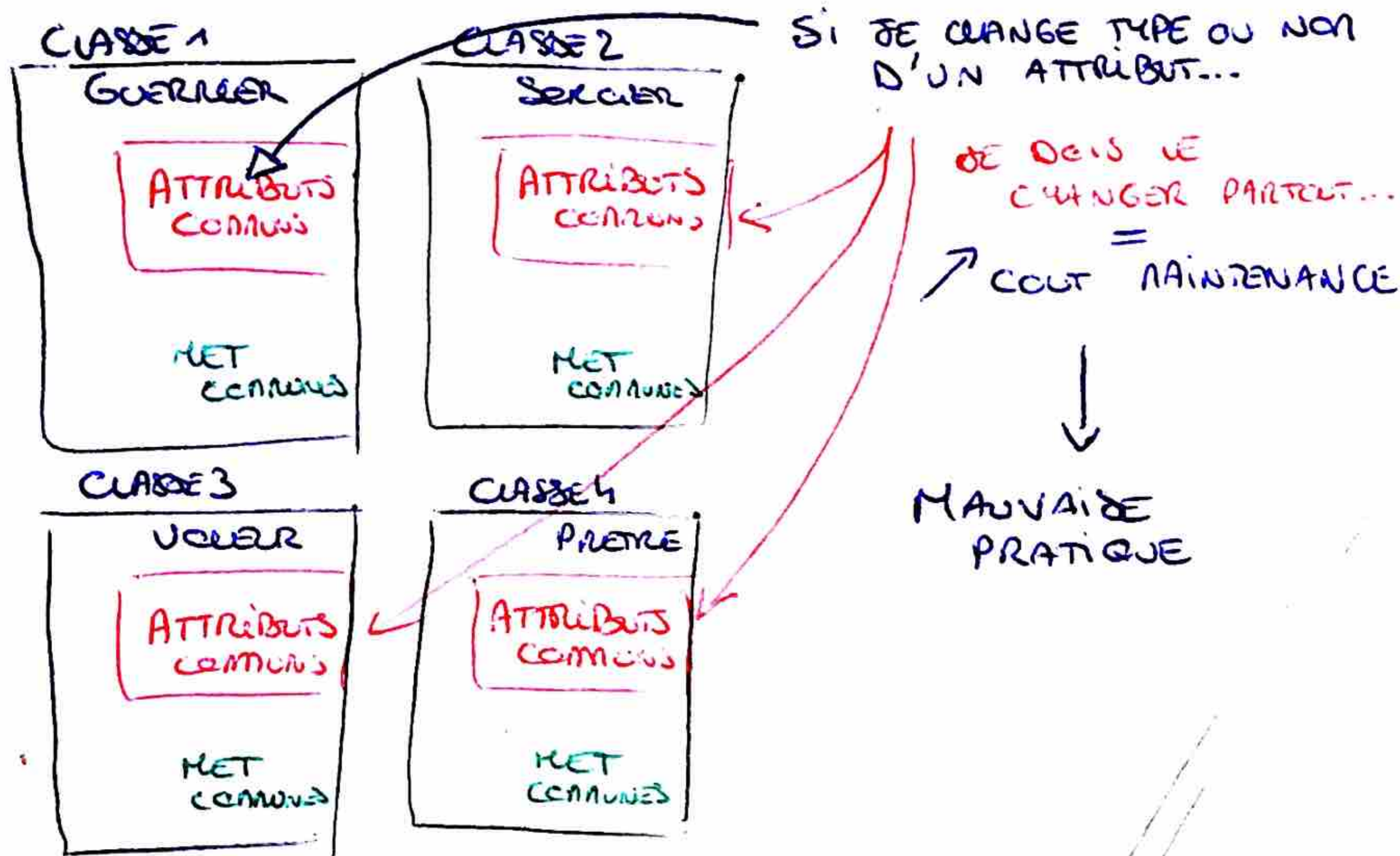
