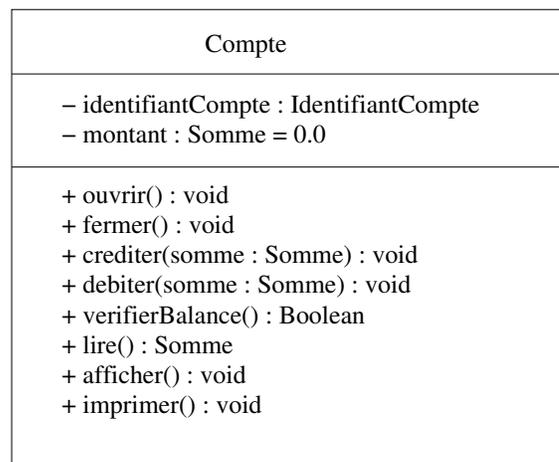


UML - diagrammes de classes

Exercice 1

Soit la classe *Compte* définie en UML comme suit :

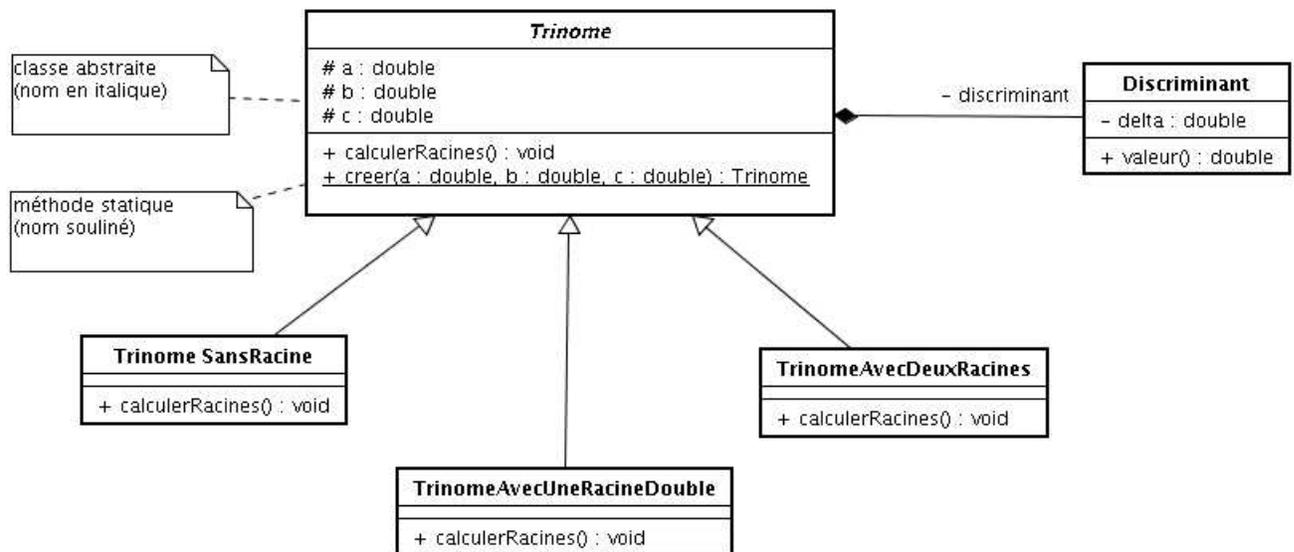


1. Donner la classe correspondante en Java.
2. Donner une autre représentation de diagramme de classe en UML en utilisant les relations de composition.
3. Montrer dans un diagramme de classe UML la relation d'utilisation (ou de visibilité) entre la classe *Compte* et la classe *Somme*.

Exercice 2

On s'intéresse à l'évaluation des racines réelles d'un trinôme du second degré : $aX^2 + bX + c$.

Donner le code Java correspondant au diagramme de classe suivant :



Rappels mathématiques :

Le discriminant δ vaut $b^2 - 4ac$.

- si $\delta < 0$: il n'y a pas de racines réelles.
- si $\delta = 0$: les 2 racines sont confondues ; leur valeur est : $-\frac{b}{2a}$
- si $\delta > 0$: il y a 2 racines : $x_1 = \frac{-b + \sqrt{\delta}}{2a}$ et $x_2 = \frac{-b - \sqrt{\delta}}{2a}$

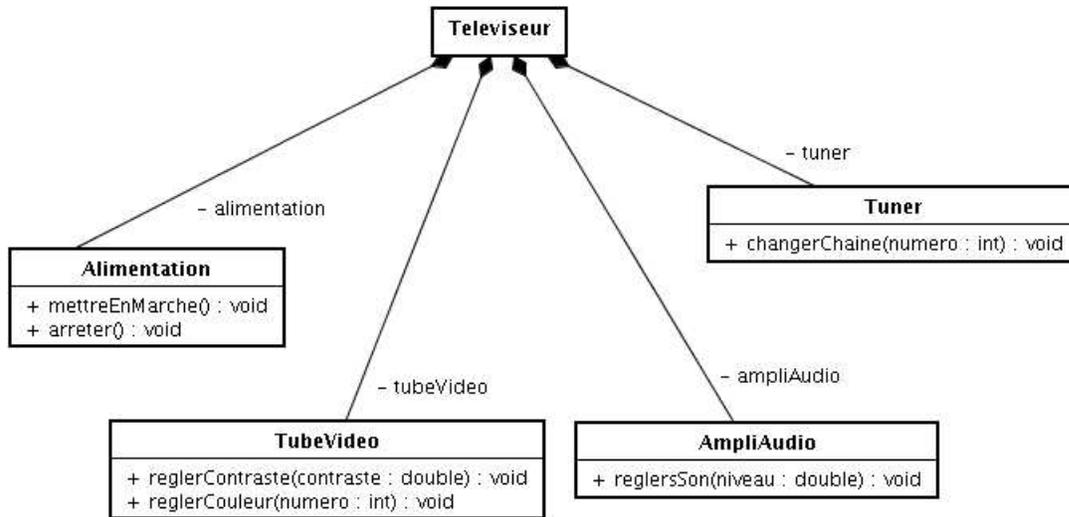
Rappel java :

