Lycée TOUCHARD WASHINGTON

Section des Techniciens Supérieurs

Systèmes Numériques option Informatique et Réseaux

académie Nantes

Projet Banque

~ TP Ctrl 1 ~
Codage C++
Héritage et Composition

Version 4.2 – septembre 2022

Conditions de réalisation

Travail individuel

Durée : 4h

Travail à déposer dans le repository privé, réservé à l'apprentissage du C++, de votre GitHub.

Après chaque étape, transférez votre réalisation sur votre GitHub en indiquant le nom de l'étape (compte bancaire, compte épargne ou compte client).

Ressources utilisées :

Matériel	**Logiciel**
Un ordinateur sous Linux	Git Doxygen Qt Creator

1. CRÉATION DU PROJET

1. Créez un dossier **Projet_Banque** en local dans votre repository d'apprentissage du C++. Il servira de base pour votre projet sous Qt et votre documentation.
2. Créez un sous-dossier **Documentation** dans le dossier **Projet_Banque**, il sera utilisé par la suite pour stocker la documentation de votre logiciel.
3. Sous **QtCreator**, réalisez un projet nommé **LaBanque** de type application console C++ sans utiliser la librairie Qt. Il sera suivi par Git et placé dans un sous dossier du dossier **Projet_Banque**.
4. Ajoutez la classe **Menu**, fournie en ressource, à ce projet .

2. ÉTAPE N°1 : COMPTE BANCAIRE

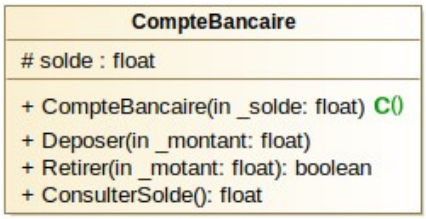
Un compte bancaire possède à tout moment une donnée : son solde. Ce solde peut être positif (compte créditeur) ou négatif (compte débiteur).

À sa création, un compte bancaire a un solde égal à 0 par défaut. Il est cependant possible de créer un compte en précisant un solde initial différent de 0.

Utiliser son compte consiste à pouvoir y faire des dépôts et des retraits. Pour ces deux opérations, il faut connaître le montant de l'opération. Le retrait ne peut se faire que si le compte est suffisamment pourvu.

Il est également possible de consulter le solde du compte.

La modélisation du compte bancaire est représentée par la figure suivante :

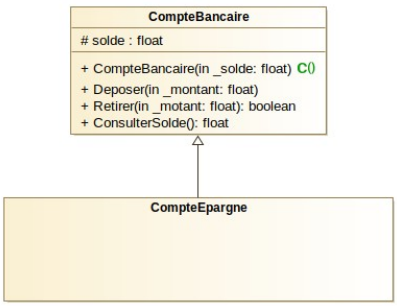


5. Réalisez le codage de cette classe en respectant sa modélisation **UML** et la description proposée.

6. Créer un fichier texte compteBancaire.txt permettant d'afficher le menu suivant :

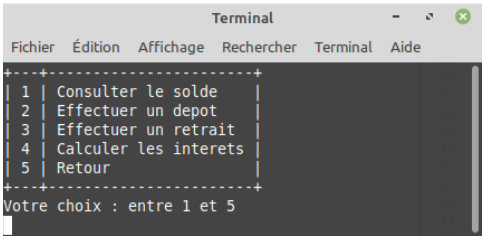
7. Réalisez un programme principal permettant de tester les différentes fonctionnalités du compte bancaire à partir du menu. Tous les affichages sont réalisés dans le programme principal.
8. Modifiez votre programme en intégrant la gestion d'exception de la classe **Menu**.
9. Documentez votre classe **CompteBancaire** ainsi que les fichiers associés et générez la documentation **Doxygen** correspondante dans le sous-dossier **Documentation** de **Projet_Banque**.

3. ÉTAPE N°2 : COMPTE ÉPARGNE



Comme le montre la modélisation UML ci-contre, on peut considérer un compte épargne comme une sorte de compte bancaire. Il en possède les mêmes fonctionnalités avec en plus avec une donnée **tauxInterets** représentant le taux d'intérêt par exemple 2 % et une méthode **CalculerInterets** qui sera utilisée périodiquement pour déterminer le gain apporté par les intérêts. On souhaite que lors de la création du compte épargne le taux d'intérêt soit fixé à la valeur en vigueur depuis le 1er août, 2 % . Le taux étant amené à être modifié, il est nécessaire de prévoir également une méthode **ModifierTaux** afin de pouvoir effectuer la mise à jour de la valeur du taux.