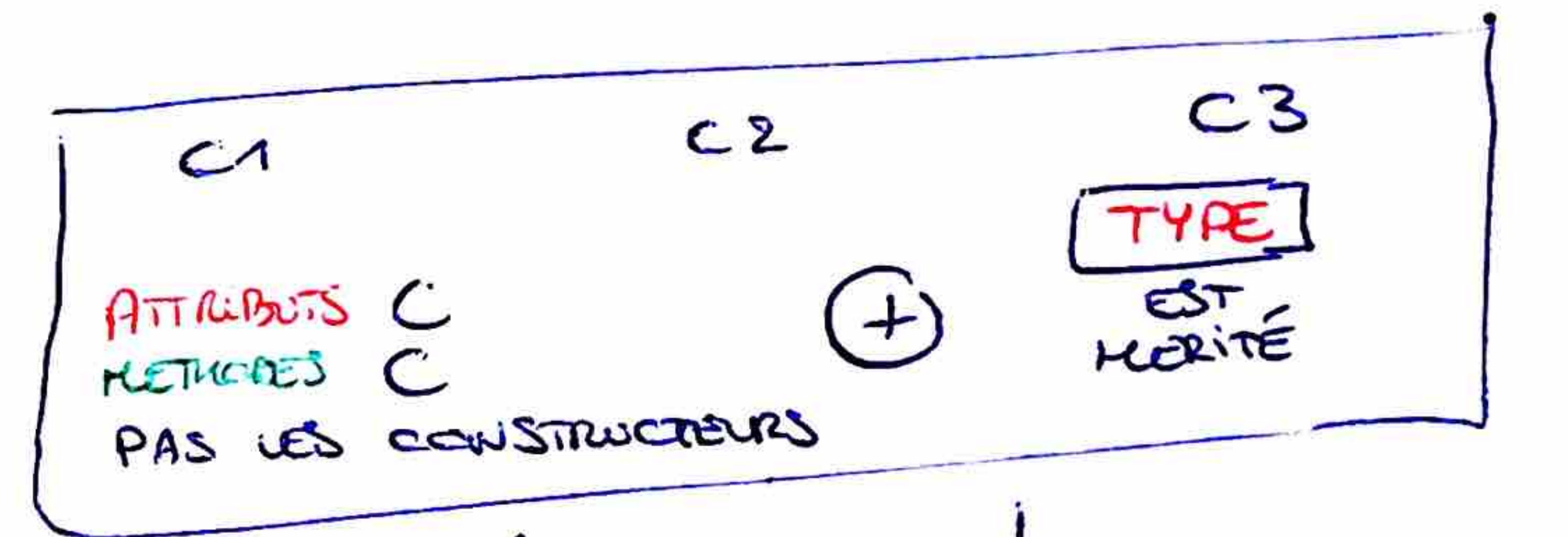


