

# Sommaire de l'activité

**Partie1: Définir l'héritage et le polymorphisme**

**Partie2: Connaitre l'encapsulation**

**Partie3: Caractériser l'abstraction**

# Description de l'activité

---

## Compétences visées

- Maîtrise des concepts de base relatifs à l'héritage, les types d'héritage et la redéfinition des méthodes
- Maîtrise du principe de polymorphisme: types de variables, types des objets, appel des méthodes, etc.
- Maîtrise du principe de l'encapsulation, intérêt et la distinction entre les différents niveaux de visibilité des membres d'une classe.
- Bonne utilisation des classes abstraites et des interfaces

## Recommandations clés

- Bonne compréhension des notions et des exemples présentés.
-

# Activité 2

CONNAITRE LES PRINCIPAUX PILIERS DE  
LA POO

**Partie1: Définir l'héritage et le polymorphisme**

3

# QCM

## 1. Quelle assertion se rapproche le plus du concept d'héritage?

- A. C'est un mécanisme qui permet de changer le type des objets.
- B. C'est une relation de spécialisation entre une classe et ses classes filles.
- C. C'est un synonyme du concept « instanciation ».
- D. Cela permet à un objet de connaître la valeur d'un attribut d'un autre objet.

## 2. Qu'est-ce qu'une « super classe »

- A. une classe mère
- B. une classe ayant des membres publiques
- C. une classe possédant des privilèges

## QCM (suite)

**3. La classe B hérite de la classe A. Si A possède 3 méthodes et que B en possède 2 qui lui sont propres, combien de méthodes différentes un objet de type B pourra-t-il utiliser?**

- A. 3
- B. 5
- C. 6

**4. l'héritage multiple est un mécanisme dans lequel une classe**

- A. Peut hériter des méthodes et des attributs de plus d'une super-classe
- B. Peut hériter des méthodes et des attributs d'une seule super-classe
- C. Peut hériter des méthodes d'une des classes filles

# QCM (suite)

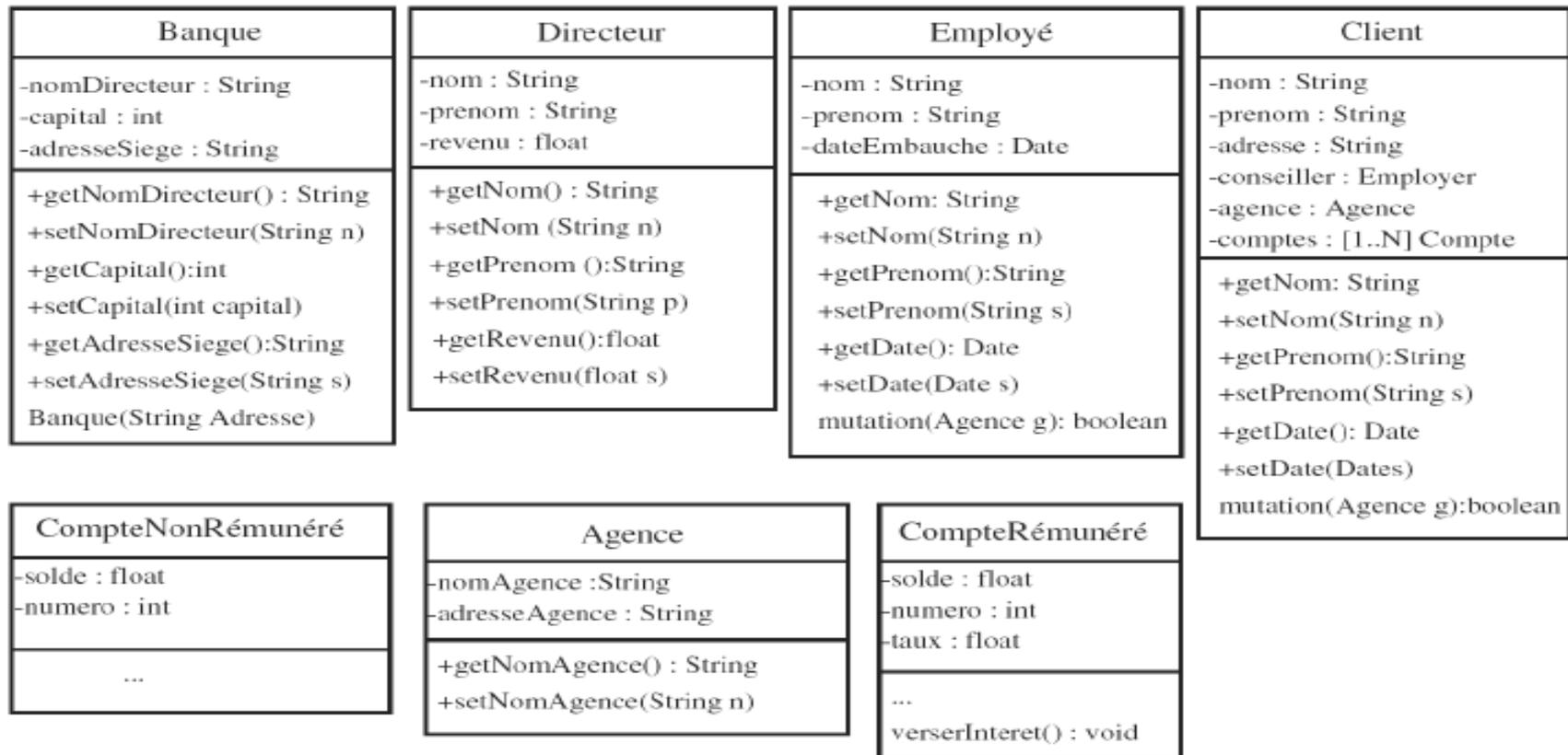
## 5. Dans la redéfinition d'une méthode:

