CHAPITRE 2

Définir le polymorphisme

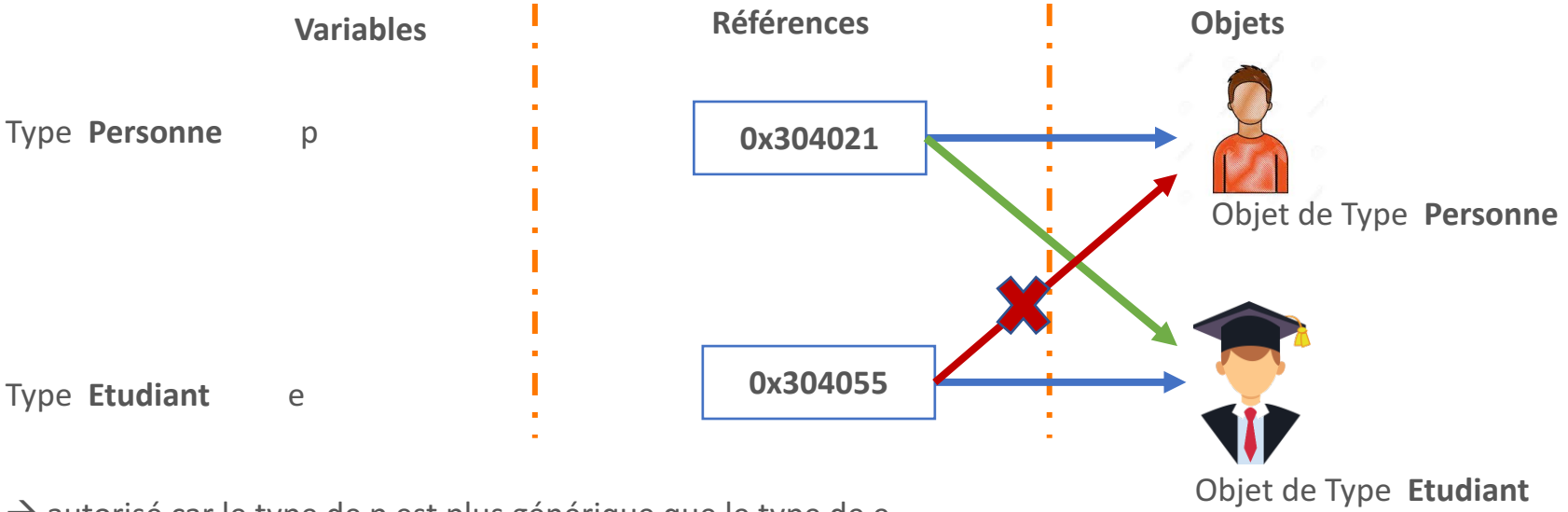
1. **Principe du polymorphisme**
2. Redéfinition des méthodes
3. Surcharge des méthodes

Principe de polymorphisme

- Le polymorphisme désigne un concept de la théorie des types, selon lequel un nom d'objet peut désigner des instances de classes différentes issues d'une même arborescence.

Exemple :