HERITAGE



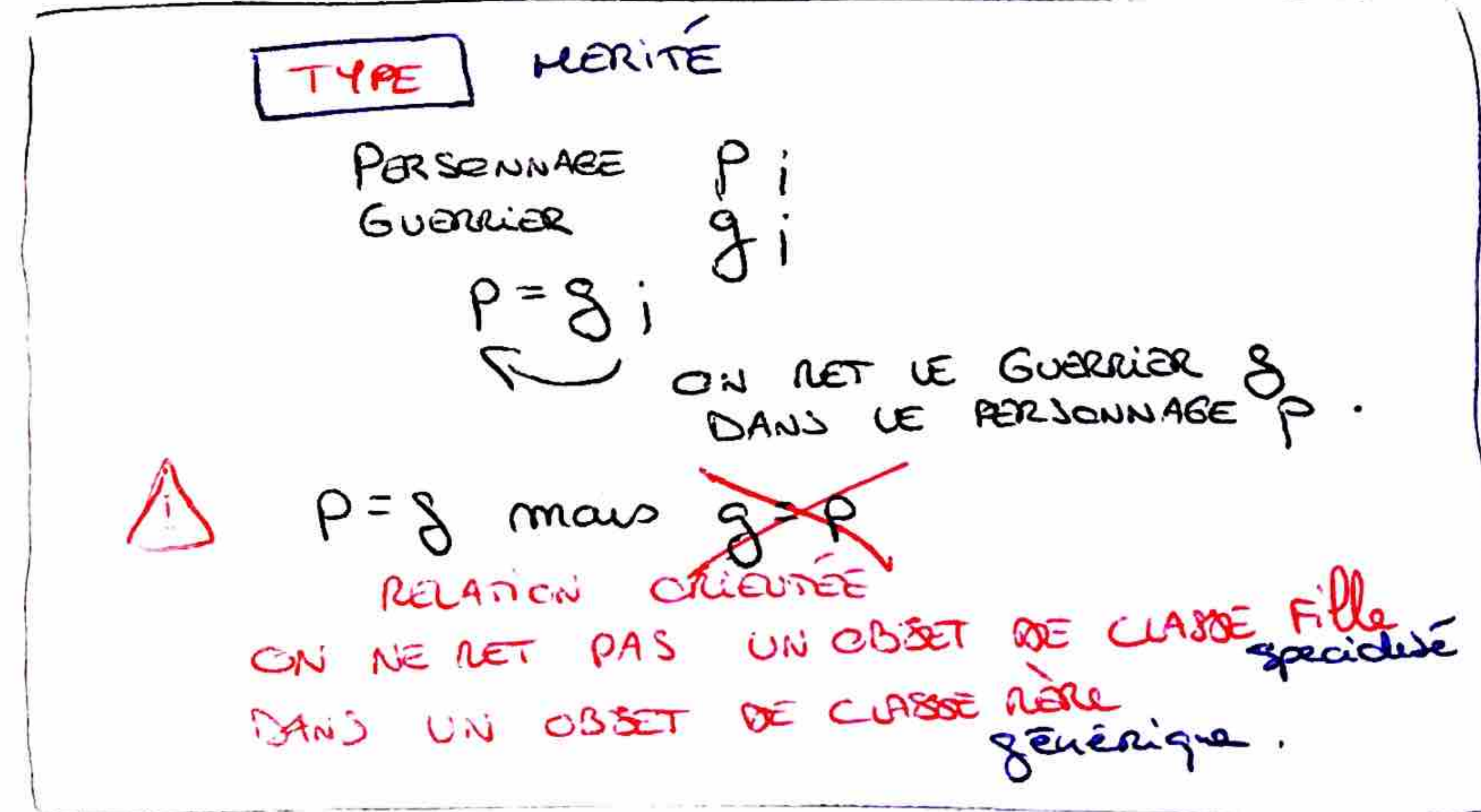