- A. La nom de la méthode est modifiée
- B. Le corps de la méthode est modifié
- C. Le nom et le corps de la méthode sont modifiés

# Exercice 1

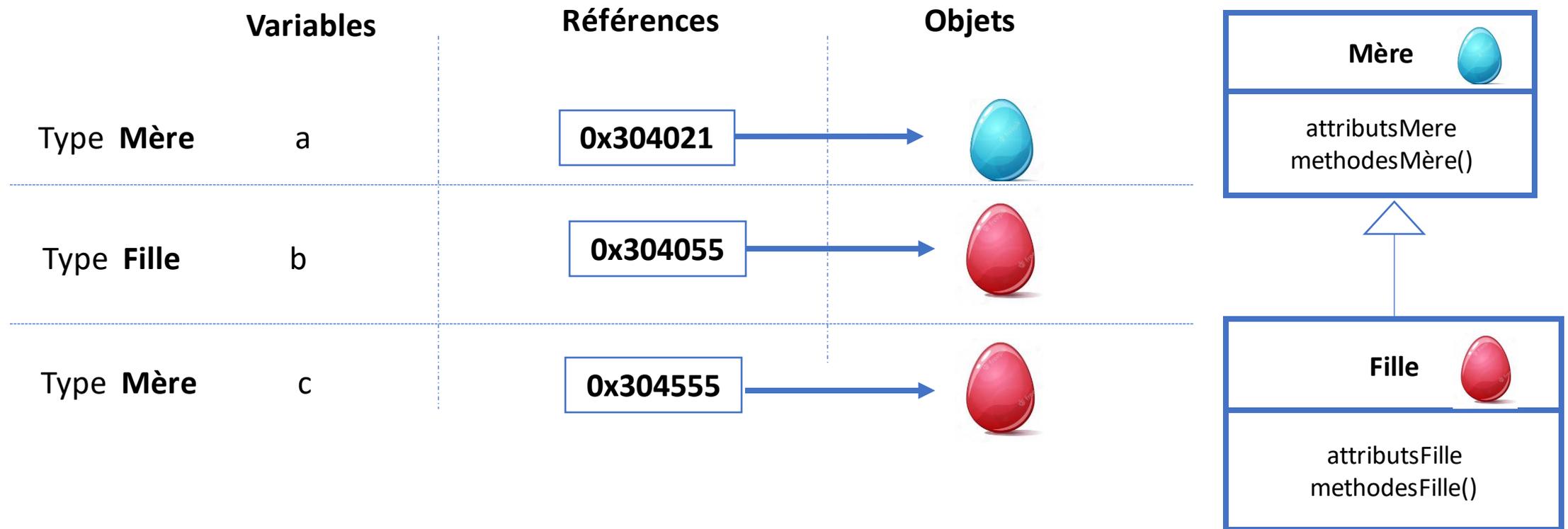
Soient les classes suivantes:

Analysez ces classes et utilisez la généralisation pour factoriser au mieux la description des propriétés.



## Exercice 2

Soient les variables a, b, c suivantes qui font référence à des objets de type classe Mère ou Fille.



## Exercice 2 (suite)

Dites si il est possible d'accéder aux membres des classes Mère et Fille à partir d'une variable. Compléter le tableau par Vrai ou Faux

	a	b	c
attributsMere			
methodesMère()			
attributsFille			
methodesFille()			

# Correction QCM

**1. Quelle assertion se rapproche le plus du concept d'héritage?**

B. C'est une relation de spécialisation entre une classe et ses classes filles.

**2. Qu'est-ce qu'une « super classe »**

A. une classe mère

**B. 3. La classe B hérite de la classe A. Si A possède 3 méthodes et que B en possède 2 qui lui sont propres, combien de méthodes différentes un objet de type B pourra-t-il utiliser?**

