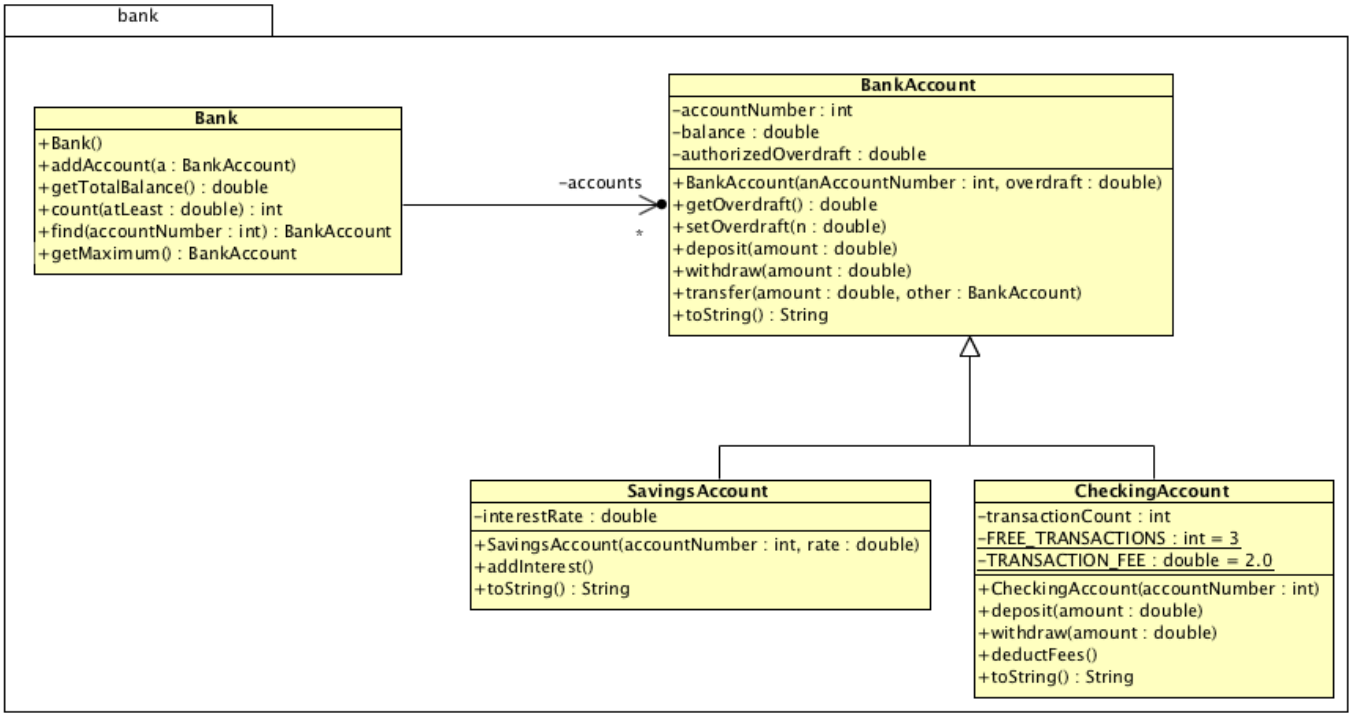
M2103– TD 6

Objectifs de la séance :

- Apprendre à utiliser l'héritage
  - chaînage des constructeurs
  - surcharge de méthodes

L'application Bank

Nous allons définir une petite application qui permet à une banque de gérer des comptes en banque et de réaliser des opérations telles que retrouver un compte à partir de son numéro, obtenir le compte en banque dont la balance est la plus grande, obtenir les numéros des compte sont le solde est au moins égal à une valeur donnée.
On remarquera que les classes sont dans un package (bank).



La classe CompteBancaire

Un compte bancaire est représentée par :

- accountNumber : le numéro classique d'un compte qui est un entier
- balance : le solde courant du compte
- authorizedOverdraft : le découvert maximum autorisé pour le compte (a priori un nombre nul ou négatif)

Les méthodes de cette classe permettent de faire des dépôts, des retraits et des transferts vers un autre compte.

Les comptes sont créés avec un solde (balance) de 0.

La classe BankAccount est spécialisée par deux sous-classes :

- **SavingsAccount** ajoute à un compte la possibilité de cumuler des intérêts. Il possède un taux d'intérêt passé en paramètre du constructeur de cette classe. Les intérêts sont ajoutés manuellement.
- **CheckingAccount** est un compte chèque pour lequel les transactions sont payantes au delà de 3. les transactions sont comptées au fur et à mesure des aouts et retraits et les frais sont déduits manuellement plus tard.

La classe Bank

La classe `Bank` est chargée de gérer un ensemble de comptes bancaires.

- double getTotalBalance() : retourne la somme des soldes de tous les comptes ;
- int count(double atLeast) : retourne le nombre de comptes dont le solde est au moins la valeur *atLeast* ;
- BankAccount find(int accountNumber) : retourne le compte dont le numéro est *accountNumber ;*
- BankAccount getMaximum() : retourne le compte ayant le plus grand solde.

1- Héritage des attributs et des méthodes

Dessiner un diagramme d'objets correspondants aux objets définis par :

```
SavingsAccount momsSavings = new SavingsAccount(203, 0.5);
CheckingAccount harrysChecking = new CheckingAccount(204);
BankAccount dadsAccount = new BankAccount(200, -100).
```

2- Définition des sous-classes et chaînage des constructeurs

- Ecrire pour la classe **BankAccount** : le constructeur et les méthodes de dépôt et de retrait.
- Ecrire les entêtes de définition des deux nouvelles classes **SavingsAccount** et **CheckingAccount** ainsi que leurs constructeurs.

3- Surcharge de méthodes

Rappel : Quand on utilise la pseudo-variable `super` suivie d'un message, pour appeler une méthode de la superclasse, alors ce message peut être situé n'importe où dans la méthode.

- Ecrire les méthodes de dépôt, de retrait pour les deux sous-classes.
- Ecrire les méthodes **toString()** pour les trois classes.

M2103– TP 6

Le TP consiste à écrire l'ensemble des classes du package bank.

Ecrire un méthode scénario (à l'extérieur du package)pour vérifier le fonctionnement des 3 classes de compte.

Ecrire une méthode scénario (à l'extérieur du package) pour vérifier le fonctionnement de la classe Bank.