Projet 13 : Gestion des comptes bancaires

**NB :**

* **Toutes les fonctionnalités de ce projet doit être réaliser en utilisant la bibliothèque stl. (pour parcourir, utiliser les itérateurs, pour les traitements, utiliser les algorithmes)**
* **Pour la date, utiliser une structure pour stocker le jour, le mois et la date.**

On considère la conception suivantes :

* **Classe Agence bancaire**:
* Attributs privés : id agence, adresse .
* Relations : Agence – Compte bancaire (1 à \* ), Agence bancaire - Employé (1 à \*)
* **Classe Employé**:
* Attributs privés : id employé, nom, poste .
* Relations : Employé – Agence bancaire (\* à 1 )
* **Classe Compte bancaire (classe abstraite)**:
* Attributs privés : id compte, solde, date création.
* Relations : Compte bancaire – agence (\* à 1), Compte bancaire – Client (\* à 1) , Compte bancaire – opération (1 à \*), classe de base pour les classes : compte épargne et compte ;
* Méthode : relevé bancaire

NB : le relevé bancaire doit contenir les informations suivantes :

- L’identifiant de compte : ……. – Date: ……

- Nom client : ……

- Liste des opérations :

|  |  |  |  |
| --- | --- | --- | --- |
| Référence opération | Date opération | Type opération | montant |
| …. | … | …. | …. |
| ….. | …. | …. | …. |

* Total des opération : ……..
* Solde du compte : …… .
* **Classe Compte épargne**:
* Attributs privés : taux d’intérêt.
* Relations : sous classe de la classe Compte bancaire;
* **Classe Compte retrait**:
* Attributs privés : limite découvert.
* Relations : sous classe de la classe Compte bancaire;
* **Classe Client** :
* Attributs privés : id client, nom, contact ;
* Relations : Client – Compte bancaire (agrégation 1 à \*) ;
* **Classe Opération** :
* Attributs privés : référence opération, date opération, type opération (retrait, dépôt), montant ;
* Relations : Opération – Compte bancaire (\* à 1) ;

**Travail à faire :**

1. Créer les différentes classes ;
2. Dans le main
   1. Créer un menu de navigation pour les différentes fonctionnalités

***\*\*Fonctionnalité 1 : gestion Agence bancaire \*\****

1. Dans un std::vector demander à l’utilisateur de remplir la liste des agence bancaire.
2. Afficher le nombre des employés de chaque agence
3. Permettre à l’utilisateur modifier/supprimer une agence

***\*\*Fonctionnalité 2 : gestion Client \*\****

1. Dans un std::deque demander à l’utilisateur de remplir les clients
2. Permettre à l’utilisateur modifier/supprimer un client
3. Afficher pour chaque client, le nombre des comptes crées

***\*\*Fonctionnalité 3 : gestion des Comptes bancaires \*\****

NB : utiliser les informations déjà enregistrées dans le std ::deque et le std ::vector

1. Dans un std::map, associer l’identifiant compte avec un objet compte bancaire
2. Permettre à l’utilisateur d’ajouter/ modifier/ supprimer un compte bancaire
3. Afficher le nombre des comptes bancaires qui ont été créé il y a 5 ans et dont le solde dépasse les 50000dh
4. Afficher le relevé bancaire de chaque compte

***\*\*Fonctionnalité 4 : gestion Opération\*\****

NB : utiliser les informations déjà enregistrées dans le std ::map, std ::vector et le std ::deque

1. Dans un std::list, demander à l’utilisateur de remplir la liste des opérations,
2. Permettre à l’utilisateur d’ajouter / modifier/ supprimer des opérations