B.5

**4. l'héritage multiple est un mécanisme dans lequel une classe**

A. Peut hériter des méthodes et des attributs de plus d'une super-classe

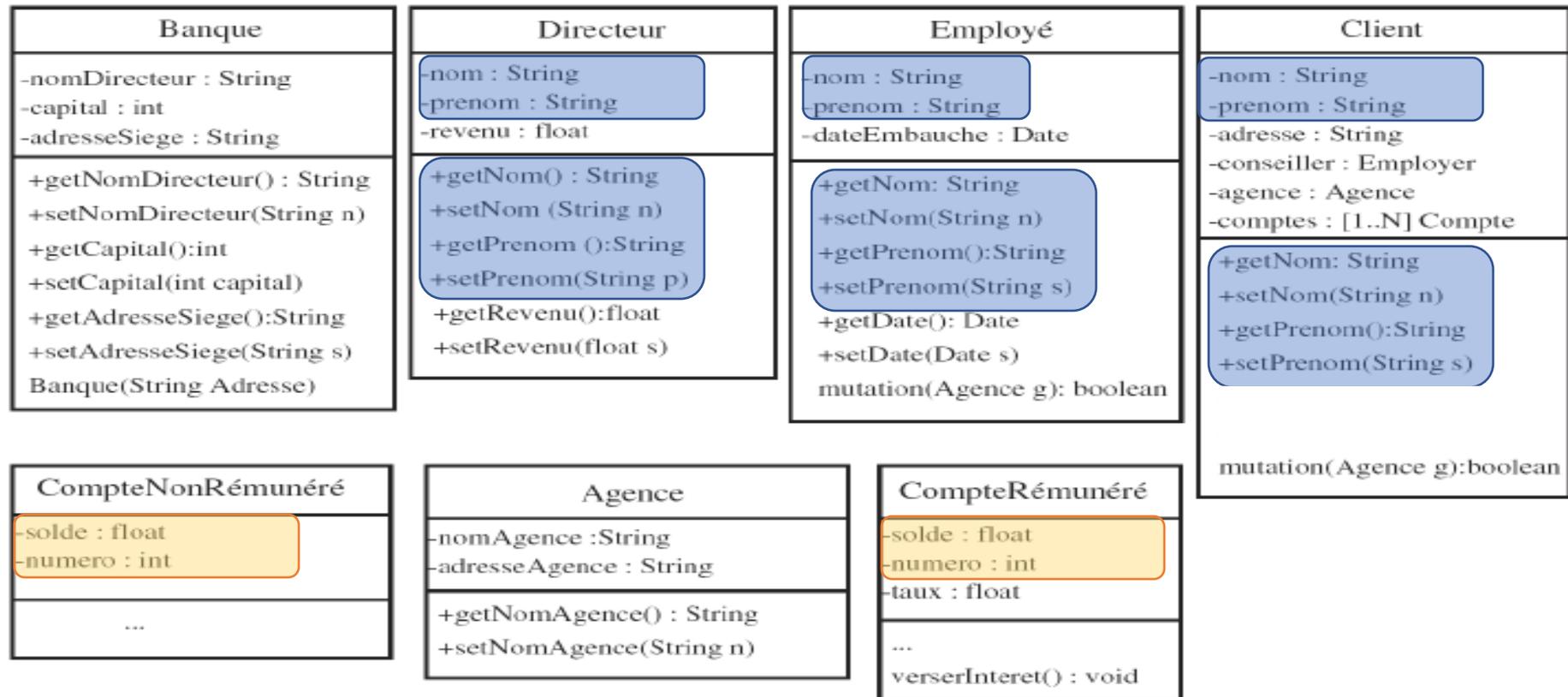
# Correction QCM (suite)