10. Sous Qt dans le projet **LaBanque** créez la classe **CompteEpargne** en respectant la description ci-dessus. Lors de la création du compte épargne, comme pour le compte bancaire, on garde toujours la possibilité soit d'ouvrir le compte avec un certain montant, soit par défaut de lui affecter une valeur nulle.
11. Codez les différentes méthodes de la classe **CompteEpargne**. Pour rappel, le taux d'intérêt est exprimé en pour cent.
12. Documentez votre classe **CompteEpargne** ainsi que les fichiers associés et générez la documentation Doxygen correspondante dans le sous-dossier **Documentation** de **Projet_Banque**.
13. Réalisez un deuxième fichier texte nommé compteEpargne.txt permettant de tester les fonctionnalités du compte épargne comme le montre la figure suivante :

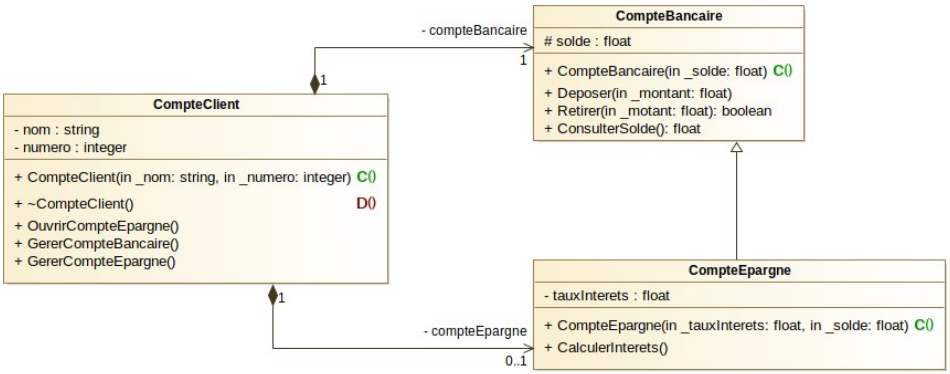


14. **Après avoir mis en commentaire** votre premier programme principal, créez une nouvelle fonction main afin de vérifier le fonctionnement du compte épargne.

4. ÉTAPE N°3 : COMPTE CLIENT

Lorsqu'un client adhère à une banque, l'employé crée un compte client. Ce compte client se compose d'un compte bancaire pour les opérations courantes et éventuellement d'un compte épargne au choix de la personne. Dans tous les cas, un compte client est limité à 1 et 1 seul compte bancaire et au plus 1 compte épargne.

La classe **CompteClient** complète la modélisation UML de l'application :



15. Déclarez la classe **CompteClient** sous Qt en tenant compte des relations que vous avez définies précédemment. L'implémentation choisie pour ces relations est **une forme dynamique**.
16. Le constructeur de la classe **CompteClient**, initialise le nom et le numéro de compte du client, initialise le pointeur sur le compte épargne avec la valeur **nullptr** et alloue dynamiquement la mémoire pour un **CompteBancaire** avec un solde nul.
17. Le destructeur libère la mémoire obtenue pour le compte bancaire et éventuellement libère la mémoire pour le compte épargne s'il existe.
18. La méthode **OuvrirCompteEpargne()**, si le compte n'existe pas, demande à l'utilisateur le montant du solde de départ et le taux de la rémunération du compte et alloue dynamiquement la mémoire pour un compte épargne. Si toute fois le compte, existe déjà, un message prévient l'utilisateur et l'opération est arrêtée.
19. Les deux autres méthodes reprennent le code que vous avez réalisé dans les deux précédents programmes principaux en tenant compte de la manière dont sont implémentés les instances du compte bancaire et du compte épargne
20. Réalisez un fichier nommé client.txt permettant la gestion du menu suivant :

21. Après avoir mis en commentaire votre précédent programme principal, réalisez le programme principal permettant la gestion des deux types de comptes pour un client nommé « Albert » et dont le numéro de compte est 1.