seul paramètre (numérateur).
3. Définir une fonction membre afficher ().
4. Définir les opérateurs arithmétiques et de comparaison (+, -, *, / et ==) entre deux fractions ayants le même dénominateur.