**Exercice 1**

Dans cet exercice, on veut implémenter une entité compteur définie par une valeur entière. Cette valeur est positive ou nulle à la création d'un objet compteur. Par ailleurs, elle ne peut varier que par pas de 1 (incrémentation ou décrémentation). Il est à noter que la valeur d'un objet compteur est toujours positive ou nulle.

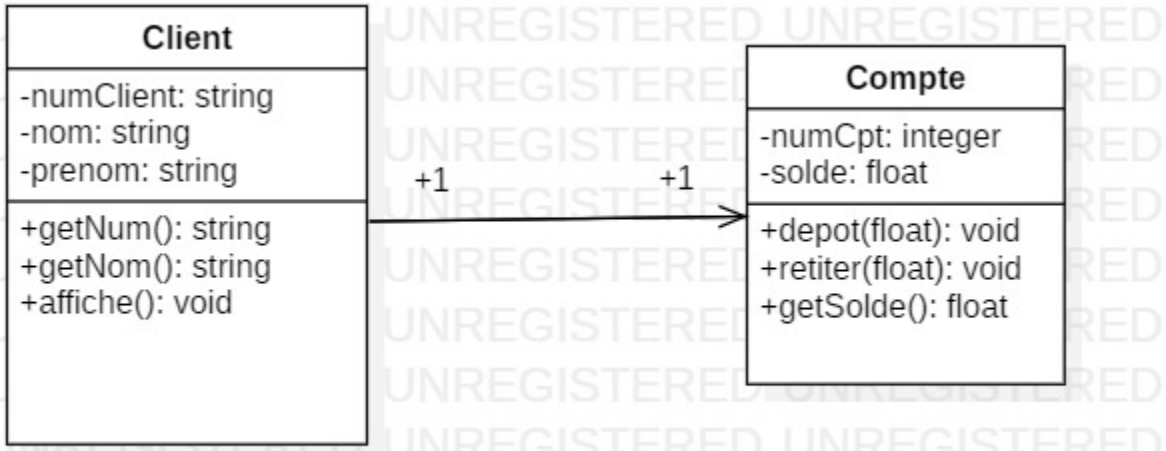
1. Implémenter en C++ l'entité compteur, prévoir les classes *Compteur.h* et *Compteur.cpp*.
2. Ecrire une classe *application* pour créer et manipuler des objets compteurs.

**Exercice 2**

Considérons le diagramme de classes suivants :


1. Certains attributs et méthodes des classes *Client* et *Compte* ont été omis. Les méthodes de ces deux classes sont toujours invoquées par un objet, dit l'objet courant. Les méthodes *getNum* et *getNom* retournent respectivement le numéro et le nom de l'objet client courant. La méthode *affiche* visualise sur la console le numéro, le nom, le prénom et le solde de l'objet client courant. La méthode *depot* (resp. *retirer*) ajoute (resp. retranche) un montant à l'attribut solde du compte courant. La méthode *getSolde* retourne le solde du compte courant. Donner les classes C++ suivantes :
   - *Client.h*
   - *Client.cpp*
   - *Compte.h*
   - *Compte.cpp*

2. On veut rajouter une méthode *effectuer* permettant d'effectuer un virement d'un compte à un autre. Dans quelle classe se trouvera cette méthode?
3. Dites quels sont les changements à apporter à votre implémentation C++ pour prendre en charge la modélisation suivante.




D'après vous, laquelle des deux représentations est à retenir ? Justifier votre réponse.

**Exercice 3**

1. Définir une classe Point avec deux coordonnées de type double Rajouter les méthodes suivantes :

- Un constructeur sans paramètre.
- Un constructeur par paramètres.
- *getX et getY* : qui retournent l'abscisse et l'ordonnée du point.
- *setX et setY* : qui modifient les coordonnées.
- *milieu* : qui retourne le point milieu entre deux points.
- *Symetrie* : qui renvoie un nouvel objet **Point** représentant la symétrie du point courant par rapport à l'origine.
- *egal* : qui teste si le point courant est superposé avec un autre point passé en paramètre, en terme de coordonnées.
- *distance* : qui retourne la distance entre le point courant et un autre point passé en paramètre.

- *afficher* : qui affiche les propriétés du point courant.
- *Deplace* : qui déplace un point (sans en créer un autre) vers un autre emplacement du plan.

2. Définir une classe d'exécution Application qui permet de :
   - Créer deux points p1 et p2 tels que p1(5,6) et p2 (-2, -7).
   - Afficher les caractéristiques du point milieu entre p1 et p2.
   - Afficher la distance entre p1 et p2.
   - Afficher les propriétés du point p11, représentant la symétrie du point p1 par rapport à l'origine.
   - Déplacer p2 pour le superposer avec le point p1.