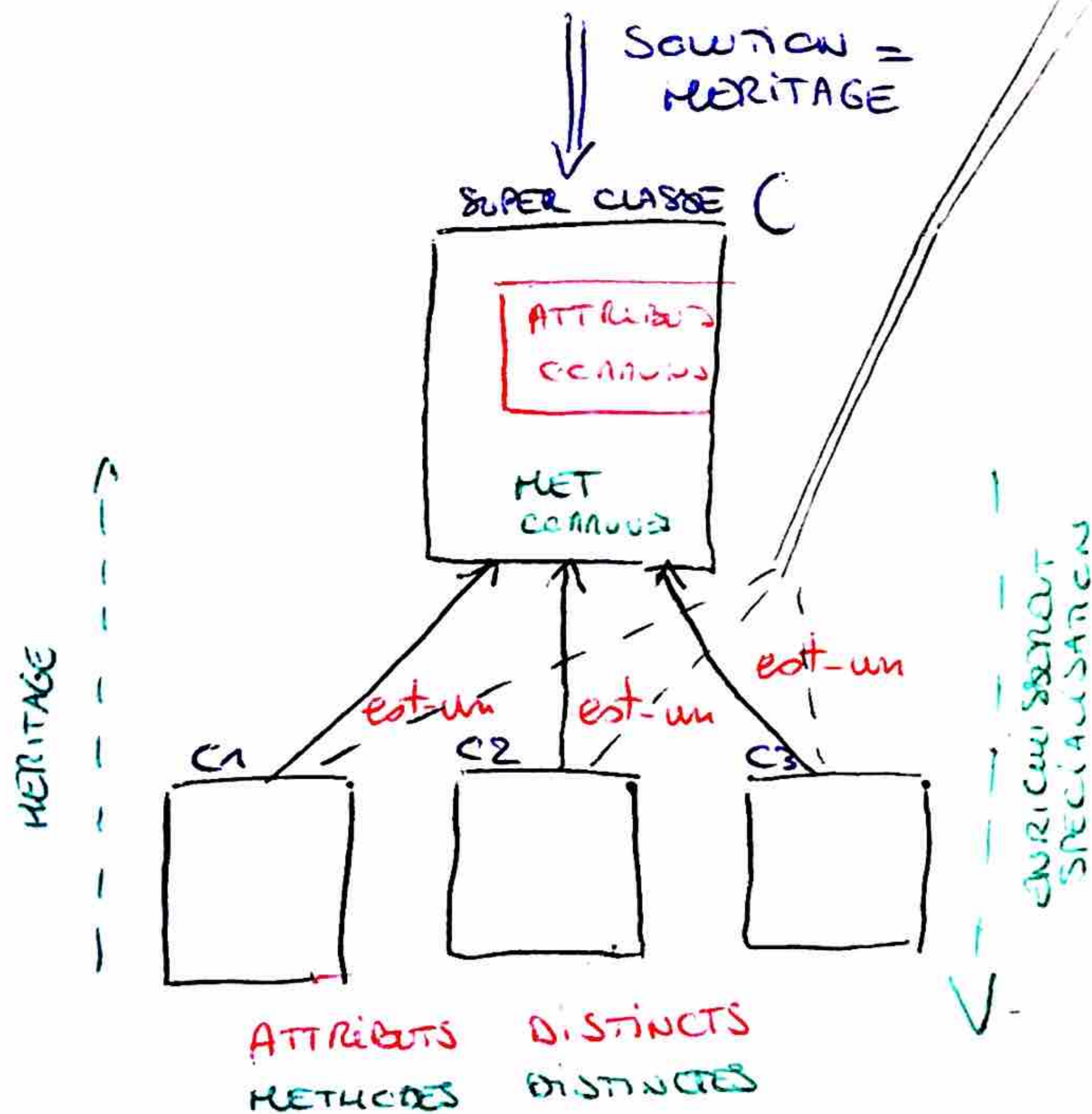
QU'EST CE QUI EST HÉRITÉ ?