## 5. Dans la redéfinition d'une méthode:

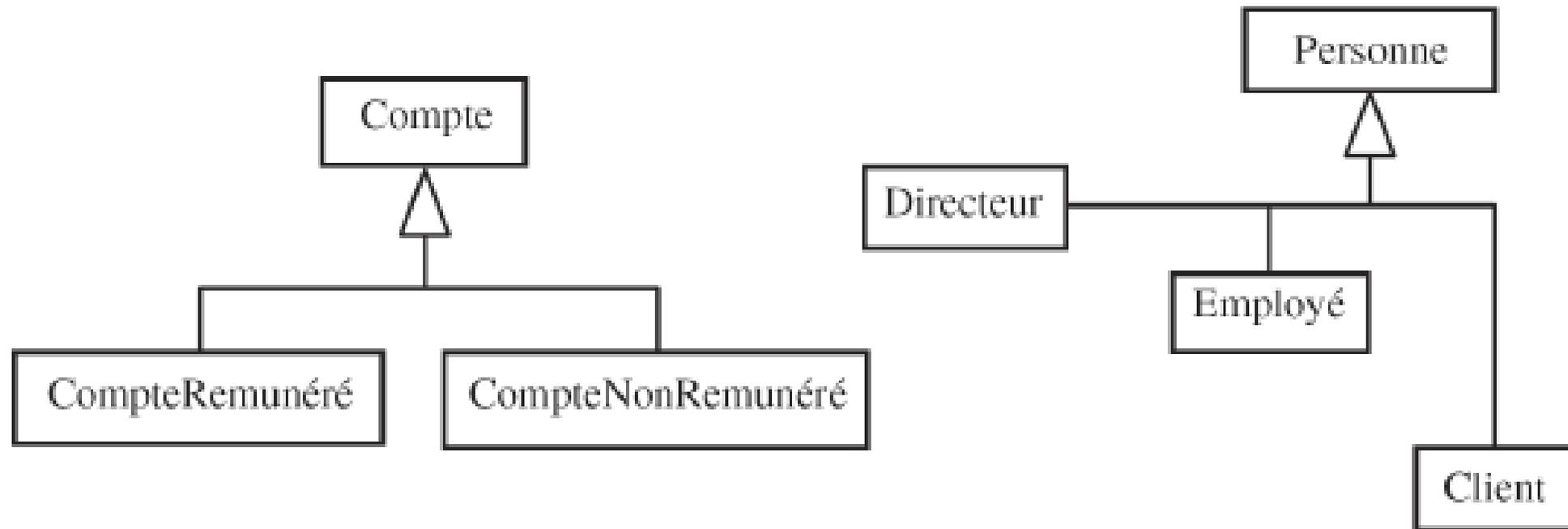
B. Le corps de la méthode est modifié

# Correction Exercice1

1. Demander d'analyser les classes et de dégager les attributs et méthodes communs entre les classes présentées.

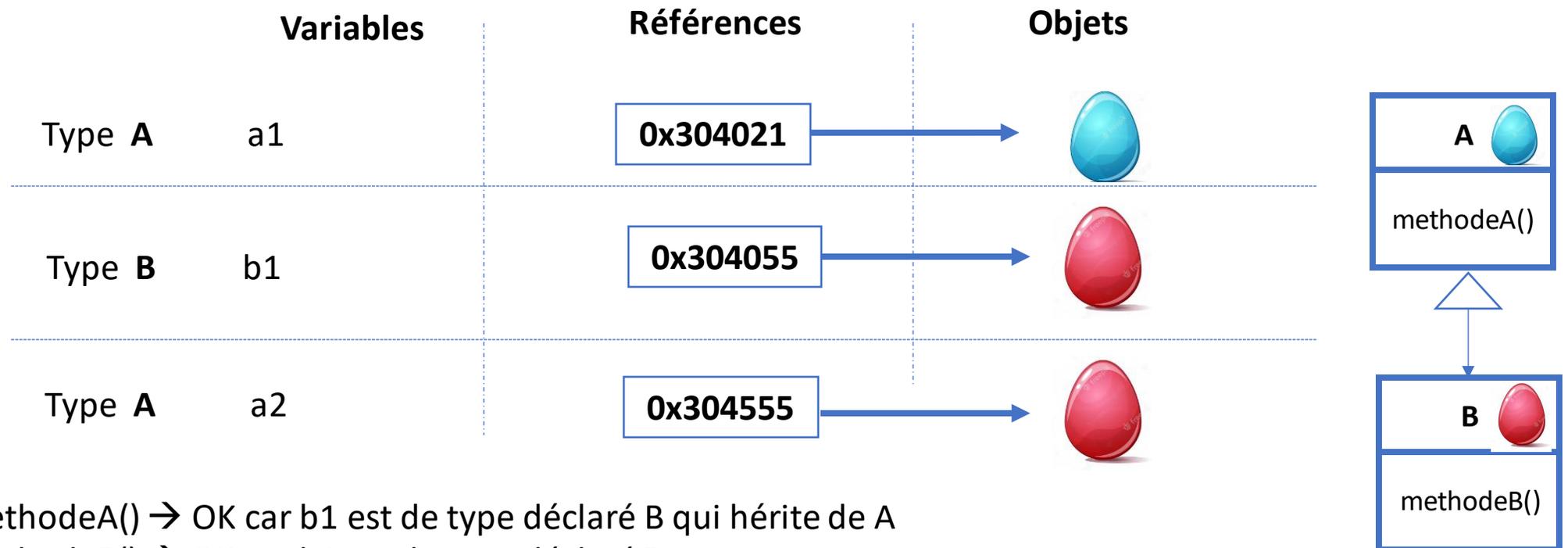


## Correction Exercice1(suite)



# Correction Exercice2

1. Rappelez le principe de polymorphisme: le type de la variable est utilisé par le compilateur pour déterminer si on accède à un membre (attribut ou méthode) valide. Exemple du cours



b1.MethodeA() → OK car b1 est de type déclaré B qui hérite de A

b1.MethodeB() → OK car b1 est de type déclaré B

a2.MethodeA() → OK car a2 est de type déclaré A

**a2.MethodeB() → ERREUR car a2 est de type A (même si le type l'objet référencé est B)**

# Correction Exercice2 (suite)

2. Demander de s'inspirer de l'exemple du cours et de compléter le tableau par Vrai ou Faux

	a	b	c
attributsMere	vrai	vrai	vrai
methodeMère()	vrai	vrai	vrai
attributsFille	faux	vrai	faux
methodeFille()	faux	vrai	faux