3. Définir une classe cercle caractérisée par son centre (de type Point) et le rayon (de type double). Rajouter les méthodes qui réalisent ce qui suit :
   - Un constructeur avec paramètres.
   - Affiche les propriétés du cercle.
   - Détermine le périmètre du cercle.
   - Affiche l'équation du cercle.
   - Détermine la surface du cercle.
   - Détermine si un point p de coordonnées x et y appartient ou non au cercle (dans le périmètre).
   - Compare deux cercles (% surface) et rend 0 si égalité, 1 si cercle courant > au second et -1 sinon.

4. Définir une classe Segment qui représente les objets segments. Un segment est défini par deux extrémités de type point. Rajouter les méthodes suivantes :
- *affiche* : affiche les coordonnées des points d'un segment.
- *milieu* : retour le milieu du segment.
- identique : retourne vrai si les extrémités de deux segments coïncident (considérer l'égalité des extrémités en terme de coordonnées).
- *déplace* : déplace un segment vers un autre emplacement du plan.

4. Modifier la classe Segment pour que les segments soient colorés.

**Exercice 3**

1. Définir une classe Monôme composée de deux attributs coefficient de type réel et degré de type entier. On rappelle qu'un monôme s'écrit sous la forme $c\ x^d$ tel que $c$ représente le coefficient et $d$ le degré.

2. - A-t-on besoin d'un attribut pour représenter x ? Justifier votre réponse dans un commentaire.

Compléter la classe par les méthodes suivantes :

- Un constructeur avec deux paramètres.
- Un constructeur avec un seul paramètre réel qui permet de créer un monôme avec un degré nul.
- Une méthode dérivé () qui permet de retourner le monôme dérivé
- Une méthode évaluer ( float x) qui retourne la valeur du monôme pour la valeur x ;
- Une méthode afficher le monôme sous la forme suivante : ***cx^d***
-
- Une méthode qui teste l'égalité de deux monômes.
- Une méthode produit qui retourne le produit de deux monômes

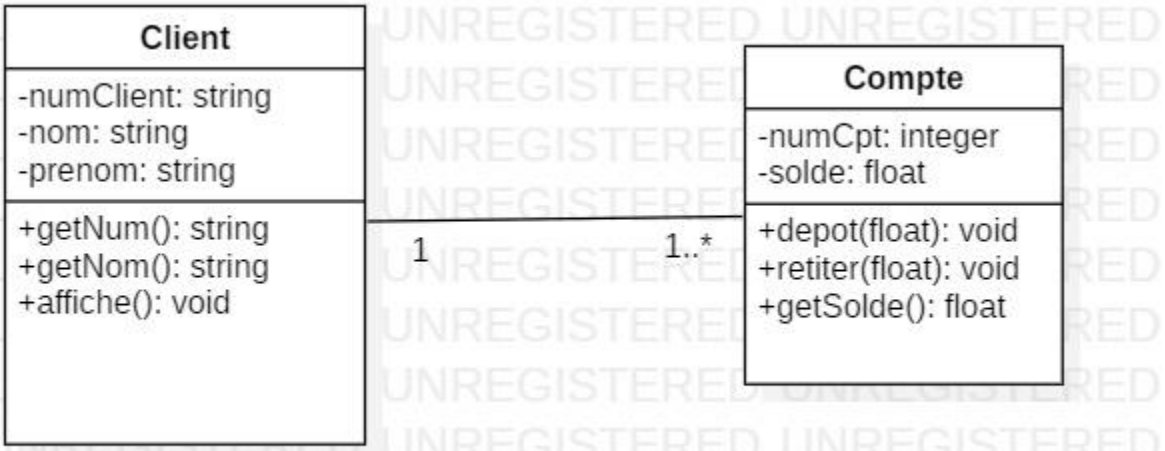
3. Définir une classe Polynôme du second degré. Compléter la classe par les méthodes suivantes :
   - Un constructeur,
   - *racines :* qui permet de résoudre et afficher les résultats de l'équation P(x) = 0,
   - *egal* :Compare deux objets Polynômes.
   - *Evaluer* : évalue le polynôme courant pour une valeur donnée passée en paramètre.
   - *Monome2* : retourne le monôme de plus de degré du polynôme courant.

4. Ecrire une classe Application qui réalise ce qui suit :

   - Crée deux polynômes p1 et p2 et affiche le message «Polynômes identiques « si p1 est égal à p2.
   - Calcule p1(10).
   - Affiche les résultats de l'équation p1(x) = 0.
   - Calcule la valeur du monôme de degré 2 de de p1 au point x=20

**Exercice 4**

1 Reprendre l'exercice 2 et modifier votre implémentation pour la rendre conforme au diagramme suivant :
2 Prévoir une classe application pour tester vos classes.




*« De petits coups, répétés souvent, abattent de grands chênes. »*
*Proverbe latin : Les proverbes et dictons latins (1757)*