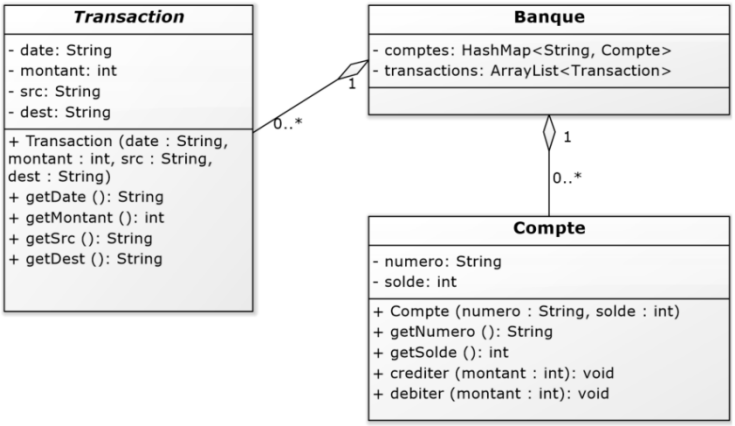
# Art du Développement Moderne – TD 1 – « Prélude »

## Exercice 1 - Entraînez-vous pour améliorer votre maîtrise d'IDE et remise à niveau Java

Considérons le programme Java correspondant au diagramme de classes UML (illustré à droite) permettant de gérer les transactions (transferts) entre les comptes bancaires. Chaque compte bancaire a un numéro alphanumérique unique et un solde (toujours supérieur ou égal à 0). Les deux méthodes « crediter » et « debiter » permettent de modifier le solde en ajoutant ou en enlevant un montant. Pour chaque transaction, on a besoin de connaître sa date (chaîne de caractères sous format « AAAA-MM-JJ »), le montant du transfert, le numéro du compte « source » (src) et le numéro du compte « destination » (dest).




**1.1** Ecrivez le code de base de toutes les classes, avec leurs attributs, méthodes et constructeurs mentionnés dans le diagramme.

**1.2** Ecrivez le code Java d'une nouvelle méthode « **ajouterCompte** » dans la classe « **Banque** » permettant d'ajouter un nouveau compte. On indique le numéro du compte à ajouter, ainsi que le solde initial. Si le numéro du nouveau compte existe, l'ajout sera refusé et la méthode lève une « *IllegalArgumentException* » en donnant un message d'erreur significatif. La même exception, en donnant un message d'erreur différent, sera levée quand le solde initial indiqué est inférieur à 0.

**1.3** Ecrivez le code Java d'une nouvelle méthode « **ajouterTransaction** » dans la classe « **Banque** » permettant d'ajouter une nouvelle transaction. La méthode met à jour également les soldes des deux comptes concernés. La méthode reçoit quatre paramètres : la date et le montant de la transaction, le numéro du compte « source » et celui du compte « destination ». Si un de deux numéros (ou les deux) n'existe pas, la méthode lève une « *NoSuchElementException* » en donnant un message d'erreur significatif. Si le nouveau solde du compte « source » devient négatif après la soustraction du montant de la transaction, l'ajout sera également refusé et la méthode lève une « *IllegalArgumentException* » en donnant un message d'erreur significatif.

**Avant de programmer :** Configurez bien les raccourcis claviers fréquemment utilisés. Lisez bien le sujet et rassurez-vous d'une compréhension totale, afin de ne pas perdre beaucoup de temps sur la réflexion (hésitation) lors de la programmation.

Réalisez le programme 1.1 – 1.3 et mesurez précisément le temps utilisé pour chaque exercice. Analysez les points perfectibles. Si vous n'arrivez pas à terminer cet exercice avant 11H30, vous devez vous arrêter. Vous devrez rattraper au plus vite le retard en Java par vous-même. Dans les deux cas (fini ou non fini), vous devez faire un petit résumé (vitesse + analyse).

**Important :** Vous devez avoir un logiciel vous permettant de réaliser facilement des vidéos qui enregistrent ce que vous faites sur votre ordinateur (écran + son du micro pour explication). Ceci sera indispensable bientôt pour les TP notés (à partir de la séance 3).

## Exercice 2 – Votre expérience en développement

Vous avez tous eu des expériences en développement (programmation) depuis que vous avez commencé à étudier l'informatique. Ecrivez un texte dans lequel vous raconterez (ordre chronologique) ce que vous avez vécu comme expérience en programmation avec différents langages : projets universitaires, petit travail personnel pour le plaisir, stage, travail à l'entreprise, etc. Pour chaque expérience, vous décrirez surtout votre ressenti, par exemple : En quels aspects cela vous a enrichi ? Qu'avez-vous appris pour la suite ? Comment cela vous a influencé pour votre orientation (carrière) ? Cet exercice est très important en termes de SEDAMOP, car il vous entraîne à vous exprimer (**rétrospectives**) à ce sujet, pour les occasions importantes, comme par exemple un entretien d'embauche. Pour cette séance, vous déposerez une première version (NOM_Prénom_ADM_1.pdf), avec le résumé de l'exercice 1, dans le devoir créé par l'enseignant sur Teams (non évalué, mais votre attitude et rigueur **seront bien évaluées** cette fois-ci en Séance 1) au plus tard à 12H15.

En séance 2, l'enseignant fournira un texte modèle qui pourrait vous inspirer et vous aurez une version améliorée du rendu. Cette dernière sera alors évaluée.