# Activité 2

CONNAITRE LES PRINCIPAUX PILIERS DE  
LA POO

**Partie2: Connaitre l'encapsulation**

16

# QCM

**1. Parmi ces qualités, laquelle n'est pas un bénéfice majeur de l'utilisation de l'encapsulation?**

- A. Modularité
- B. Amélioration des performances
- C. Sécurité du code
- D. Facilité d'évolution du code

**2. La portée « protégée » empêche l'accès aux méthodes et attributs qui suivent depuis l'extérieur de la classe, sauf:**

- A. Dans les classe filles
- B. Dans le prog principal
- C. Dans a classe mère

# QCM (suite)

## 3. Quelles méthodes et variables peuvent être utilisées dans une classe héritée ?

- A. publiques ou protégées
- B. Toutes
- C. Privées ou publiques
- D. privées ou protégées

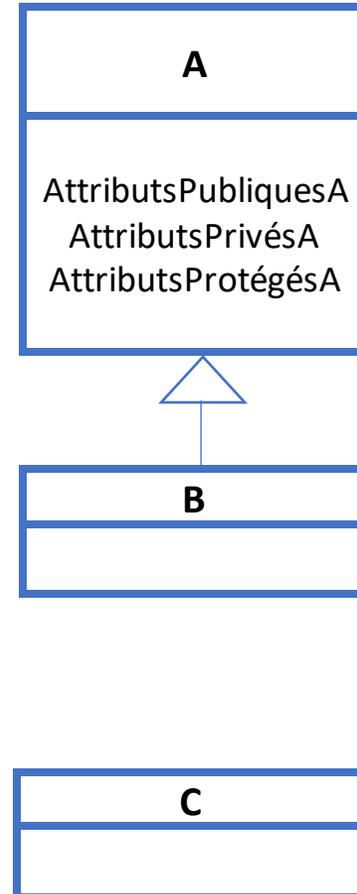
## 4. Comment rendre les attributs et les méthodes accessibles uniquement aux classes filles? En les définissant:

- A. Publiques
- B. Privés
- C. Abstrait
- D. Protégés

# Exercice 1

Soit les classes A, B et C tel que B hérite de A. Compléter par Vrai ou Faux le tableau suivant des affirmations

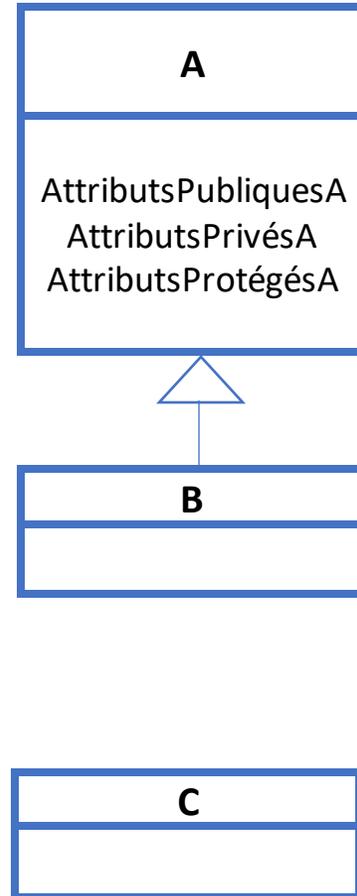
	Affirmation	V/F
1	La classe B hérite de AttributsPubliquesA	
2	Dans la classe B on peut accéder à Dans la classe B on peut accéder à AttributsPubliquesA depuis l'instance courante.	
3	La classe B hérite de AttributsProtégésA	
4	Dans la classe B on peut accéder à AttributsProtégésA depuis l'instance courante	
5	La classe B hérite de AttributsPrivésA	
6	Dans la classe B on peut accéder à AttributsPrivésA depuis l'instance courante.	
7	Dans la classe C on peut accéder à AttributsPubliquesA depuis un référence sur un objet de type A.	
8	Dans la classe C on peut accéder à AttributsProtégésA depuis un référence sur un objet de type	



# Exercice 1

Soit les classes A, B et C tel que B hérite de A. Compléter par Vrai ou Faux le tableau suivant des affirmations

	Affirmation	V/F
9	Dans la classe C on peut accéder à <code>AttributsPrivésA</code> depuis un référence sur un objet de type A.	
10	Dans la classe C on peut accéder à <code>AttributsPubliquesA</code> depuis un référence sur un objet de type B.	
11	Dans la classe C on peut accéder à <code>AttributsProtégésA</code> depuis un référence sur un objet de type	
12	Dans la classe C on peut accéder à <code>AttributsPrivésA</code> depuis un référence sur un objet de type B.	



