

Spécifications techniques détaillées de la classe *Banque*

Edition A – Rev. 0

Mission

Une banque est organisée au niveau national (France) en une partition de régions. Cette organisation ne coïncide pas nécessairement avec les 26 régions administratives.

Chaque région est couverte commercialement par une partition d'agences. Le siège social de la banque est lui même une agence.

La mission de chaque agence est détaillée dans la spécification technique détaillée de l'entité "Agence".

Au sein de cette organisation, les principales fonctions de l'entité "Banque" sont les suivantes :

- attribution centralisée des numéros de guichet,
- attribution centralisée des numéros de compte (courant et épargne),
- consolidation et archivage périodique des statistiques fournies par chaque agence,
- réponses aux enquêtes administratives et financières

I- Exigences fonctionnelles

Les informations à conserver pour une banque sont :

- | | |
|---|--------------------------|
| 1. son nom | alphabétique |
| 2. un code numérique (5 chiffres) qui identifie la banque | numérique |
| 3. l'adresse de son siège social | alphanumérique |
| 4. son numéro SIRET (14 chiffres) | numérique |
| 5. la description de son organisation régionale | dictionnaire des régions |
| 6. l'état courant du registre des numéros de guichet | numérique |
| 7. l'état courant des registres des numéros de comptes | numérique |

Chaque instance de la classe possèdera les propriétés suivantes :

- | | |
|--|--------------------------------|
| - son nom | <i>nomBanque</i> |
| - un code numérique (5 chiffres) qui identifie la banque | <i>codeBanque</i> |
| - l'adresse de son siège social | <i>adresseBanque</i> |
| - le numéro SIRET (14 chiffres) | <i>numeroSIRET</i> |
| - la description de son organisation régionale | <i>regions</i> |
| - l'état courant du registre des numéros de guichet | <i>registreAgences</i> |
| - l'état courant du registre des numéros de comptes courants | <i>registreComptesCourants</i> |
| - l'état courant du registre des numéros de comptes épargne | <i>registreComptesEpargne</i> |

L'entité "Banque" sera instanciable.

La classe devra gérer un constructeur normal, doté des paramètres suivants : adresse de la banque, code d'identification de la banque et numéro SIRET.

A l'exécution du constructeur normal, l'organisation commerciale ne sera décrite que plus tard.

La classe devra interdire toute duplication de banques.

Pour chaque région, la banque gère un dictionnaire de ses agences, dont la clé d'indexation est le numéro de guichet de chaque agence.

Tous les champs feront l'objet d'accesseurs de consultation.

Aucun champ n'est modifiable globalement après création (pas d'accesseurs de modification).

La gestion des numéros de compte est nationale.

L'entité "*Banque*" offre les services suivants :

- ✓ ouverture et fermeture d'agences
- ✓ délivrance des numéros de compte
- ✓ consolidation des statistiques agences
- ✓ élaboration du tableau de bord pour son conseil d'administration

L'entité "*Banque*" effectue les consolidations **par région** et au **plan national** des calculs statistiques suivants :

- ✓ nombre de clients possédant au moins un compte
- ✓ 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)

Les enquêtes administratives et financières auxquelles est tenue de répondre une banque sont les suivantes :

- ✓ liste des numéros de compte ouverts d'un client désigné par son numéro INSEE,
- ✓ montant des dépôts sur un compte désigné par son numéro,
- ✓ nombre total courant d'agences (état national),
- ✓ montant total courant des dépôts dans la banque cible

Pour la gestion des régions commerciales, la classe fournira une méthode ***creerRegion*** et une méthode ***supprimerRegion***. Une région ne peut être supprimée que si toutes ses agences ont été fermées au préalable.

Pour la gestion des agences bancaires, la classe fournira une méthode ***ouvrirAgence*** et une méthode ***fermerAgence***. Une agence ne peut être fermée que si tous les comptes qu'elle gère ont été transférés au préalable dans une autre agence (méthode ***transférerComptes***).

Pour les calculs statistiques, la classe fournira les méthodes suivantes :

- ***nombreClients*** : nombre de clients possédant au moins un compte dans la banque
- ***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

Pour les réponses aux enquêtes administratives et financières, la classe fournira les méthodes suivantes :

- ***comptesClient*** : liste des comptes du client cible
- ***depotsClient*** : total des dépôts d'un compte cible
- ***nombreAgences*** : nombre total d'agences
- ***totalDepots*** : montant total des dépôts

II- Exigences d'interface

La méthode ***creerRegion*** recevra en paramètre la désignation de la région commerciale à créer.

La méthode ***supprimerRegion*** recevra en paramètre la désignation de la région commerciale à supprimer.

La méthode ***ouvrirAgence*** recevra en paramètres :

- le nom de la région commerciale de rattachement
- l'adresse commerciale de cette agence

Elle fournira en retour le numéro de guichet de la nouvelle agence.

La méthode ***fermerAgence*** recevra en paramètre le numéro de guichet de l'agence cible.

La méthode ***transférerComptes*** recevra en paramètres :

- le numéro de guichet de l'agence d'origine
- le numéro de guichet de l'agence de destination

La signature formelle des méthodes de calculs statistiques sera analogue aux méthodes de même nom des agences bancaires, avec désignation éventuelle en sus (premier paramètre) de la région commerciale cible.

II- 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 de cette classe sont :

< A COMPLETER >