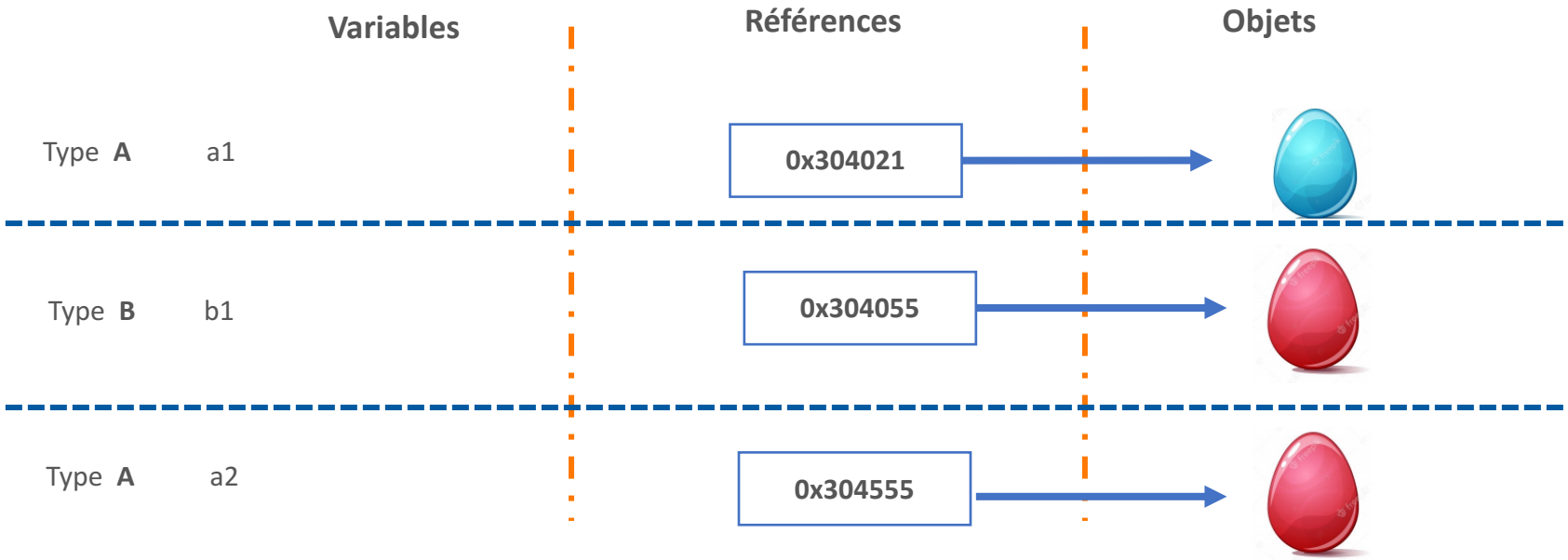
- Une instance de Etudiant peut « être vue comme » une instance de Personne **(Pas l'inverse!!)**

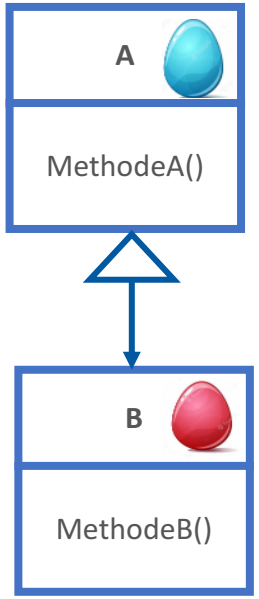


- = e → autorisé car le type de p est plus générique que le type de e
- e = p→ non autorisé car le type de e est plus spécifique que celui de p

PARTIE 2

- Le type de la variable est utilisé par le compilateur pour déterminer si on accède à un membre (attribut ou méthode) valide





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)

PARTIE 2

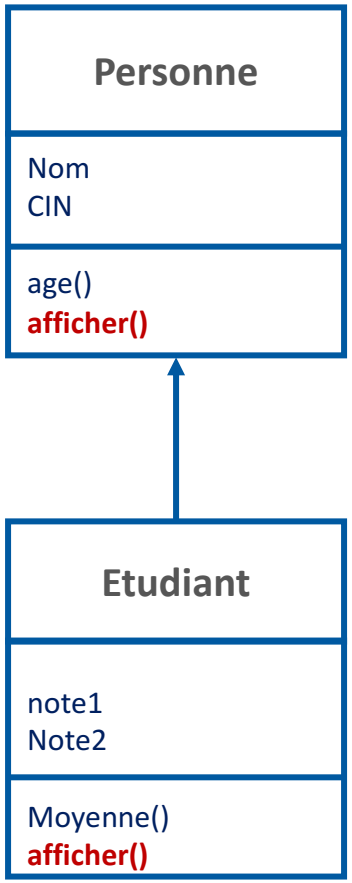
CHAPITRE 2

Définir le polymorphisme

1. Principe du polymorphisme
2. **Redéfinition des méthodes**
3. Surcharge des méthodes

Redéfinition des méthodes héritées

- La redéfinition d'une méthode héritée est motivée par le fait que sa version, dans la classe mère, ne correspond pas aux besoins de la classe fille.
- La redéfinition permet alors de proposer **un code différent** à une méthode héritée tout en **gardant son entête**. Autrement, il ne peut s'agir d'une redéfinition mais d'une nouvelle méthode complétement différente de celle héritée.
- A partir d'une classe fille, il est possible, à tout moment, d'invoquer une méthode redéfinie, dans sa version initiale, déclarée dans la classe mère.



