

Durant cette séance, vous allez :

- Comprendre la notion d'encapsulation et connaître les niveaux de visibilité en C++.
- Définir les avantages de l'encapsulation et son intérêt.
- Implémenter les getters et les setters dans une classe.

1 Intérêt des méthodes d'accès

Donner la sortie de ce programme :

```
#include <iostream>
#include "Compte.h"
using namespace std;

int main()
{
    Compte c3;
    c3.solde=500;
    c3.numCompte=1;
    if(c3.solde=100)
        cout <<"-----un débit-----" <<endl ;
    else
        cout <<"-----un crédit -----" <<endl ;
}
```

Est-ce que la valeur de solde a été changée ?
Essayez d'implémenter la solution à ce problème.

2 Classe Compte

Créer un projet Banque sous votre répertoire de travail.

Partie 1 :

Créer la classe Compte dans le fichier compte.h ayant les attributs suivants :

- numCompte :entier
- nomProprietaire : chaîne de caractères
- solde : réel

Et les opérations suivantes :

- saisir : permettant la saisie des informations
 - retirerArgent : retire du solde une valeur passée en paramètre et retourne vrai. Si le solde est insuffisant, la méthode retourne faux.
 - deposerArgent : additionne une somme donnée au solde.
 - consulterSolde : affiche le solde actuel.
 - transfererArgent : transfère une somme donnée vers un compte donné en para- mètre
- Définir les opérations en dehors de la classe compte dans un fichier compte.cpp

Créer un programme principal main.cpp permettant de : Définir un Compte A et lui attribuer les valeurs suivantes :

6001
BEN SALEH MOHAMED
850.175

Ajouter au compte 6001 un montant de 100 dinars et afficher le nouveau solde.

Partie 2 :

Créer la classe Banque dans le fichier Banque.h ayant les attributs suivants :

- nomBanque : chaîne de caractères
- Tab[50] : Compte

Et les opérations suivantes :

- chercherCompte(int RIB, int*n) : permettant de vérifier l'unicité des numéros des comptes à ajouter
- ajouter(Compte, int*n) : permettant d'ajouter un compte dans la banque
- afficherBanque(int n) : permettant d'afficher tous les comptes dans la banque.

3 Opérateur Téléphonique

Un opérateur téléphonique souhaite gérer l'ensemble de ces abonnés. Nous allons définir dans cet exercice la classe principale Abonne. Pour chaque abonné, nous allons garder les informations suivantes :

- nom : chaîne de caractères.
- prénom : chaîne de caractères.
- numéro d'appel : entier.
- CIN : entier.
- solde : réel double.

Et les opérations (méthodes) suivantes :

- saisir : permettant la saisie des informations
- afficher : permettant l'affichage des informations
- recharger : cette méthode prend en paramètre un réel. C'est la valeur de la recharge à additionner au solde actuel.
- consommer : cette méthode prend en paramètre le nombre de secondes et le prix d'une minute. La méthode met à jour le solde et retourne vrai. Si la valeur à consommer dépasse le solde, la méthode retourne faux sans changer la valeur du solde. Par défaut, le nombre de secondes est à 12, le prix d'une minute est à 0.18 (180 millimes).

1. Définir entièrement la classe Abonne (interface+implémentation). Il convient pour cela d'implémenter toutes les méthodes nécessaires au bon fonctionnement de la classe.

2. Écrire un programme C++ permettant de tester votre classe.

3. Définir une classe Operateur qui permettra de faire la gestion d'un ensemble des abonnés.