

Module : Conception et Programmation Objet Avancées (CPOA)

TP2 Atelier de Génie Logiciel (1h30)

PARTIE A. Rétro-ingénierie

1. Lancer Umbrello, utiliser l'assistant de rétro-ingénierie (*Code Importing Wizard*) afin d'importer les 4 classes java de l'application *Bank* (disponible sous Célène)
2. Par « glisser-déplacer » de la vue arborescente vers le diagramme de classe, ajouter la classe *Compte* au diagramme de classe
3. Observer la notation UML de cette classe et modifier au besoin les propriétés d'affichage (clic droit dans la fenêtre du diagramme) de manière à faire figurer l'ensemble des attributs et opérations.
4. Ajouter à présent les classes *Client*, *Operation* et *Bank*, enregistrer votre conception Umbrello puis exporter l'image du diagramme UML de Classes obtenu.

PARTIE B. Conception améliorée

Modifier le diagramme précédent de sorte à faire apparaître :

1. Deux types de comptes : *CompteCourant* et *CompteDepot* (sans découvert autorisé mais avec un *soldeMinimum*)
2. Une opération *crediter()* commune aux deux types de comptes
3. Une opération *debiter()* propre à chaque type de compte :
 - pour un compte courant : le débit n'est possible que si le solde final reste supérieur au découvert autorisé
 - pour un compte de dépôt : le débit n'est possible que si le solde final reste supérieur au solde minimum
4. Un taux d'intérêt et un solde minimum qui sont les mêmes pour tous les comptes de dépôt.
5. Générer le code des nouvelles classes uniquement.
6. Observer et corriger le code Java final puis le tester :
 - ajouter un compte courant « CCP1 » au client « Dupont » (client par défaut)
 - lui ajouter un compte de dépôt « CDD1 »
 - consulter le compte « CCP1 »
 - ajouter des opérations (crédits et débits) sur ces comptes puis afficher les historiques

PARTIE C. Rien n'est simple dans le domaine bancaire

Nous nous intéressons uniquement (pour le moment) au fonctionnement des débits sur un compte courant. Modifier progressivement l'application (le code Java directement tout en tenant à jour le diagramme de classes sous Umbrello) de manière à tenir compte successivement de chacune des règles suivantes :

1. un compte est bloqué dès que son solde est inférieur au découvert autorisé ; plus aucun débit n'est alors possible ; les crédits sont toujours possibles sur un compte bloqué ; le déblocage d'un compte nécessite l'intervention d'un agent de la banque.
2. il existe un plafond de débit de sorte que chaque client peut décider d'un commun accord avec son conseiller financier – et pour se prémunir de tout piratage – d'interdire sur son compte toute opération consistant en un débit supérieur à ce plafond.
3. la banque décide de son côté de plafonner les débits à 100€¹ lorsqu'un compte est à découvert.
4. (*bonus*) pour limiter l'impact lié à un piratage du compte, la banque met en place un système de plafond de débits sur une fenêtre glissante : la somme des débits sur les 30 derniers jours ne doit pas dépasser un certain plafond. Une opération de débit qui ne satisfait pas cette règle est refusée.

Une fois toutes ces règles cumulées implémentées, tester l'application de façon à vérifier que les débits sont gérés de manière conforme à ce qu'a défini la banque.

5. Pensez-vous que la procédure de gestion des comptes et débits sera facilement modifiable ? Y'a-t'il des portions de votre codes (expressions ou séquences d'instructions) dupliquées ?

→ Rendez-vous sur Célène pour un QCM de restitution de connaissances (5').

¹ Ce montant peut être réévalué par la banque à tout moment, mais il s'applique de façon indifférenciée sur tous les comptes.