Exemple :

- Soit la classe Etudiant qui hérite de la classe Personne
- La méthode afficher() de la classe Personne affiche les attributs Nom, CIN d'une personne
- La classe Etudiant hérite la méthode afficher() de la classe Personne et
- la **redéfinit** , elle propose un nouveau code (la classe Etudiant **ajoute** l'affichage
- des attributs notes1 et note2 de l'étudiant)

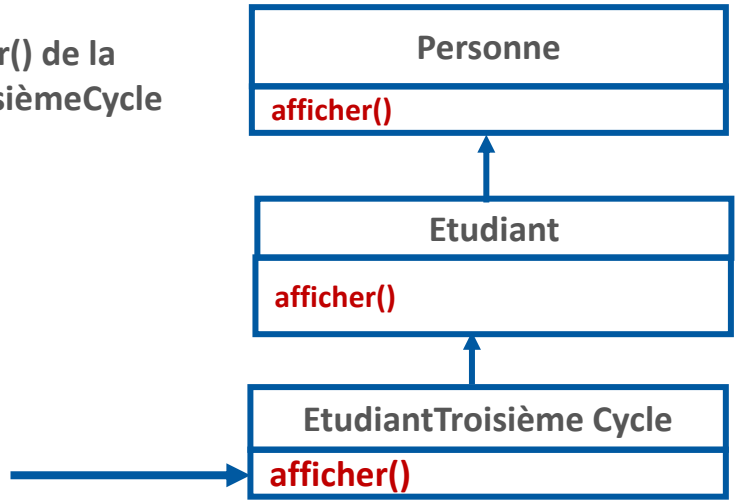
Mécanisme de la liaison retardée

- Soit 3 classes A, B et C, B hérite de A et C hérite de B.
- Soit o une instance (objet) de la classe A. On désire invoquer la méthode m() à partir de o, comme suit : **« o.m() »**
- o peut appeler m() qui représente la méthode appartenant à la classe A. La méthode peut être appelée par les classes filles de A à savoir B et C.
- Supposons que la classe C définit ou redéfinit la méthode m(). Si un objet de la classe C fait appel à la méthode m(), c'est celle de la classe C qui sera exécutée, en cas d'échec la recherche se poursuit au niveau de la classe B puis A et ainsi de suite.

Exemple : Soit etc un objet de type EtudiantTroisièmeCycle

1er cas

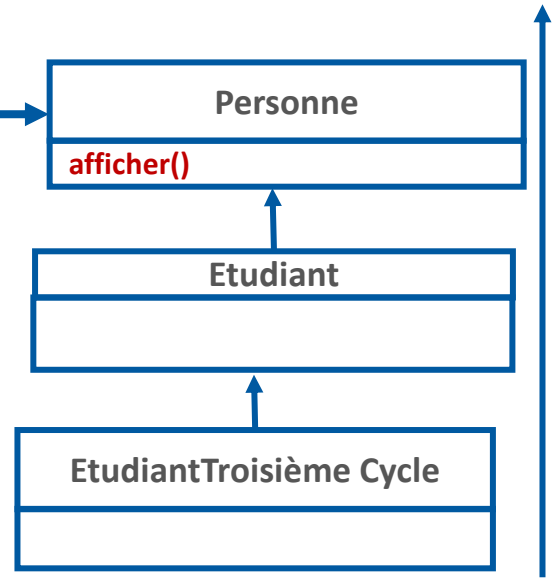
La méthode afficher() de la classe EtudiantTroisièmeCycle sera exécutée



2ème cas

etc.afficher()

La méthode afficher() de la classe Personne est exécutée



PARTIE 2

CHAPITRE 2

Définir le polymorphisme

1. Principe du polymorphisme
2. Redéfinition des méthodes
3. **Surcharge des méthodes**

- La surcharge d'une méthode permet de **définir plusieurs fois une même méthode** avec **des arguments différents**.
- Le compilateur choisi la méthode qui doit être appelée en fonction du nombre et du type des arguments.
- Une méthode est surchargée lorsqu'elle exécute des actions différentes selon le type et le nombre de paramètres transmis.