

Evaluation de BTSSN IR2.

Gestion de comptes bancaires

Nous vous proposons de réaliser une classe de gestion de comptes bancaires. Cette gestion comprendra l'id de l'utilisateur : ident, le solde de ce compte : solde, ainsi que les opérations de versements et retraits effectués sur ce compte.

I. Présentation :

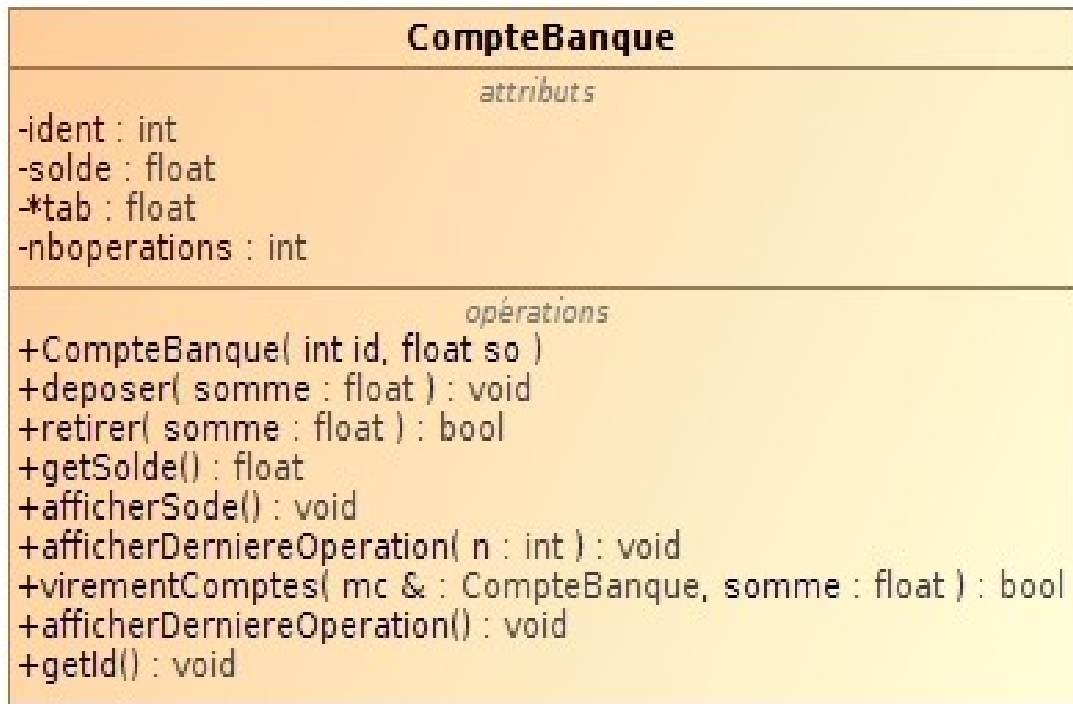


Illustration 1: sous ensemble du diagramme de classe d'une gestion de comptes bancaires.

I.1) Réaliser le projet GestionComptePNom utilisable à partir du makefile, Les fichiers utilisables seront stockés dans un répertoire Classes et se nommeront pour définir la classe CompteBanque comptebanque.h et comptebanque.cpp et le fichier de test testcomptes.cpp (On vous donne le Makefile Associé)

I.2) En vous servant du Illustration 1: sous ensemble du diagramme de classe d'une gestion de comptes bancaires. Réaliser le fichier comptebanque.h où seront définis les attributs privés ident et solde ainsi que le constructeur CompteBanque et toutes les opérations ou méthode.

CompteBanque(int id, float so)

I.3) Réaliser le constructeur CompteBanque(int id, float so) dans un fichier comptebanque.cpp.

I.4) Faire le programme de test de telle sorte que les attributs ident et solde prennent respectivement les valeurs 10 et 40 et où le constructeur est appelé dynamiquement.

I.5) Faire la méthode afficherCompte qui permet d'afficher pour un numéro de compte le solde de ce compte.

Appeler le professeur

I.6) L'ouverture d'un compte bancaire se fait par défaut avec un solde minimum de 20 Euros
Définir le prototype du constructeur avec un solde par défaut correspondant au solde minimal de 20 Euros

Tester votre solution avec l'instantiation suivante dans le programme de test:

`CompteBanque *mcb=newCompteBanque(10);`

I.7) Réaliser et tester les Getteur `getSolde()` et `getId()` qui permettent de récupérer le solde et l'identification d'un compte

Appeler le professeur

Dans les fonctions suivantes vous considérerez que la valeur somme passée en paramètre est positive.

I.8) Réaliser et tester la méthode `void deposer(float somme)`.

Tester pour différents cas judicieux, votre méthode.

I.9) Réaliser et tester la méthode `bool retirer(float somme)`.

Cette méthode retournera `true` si l'opération est possible, c'est à dire que le solde reste positif à la fin de l'opération.

Faire les tests nécessaires à votre méthode.

I.10) Réaliser la méthode de transfert qui permet de faire un virement bancaire d'un compte vers un autre compte, à condition que le compte débiteur ne soit pas à découvert après l'opération.

Faire les tests nécessaires de cette fonction de virementComptes.

➤ Appeler le professeur

I.11) Réaliser les opérateurs `+=` associé à la méthode `deposer`

I.12) Proposer une solution et un test adapté pour transformer la méthode `retirer` en un opérateur `-=`.

➤ Appeler le professeur

Dans la suite on vous demande

Dans la suite on introduit les attributs `int nboperations` et le tableau `*tab`. Ce tableau mémorisera les différentes opérations effectuées par l'utilisateur. Et l'attribut `nboperation` sera initialisé à 1 lors de l'appel du constructeur.

Par exemple si l'utilisateur est à sa quatrième opération et qu'il retire 40.5 Euros, `nboperations` passera de 3 à 4 et `tab[3]=-40.5`.

n'oublier pas que le premier élément d'un tableau est l'élément d'indice 0.

I.13) Dans le constructeur initialiser les variables `nboperations` et le tableau `tab`. Le tableau `tab` sera initialisé dynamiquement à 100 opérations possibles.

`tab= new float[100];`

I.14) Modifier les différentes méthodes déjà réalisées afin de tenir compte des deux nouveaux attributs.

I.15) Réaliser et tester la méthode void afficherDerniereOperation() qui permet d'afficher la dernière opération effectuée et faire le programme de test de votre solution.

Appeler le professeur

I.16) Réaliser et tester l'opération void afficherDerniereOperation qui passe le nombre d'opérations à afficher. Cette méthode passera en paramètre les n dernières opérations à visualiser. Réaliser cette méthode de telle sorte que le nombre d'opération à afficher est inférieur au nombre d'opérations effectués.