Le calcul de la racine carré peut se faire à l'aide de la méthode `sqrt` définie dans la classe `Math` comme suit : `static double sqrt(double a)`.

Notes :

- Dans cette première version, les valeurs des racines ne sont pas stockées comme variable d'instance ; le calcul et l'affichage du résultat est fait simultanément. Le stockage des valeurs des racines pourra être envisagé dans une version ultérieure.
- Dans cette version, le calcul des racines complexes n'est pas davantage envisagé. Il pourra être ajouté dans une version ultérieure :
 - $\delta < 0$ et on désire des racines réelles, la réponse reste "pas de racines".
 - $\delta < 0$ et on désire des racines complexes, les 2 solutions sont : $x_1 = \frac{-b + i\sqrt{-\delta}}{2a}$ et $x_2 = \frac{-b - i\sqrt{-\delta}}{2a}$

Exercice 3



Donner une autre représentation de diagramme de classe en UML en supprimant les relations de composition.

Exercice 4

Soit le code Java suivant :

```
class 01 {
    public static int var01 = 124;
    protected double unAttribut = 0.0 ;
    private 02 lien02 ;
    private 03 lien03 ;

    public 01(03 lien 03) {
        lien02 = new 02();
        this.lien03 = lien03;
    }
}

class 02 {
    public 02() {}
}

class 03 {}
```

Donner un diagramme de classe en UML en mettant en évidence les liens de composition et d'agrégation.

Exercice 5

Dans l'application considérée, chaque client possède un compte personnel. A chaque commande pouvant s'effectuer par téléphone, par courrier traditionnel ou par Internet, le compte d'un client est crédité du montant de la commande. A la réception d'un paiement par chèque, par virement bancaire ou encore par carte bancaire, le compte sera débité de la somme correspondante. Périodiquement, des vérifications sont effectuées ainsi que des rappels aux clients si la balance du compte n'est pas à zéro.

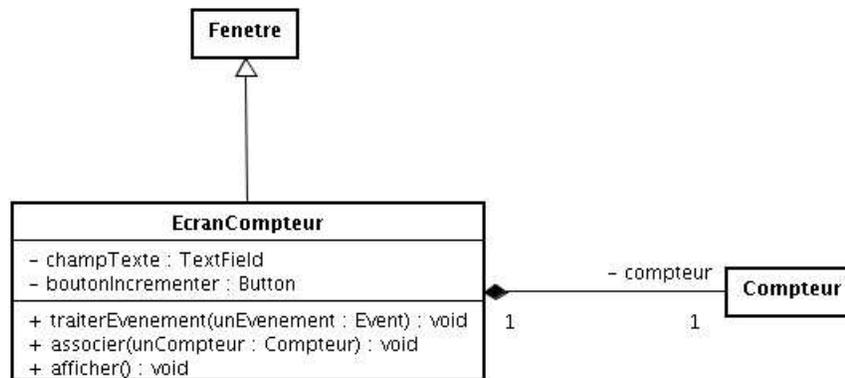
Questions :

- A partir du texte, donner les noms des classes qu'on pourrait déduire.
- Donner la représentation graphique représentant le compte d'un client.

Exercice 6

On désire gérer un compteur avec la possibilité de l'incrémenter, de le remettre à zéro et de lire sa valeur.

1. Nous pouvons définir un compteur comme *int* ou définir une classe *Compteur*. Dire pourquoi la première proposition est peu judicieuse. Définir la classe *Compteur* sous forme de classe en UML.
2. Pour définir une application utilisant la classe *Compteur*, une fenêtre graphique d'interface homme-machine (IHM) constituée d'un champ texte et d'un bouton est définie comme suit :



La classe *Compteur* a été définie, livrée et est en cours d'exploitation quand apparaissent de nouvelles applications avec un compteur dont l'incrément n'est plus forcément égal à 1. Nous pourrions modifier la classe *Compteur* pour ajouter une variable interne qui serait initialisée lors de la création d'un objet particulier. Cette proposition n'est pas retenue car elle risque de toucher aux applications existantes qui fonctionnent. Donner une autre solution pour réaliser la modification souhaitée.

3. Pour réaliser l'indépendance entre l'écran et le compteur, on souhaite ajouter une interface *RoleCompteur*. Donner le nouveau diagramme de classe.
4. Nous disposons d'une classe *Tourniquet* permettant de gérer, par exemple, le nombre de personnes présentes dans un musée. Que faudrait-il faire pour pouvoir re-utiliser l'écran développé pour l'application Compteur ?