

Modalité du Projet

1. Le projet est reçu par les étudiants avant le Mardi 18-04-2023 à 18h
2. Le projet est à renvoyer par mail au plus tard le vendredi 28-04-2023 à minuit
3. La solution doit être envoyée par mail à l'un des emails des professeurs :
 - a. Yassine.benaboud@gmail.com
 - b. Zineb.dafir@um5s.net.ma
4. L'objet de l'Email doit être « contrôle-POO-2023 » suivi par le nom des étudiants
5. Le projet est à travailler en binôme ou monôme.
6. L'évaluation du travail sera faite à base du code source reçu de la part des étudiants et sur un questionnement oral sur le code pour s'assurer de la bonne compréhension des deux étudiants de chaque binôme.
 - a. Chaque code source qui ne se compile pas résultera à une note maximale de 8/20
 - b. Dans le cas où le code est fonctionnel, la note sera basée sur l'évaluation du professeur responsable.

Cahier des charges

On va créer une application de gestion bancaire. L'application permet de gérer des comptes bancaires de différents types. Pour cela, on définit 5 classes principales :

A. **La classe CompteBancaire** possède les attributs suivants :

- numero (string) : le numéro du compte bancaire
- solde (double) : le solde du compte bancaire

La classe doit fournir les méthodes suivantes :

- Constructeurs
- deposer(double montant) : méthode qui permet de déposer un montant sur le compte bancaire
- retirer (double montant) : méthode qui permet de retirer un montant du compte bancaire
- afficher() : méthode qui affiche les informations du compte bancaire (numéro et solde)

B. **La classe CompteCourant** représente un compte courant et hérite de la classe CompteBancaire. La classe possède les attributs suivants :

- decouvertAutorise (double) : le découvert autorisé du compte courant.

La classe doit fournir les méthodes suivantes :

- Constructeurs

- retirer(double montant) : méthode redéfinie qui permet de retirer un montant du compte courant. Le maximum qu'on peut retirer est solde + découvert autorisé. Le solde peut être négatif.
- afficher() : méthode redéfinie qui affiche les informations du compte courant (numéro, solde et découvert autorisé)

C. La classe CompteEpargne représente un compte d'épargne et hérite également de la classe CompteBancaire. La classe possède les attributs suivants :

- tauxInteret (double) : le taux d'intérêt du compte d'épargne

La classe doit fournir les méthodes suivantes :

- Constructeurs
- déposer(double montant) : méthode redéfinie qui permet de déposer un montant sur le compte d'épargne en prenant en compte le taux d'intérêt. (à chaque dépôt, un pourcentage du montant déposé est ajouté. Ce pourcentage est le taux d'interet).
- afficher() : méthode redéfinie qui affiche les informations du compte d'épargne (numéro, solde et taux d'intérêt)

D. La classe CompteCourantEpargne représente un compte mixte qui hérite des classes CompteEpargne et CompteCourant. La classe possède les attributs suivants :

La classe doit fournir les méthodes suivantes :

- Constructeurs
- déposer(double montant) qui doit combiner le fonctionnement des deux types de comptes.
- retirer(double montant) qui doit combiner le fonctionnement des deux types de comptes.
- afficher() : méthode redéfinie qui affiche les informations du compte d'épargne (numéro, solde et taux d'intérêt)

E. La classe Banque qui permet de gérer les comptes bancaires. La classe possède les attributs suivants :

- comptesBancaires (Tableau de comptes bancaire) : les comptes bancaires gérés par la banque. Initialisé comme tableau vide et alloué dynamiquement.

La classe doit fournir les méthodes suivantes :

- ajouterCompte(CompteBancaire* compte) : méthode qui permet d'ajouter un compte bancaire à la banque
- supprimerCompte(string numero) : méthode qui permet de supprimer un compte bancaire de la banque en utilisant son numéro
- afficherComptes() : méthode qui affiche les informations de tous les comptes bancaires gérés par la banque
- getTotalSoldes() : méthode qui calcule et retourne le total des soldes de tous les comptes bancaires gérés par la banque

Ecrire un main qui permet de :

- a. Créer une banque
- b. ajouter plusieurs comptes de tous les types
- c. Déposer et retirer l'argent des comptes
- d. Supprimer quelques comptes.
- e. Afficher les comptes de la banque et le montant totale

Note : Il faut respecter les noms des classes, attributs, et methodes pour que les testes fonctionnent.