# Correction QCM

**1. Parmi ces qualités, laquelle n'est pas un bénéfice majeur de l'utilisation de l'encapsulation?**

B. Amélioration des performances

**2. La portée « protégée » empêche l'accès aux méthodes et attributs qui suivent depuis l'extérieur de la classe, sauf:**

A. Dans les classe filles

**3. Quelles méthodes et variables peuvent être utilisées dans une classe héritée ?**

A. publiques ou protégées

**4. Comment rendre les attributs et les méthodes accessibles uniquement aux classes filles en les définissant?**

D. Protégés

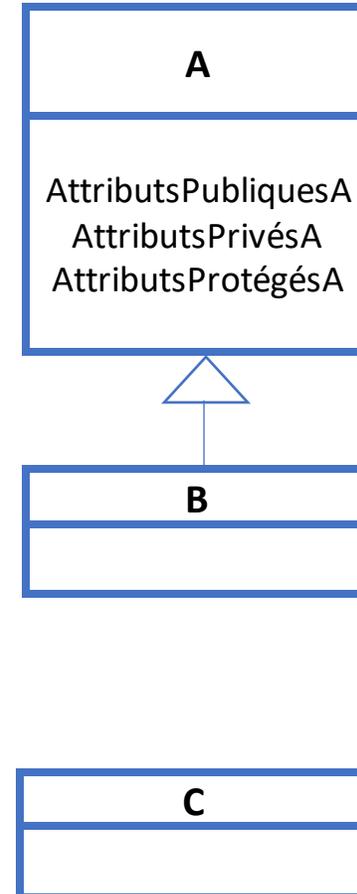
# Correction Exercice 1

1. Rappelez les règles de visibilité des membres des classes vues précédemment.
2. Résumez ces règles dans le tableau suivant:

Accès :	même classe	Classes dérivées dans le même package	Classes du même package	Classes dérivées dans un autre package	Classes des autres packages
<b>public</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>protected</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	
<b>private</b>	<b>X</b>				

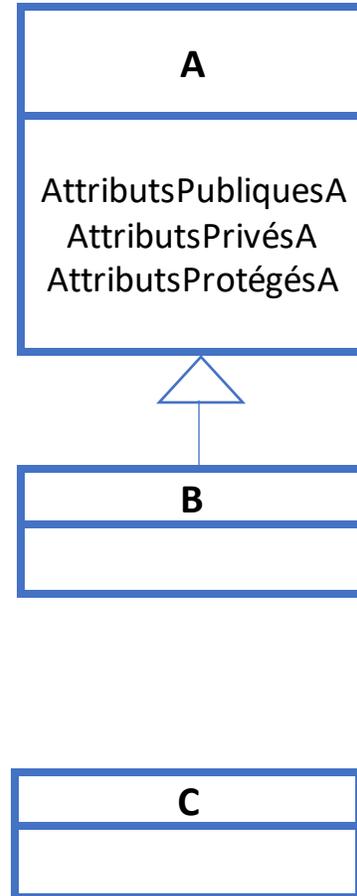
# Correction Exercice 1 (suite)

	Affirmation	V/F
1	La classe B hérite de AttributsPubliquesA	V
2	Dans la classe B on peut accéder à AttributsPubliquesA depuis l'instance courante.	V
3	La classe B hérite de AttributsProtégésA	V
4	Dans la classe B on peut accéder à AttributsProtégésA depuis l'instance courante	V
5	La classe B hérite de AttributsPrivésA	V
6	Dans la classe B on peut accéder à AttributsPrivésA depuis l'instance courante.	F
7	Dans la classe C on peut accéder à AttributsPubliquesA depuis un référence sur un objet de type A.	V
8	Dans la classe C on peut accéder à AttributsProtégésA depuis un référence sur un objet de type	F



# Correction Exercice 1(suite)

	Affirmation	V/F
9	Dans la classe C on peut accéder à AttributsPrivésA depuis un référence sur un objet de type A.	F
10	Dans la classe C on peut accéder à AttributsPubliquesA depuis un référence sur un objet de type B.	V
11	Dans la classe C on peut accéder à AttributsProtégésA depuis un référence sur un objet de type B	F
12	Dans la classe C on peut accéder à AttributsPrivésA depuis un référence sur un objet de type B.	F



# Activité 2

CONNAITRE LES PRINCIPAUX PILIERS DE  
LA POO

**Partie3: Caractériser l'abstraction**

25

# QCM

**1. Pour factoriser des attributs et méthodes communes à plusieurs classes filles, on utilise une classe:**

- A. protégée
- B. abstraite
- C. Concrète
- D. Publique

**2. Est-il possible de créer un objet à partir d'une classe abstraite ?**

- A. Oui toujours
- B. Non
- C. Oui, si elle est publique

## QCM (suite)

**3. Quelle est l'assertion qui correspond le mieux à la définition d'une interface?**

- A. C'est une méthode vide d'une classe.
- B. C'est un moyen donné à l'utilisateur d'une application OO d'interagir avec celle-ci.
- C. C'est l'ensemble des signatures d'opérations publiques d'une classe.
- D. C'est l'ensemble des attributs publics d'une classe.

