

Spécifications techniques détaillées de la classe AgenceBancaire

Edition A – Rev. 0

Mission

Une banque est organisée en une partition régionale d'agences locales. Le siège social de la banque est lui même une agence.

Chaque agence est chargée de l'accueil de la clientèle et de la gestion des comptes (comptes courants et comptes épargne) de ses clients.

Un même client peut ouvrir plusieurs comptes dans la même agence et/ou dans des agences différentes de la même banque.

Chaque agence produit également périodiquement (jour, mois et année) des résultats de calculs statistiques sur l'ensemble des comptes qu'elle gère. Ces statistiques sont destinées uniquement au siège social de la banque propriétaire.

Dans la banque gestionnaire, les numéros de guichet des agences sont attribués automatiquement à partir d'un registre interne, où les numéros sont consécutifs et démarrent à l'état initial **1001** (numéro du siège social).

Une agence offre à ses clients les services suivants :

- ✓ accueil du public
- ✓ ouverture et fermeture de comptes clients
- ✓ gestion des opérations bancaires courantes (versements et retraits)
- ✓ rémunération des comptes épargnes

Une agence effectue sur demande de la banque gestionnaire les calculs statistiques suivants :

- ✓ nombre de clients possédant au moins un compte dans l'agence
- ✓ nombre de comptes courants ouverts
- ✓ nombre de comptes épargne ouverts
- ✓ total des dépôts en compte courant
- ✓ total des dépôts en compte épargne
- ✓ nombre d'opérations bancaires effectuées sur la période considérée (semaine, mois, année)

I- Exigences fonctionnelles

Les informations à conserver pour une agence bancaire sont :

- | | | | |
|-----------------------------------|----------------|----------------------------------|---------------|
| 1. le numéro de guichet | entier positif | <i>m_numeroGuichet</i> | <i>int</i> |
| 2. l'adresse de l'agence | alphanumérique | <i>m_pAdresseAgence</i> | <i>string</i> |
| 3. l'ensemble des comptes ouverts | Objet | <i>m_comptesBancaires</i> | <i>map</i> |

La classe sera dotée d'un élément neutre, produit par le constructeur par défaut.

La classe devra gérer un constructeur normal, doté d'un seul paramètre (adresse de l'agence). Le numéro de guichet sera obtenu automatiquement, à partir de l'état courant du registre interne de la banque.

Les trois champs feront l'objet d'accesseurs de consultation. Aucun champ n'est modifiable directement et globalement après création.

A la construction, le nombre total de comptes bancaires ouverts dans une agence sera nul.

Un client peut ouvrir dans la même agence plusieurs comptes courants et/ou plusieurs comptes épargne.

Pour ouvrir un nouveau compte, la classe fournira l'accesseur de modification ***ouvrirCompte***.

Pour un compte courant, la surcharge recevra en unique paramètre la désignation du titulaire. Pour un compte épargne, la surcharge recevra en outre le taux de rémunération mensuel du compte

Dans les deux cas, la méthode retournera le numéro du compte qui a été créé.

Pour fermer un compte, la classe l'accesseur de modification **fermerCompte**, qui recevra en paramètre le numéro du compte cible et retournera le montant du solde au moment de la clôture.

Un compte fermé reste présent et comptabilisé dans la liste des comptes de l'agence.

Pour traiter les opérations bancaires courantes de ses clients, la classe fournira le service **executer**.

La méthode **executer** recevra en paramètres :

- le numéro du compte cible
- le type de l'opération
- le libellé de l'opération
- le montant de l'opération

La méthode retournera le solde du compte, après exécution.

Pour les calculs statistiques, la classe fournira les services suivants :

- **nombreClients** : nombre de clients possédant au moins un compte dans l'agence
- **nombreComptes** : nombre de comptes ouverts (courant ou épargne)
- **totalDepots** : total des dépôts en comptes (courant ou épargne)
- **nombreOperations** : nombre total d'opérations bancaires exécutées sur la période considérée

La méthode **nombreClients** n'aura pas de paramètre et retournera le cardinal calculé.

Les méthodes **nombreComptes** et **totalDepots** recevront le type de compte cible (compte courant ou compte epargne) et la désignation du titulaire. Elles retourneront la quantité calculée.

La méthode **nombreOperations** recevra en paramètre la description de la période considérée et retournera le cardinal calculé. La statistique peut être demandée pour :

- une journée cible : aaaa/mm/jj
- un mois cible : aaaa/mm
- une année cible : aaaa

II- Contraintes de performances

Une agence bancaire est susceptible de gérer plusieurs centaines de clients et par voie de conséquence plusieurs centaines de comptes (tous types confondus).

La classe devra être conçue pour minimiser les temps d'accès moyen en consultation des comptes bancaires.

III- Contraintes de développement

La classe devra satisfaire aux spécifications techniques générales du package **_Banque** et respecter toutes les contraintes de conception, de codage et de tests imposées pour les TP du module APO C++.

Les codes d'exceptions fonctionnelles propres à cette classe sont :

- 3.0 : duplication interdite pour une agence bancaire
- 3.1 : tentative de fermeture d'un compte inexistant
- 3.2 : tentative de fermeture d'un compte déjà fermé
- 3.3 : type de compte inconnu dans une agence
- 3.4 : tentative d'accès à un compte bancaire fermé
- 3.5 : numéro de compte inconnu dans l'agence