Si A a des ATTRIBUTS METHODES
= ENRICHISSEMENT

Si REDEFINITION METHODES
= SPECIALISATION

SOLUTION = HERITAGE



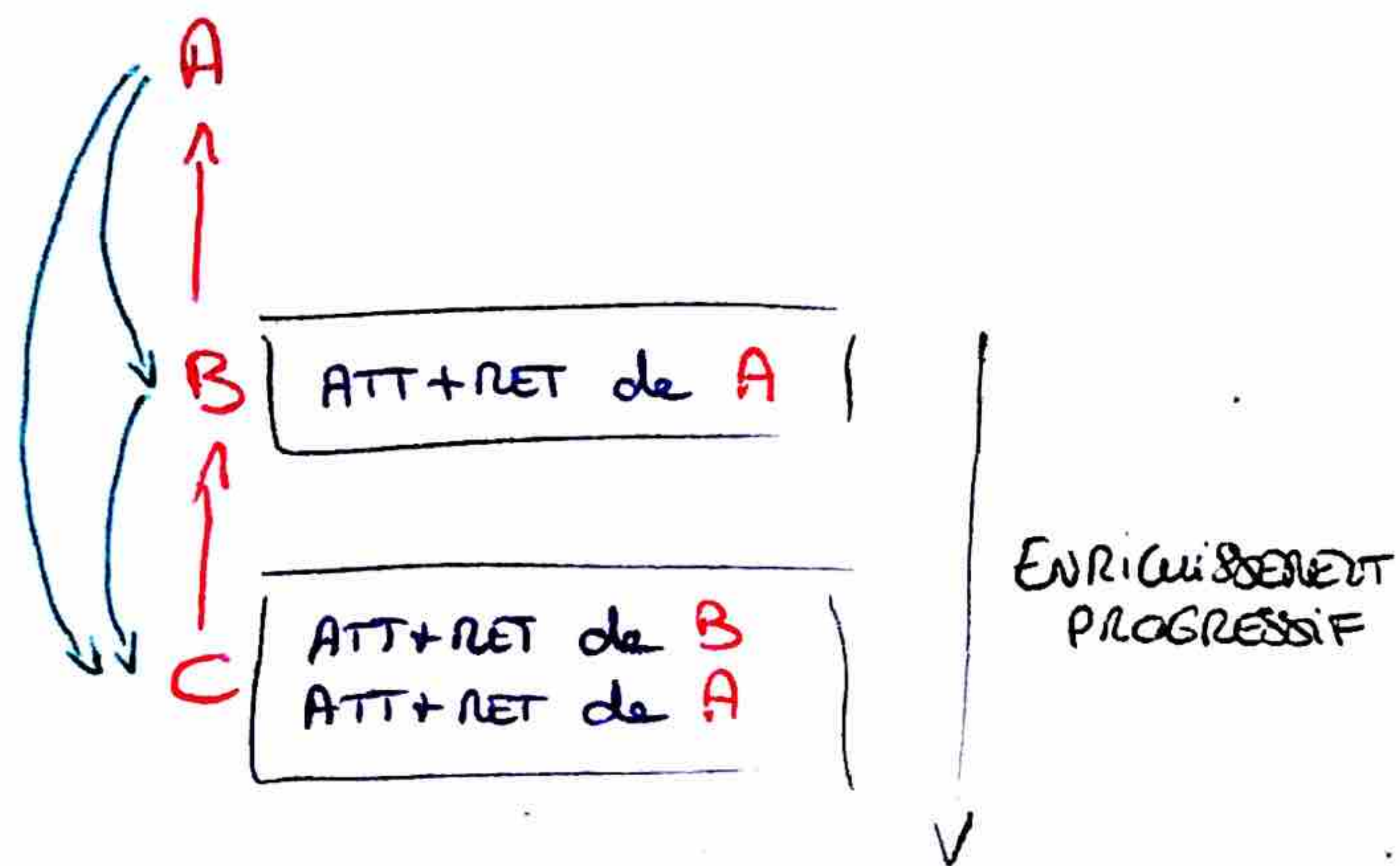
CLASSE FILLES

ON PEUT APPeler LES METHODES DE LA CLASSE MÈRE (PUBLICS).
ON PEUT APPeler LES ATTRIBUTS (MAIS VIA L'INTERFACE ! => GETTERS PUBLICS)

⚠ L'HERITAGE DÉCRIT UNE RELATION **est-un** is-a
NE DOIT JAMAIS DÉCRIRE UNE RELATION **a-un** has-a