**4. Une interface peut-elle être instanciée ?**

- A. oui si elle est publique
- B. oui si elle n'est pas privée
- C. Non

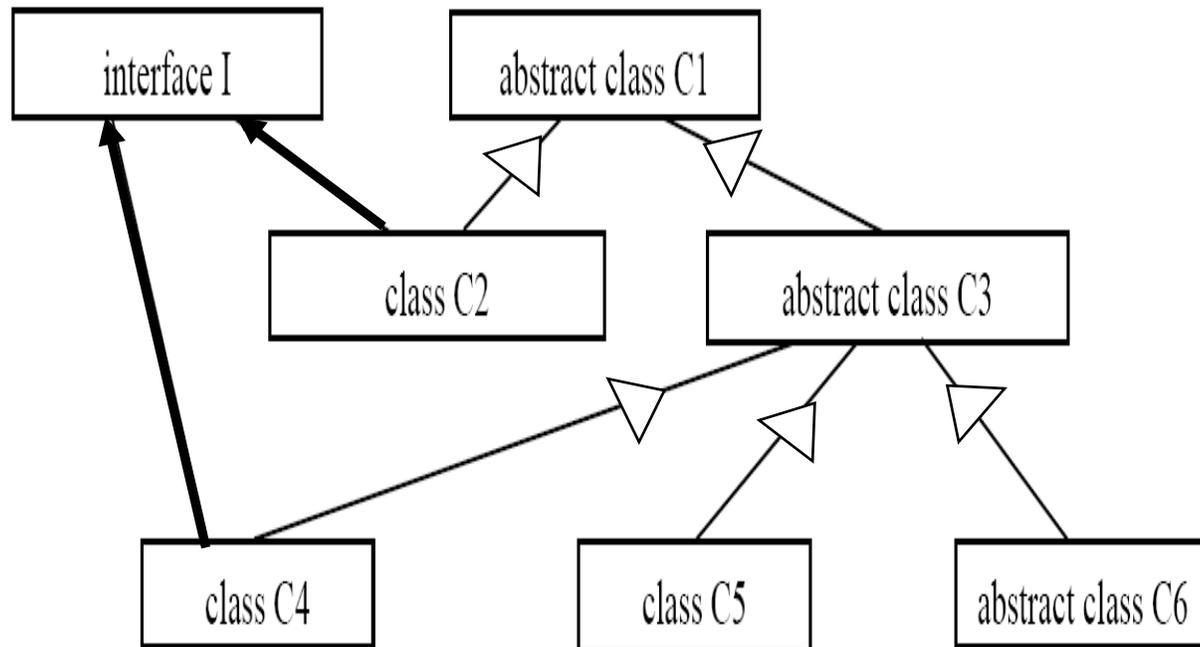
# QCM (suite)

## 4. Qu'est-ce qui est faux pour les interfaces ?

- A. Une Interface peut être le type d'une référence
- B. Une Interface déclare des méthodes sans les implémenter
- C. Une Interface peut être implémentée
- D. Une Interface peut être instanciée

# Exercice 1

On vous demande d'analyser chacun des cas suivants d'une manière indépendante des autres. Pour chaque cas, dites s'il y a des instructions erronées.



## Cas 1 :

- X de type C1 fait reference à un objet de type C4
- Y de type C1 fait reference à un objet de type C5
- X=Y

