
Cette feuille poursuit le développement du package ***_Banque***, avec la finalisation de la classe ***AgenceBancaire***.

Exercice 1. La classe *AgenceBancaire* - Version 1.4.0 / Méthode privée *nombreComptes*

On introduit dans le fichier "header" de la classe une méthode privée ***nombreComptes***.

Développer les codes sources de cette méthode à partir du squelette fourni en annexe et exécuter avec succès le module de tests unitaires également fourni en annexe.

Les comptes fermés ne doivent pas être comptabilisés par ce service. Le service retournera 0 si le paramètre ne désigne pas un client de l'agence.

Exercice 2. La classe *AgenceBancaire* - Version 1.4.0 / Première forme du service *nombreComptes*

Développer les codes sources de la forme du service ***nombreComptes*** qui reçoit en unique paramètre le type de compte, à partir du squelette fourni en annexe.

Exécuter avec succès les modules de tests unitaires également fournis en annexe.

Exercice 3. La classe *AgenceBancaire* - Version 1.4.0 / Surcharge de l'opérateur ()

Exploiter le squelette fourni en annexe pour développer la surcharge de l'opérateur () et exécuter avec succès le module de tests unitaires également fourni en annexe.

Justifier le rôle de cette surcharge.

Une surcharge de l'opérateur [] aurait elle été possible en lieu et place de celle de l'opérateur () ?

Exercice 4. La classe *AgenceBancaire* - Version 1.4.0 / Seconde forme du service *nombreComptes*

Développer les codes sources de la forme du service ***nombreComptes*** qui reçoit en paramètres la description du client (instance de la classe ***Titulaire***) et le type de compte, à partir du squelette fourni en annexe.

Exécuter avec succès les modules de tests unitaires également fournis en annexe.

Exercice 5. La classe *AgenceBancaire* - Version 1.4.0 / Troisième forme du service *nombreComptes*

Développer les codes sources de la forme du service ***nombreComptes*** qui reçoit en unique paramètre la description du client (instance de la classe ***Titulaire***), à partir du squelette fourni en annexe.

Exécuter avec succès les modules de tests unitaires également fournis en annexe.

Exercice 6. La classe *AgenceBancaire* - Version 1.4.0 / Caractère privé des opérateurs [] et ()

Pour quelle raison fonctionnelle, les surcharges des opérateurs [] et () doivent elles être privées ?

Transformer ces surcharges en méthodes privées.

Exercice 7. La classe *AgenceBancaire* - Version 1.4.0 / Nouvelle méthode privée

Définir, implémenter et valider une nouvelle méthode privée permettant de factoriser tout ou partie des traitements communs aux trois formes du service ***totalDepots***.

Exercice 8. La classe *AgenceBancaire* - Version 1.4.0 / Première forme du service *totalDepots*

Développer les codes sources de la première forme du service ***totalDepots*** qui reçoit en unique paramètre le type de compte.

Exécuter avec succès le module de tests unitaires fourni en annexe.

Dans le cas d'un compte épargne, le calcul du montant des dépôts devra inclure les arriérés éventuels de rémunération à la date du calcul.

Exercice 9. La classe *AgenceBancaire* - Version 1.4.0 / Les autres formes du service *totalDepots*

Exploiter les squelettes fournis en annexe pour développer les codes sources des deux autres formes du service *totalDepots*.

Exécuter avec succès les modules de tests unitaires fournis en annexe.

Exercice 10. La classe *AgenceBancaire* - Version 1.4.0 / Le service *nombreOperations*

Développer le service *nombreOperations* et ses modules de tests unitaires.

Exercice 11. La classe *AgenceBancaire* - Version 1.4.0 / Cas d'anomalies

Développer le module de tests unitaires de tous les cas d'anomalies.