## Cas 2 :

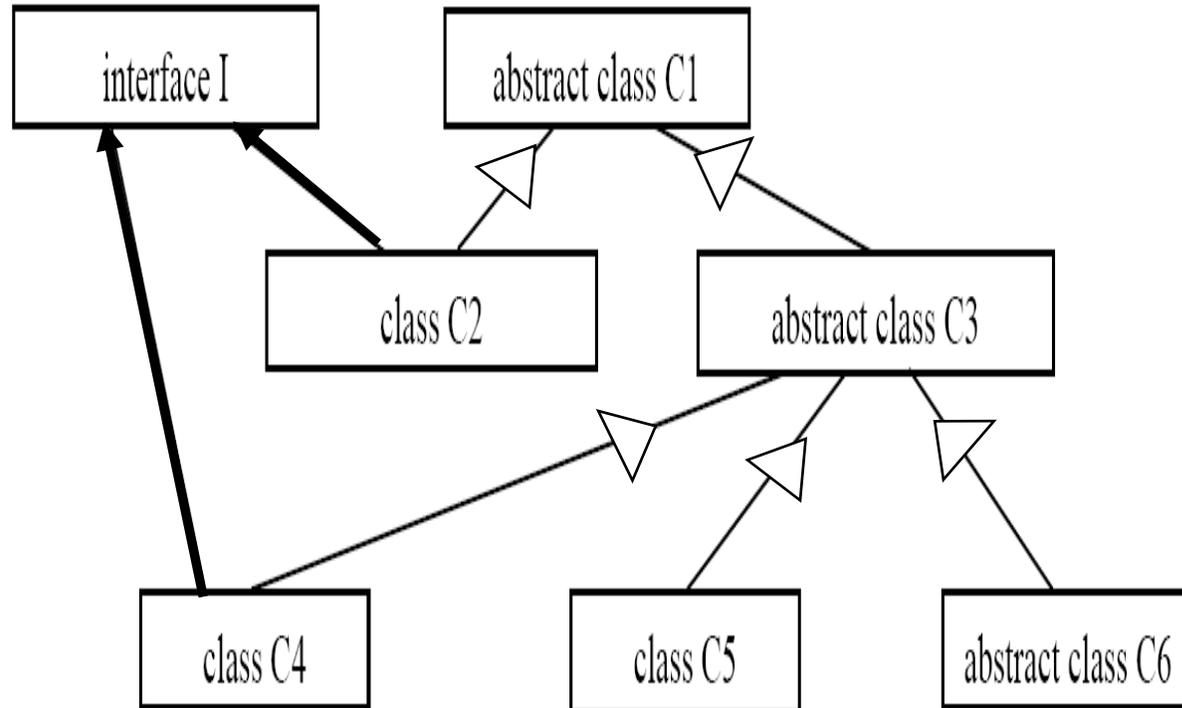
- X de type I fait reference à un objet de type C4
- Y de type C3 fait reference à un objet de type C4
- X=Y

## Cas 4 :

- X de type C3 fait reference à un objet de type C5
- Y de type C6 fait reference à un objet de type C6
- X=Y

## Exercice 1 (suite)

On vous demande d'analyser chacun des cas suivants d'une manière indépendante des autres. Pour chaque cas, dites s'il y a des instructions erronées.



### Cas 4 :

- X de type I fait reference à un objet de type C4
- Y de type I fait reference à un objet de type C2
- X=Y

### Cas 5:

- X de type C1 fait reference à un objet de type C2
- Y de type I fait reference à un objet de type C2
- X=Y

### Cas 6 :

- X de type C3 fait reference à un objet de type C5
- Y de type C5 fait reference à un objet de type C5
- X=Y

# Correction QCM

**1. Pour factoriser des attributs et méthodes communes à plusieurs classes filles, on utilise une classe:**

B. abstraite

**2. Est-il possible de créer un objet à partir d'une classe abstraite ?**

B. Non

**3. Quelle est l'assertion qui correspond le mieux à la définition d'une interface?**

C. C'est l'ensemble des signatures d'opérations publiques d'une classe.

**4. Une interface peut-elle être instanciée ?**

C. Non

# Correction QCM (suite)

## 4. Qu'est-ce qui est faux pour les interfaces ?

D. Une Interface peut être instanciée

# Correction Exercice 1

## 1. Rappeler les notions suivantes vues dans le cours:

- Une classe abstraite n'est pas instanciable mais elle peut être le type d'une variable
- Une interface n'est pas instanciable mais elle peut être le type d'une variable
- Si on a 2 variables a et b, l'instruction a=b est correcte si le type de a est plus générique que le type de b
- Une variable de type une classe de Base peut référencer un objet d'une classe Dérivée

## 2. Demander d'appliquer ces règles pour faire l'exercice

# Correction Exercice 1 (suite)

## Cas 1 : **correcte**

- X de type C1 fait reference à un objet de type C4 → correcte
- Y de type C1 fait reference à un objet de type C5 → correcte
- X=Y → correcte

## Cas 2 : **Faux**

- X de type I fait reference à un objet de type C4 → correcte
- Y de type C3 fait reference à un objet de type C4 → correcte
- X=Y → fausse (car **C3 n'implémente pas I**)

## Cas 3 : **Faux**

- X de type C3 fait reference à un objet de type C5 → correcte
- Y de type C6 fait reference à un objet de type C6 → fausse (**C6 est abstraite non instanciable**)
- X=Y

# Correction Exercice 1 (suite)

## Cas 4 : **correcte**

- X de type I fait reference à un objet de type C4 → correcte
- Y de type I fait reference à un objet de type C2 → correcte
- X=Y → correcte

## Cas 5: **faux**

- X de type C1 fait reference à un objet de type C2 → correcte
- Y de type I fait reference à un objet de type C2 → correcte
- X=Y → fausse (**C2 implémente I pas le contraire**)

## Cas 6 : **faux**

- X de type C3 fait reference à un objet de type C5 → correcte
- Y de type C5 fait reference à un objet de type C5 → correcte
- Y=X → fausse (**C3 classe mère de C5 pas le contraire**)