ENCAPSULATION POUR DES ATTRIBUTS PROPRES A SA CLASSE

TRANSITIVITÉ DE L'HERITAGE



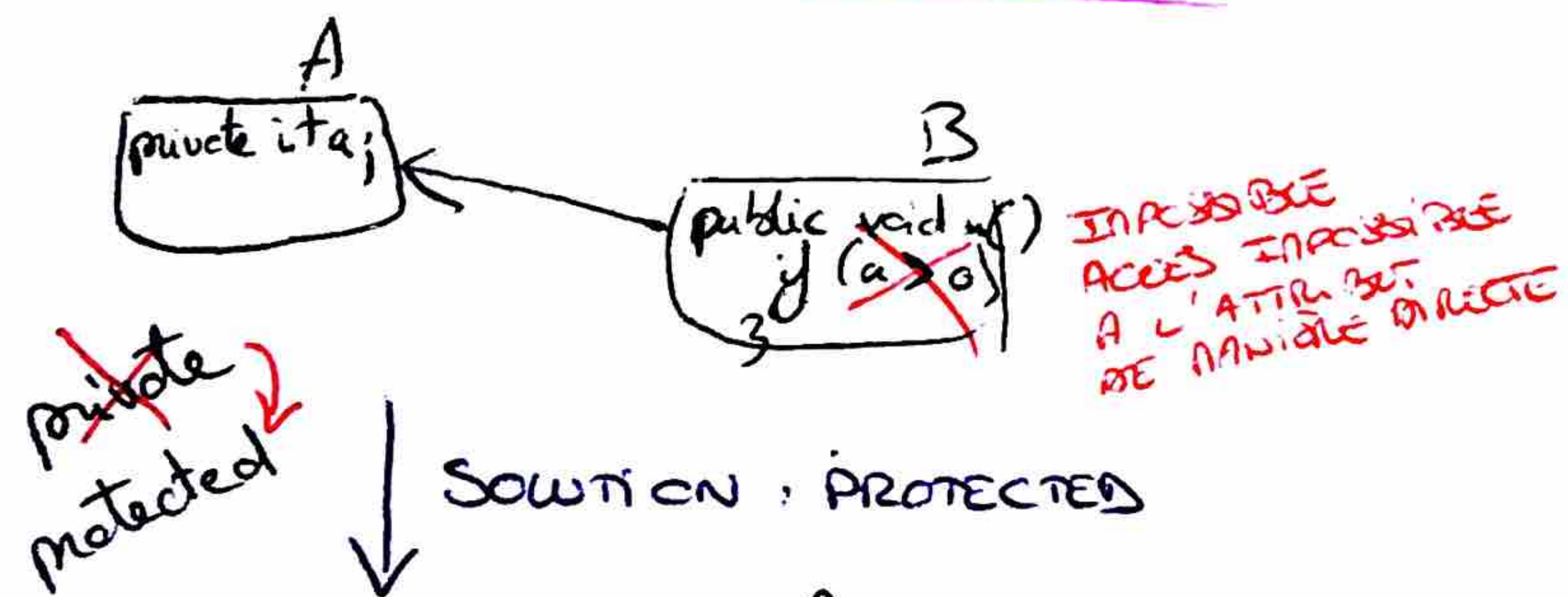
SYNTAXE DU CODE

class C1 extends C

```
{
    private double z
    private double y
}
```

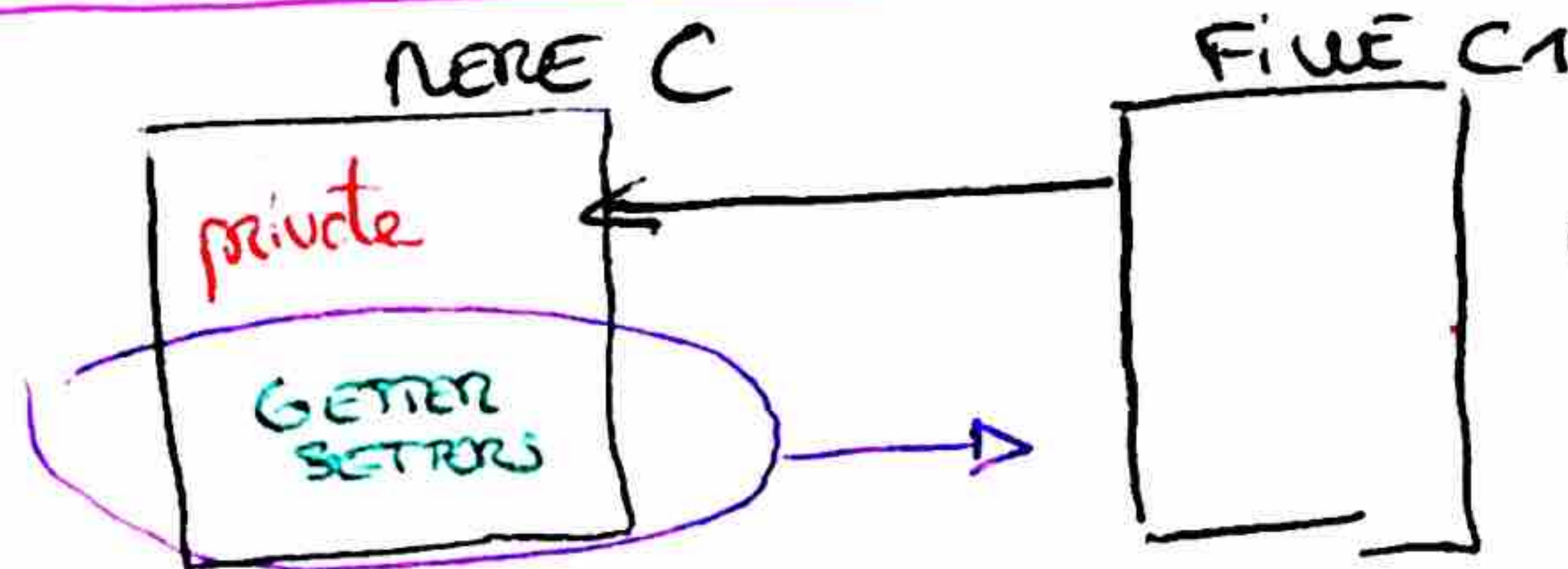
ENRICHISSEMENT

HERITAGE (DROITS ACCES PROTECTED)



REND LES ATTRIBUTS DANS LES CLASSES FIRES B VISIBLES + DES AUTRES CLASSES DU MÊME PACKAGE

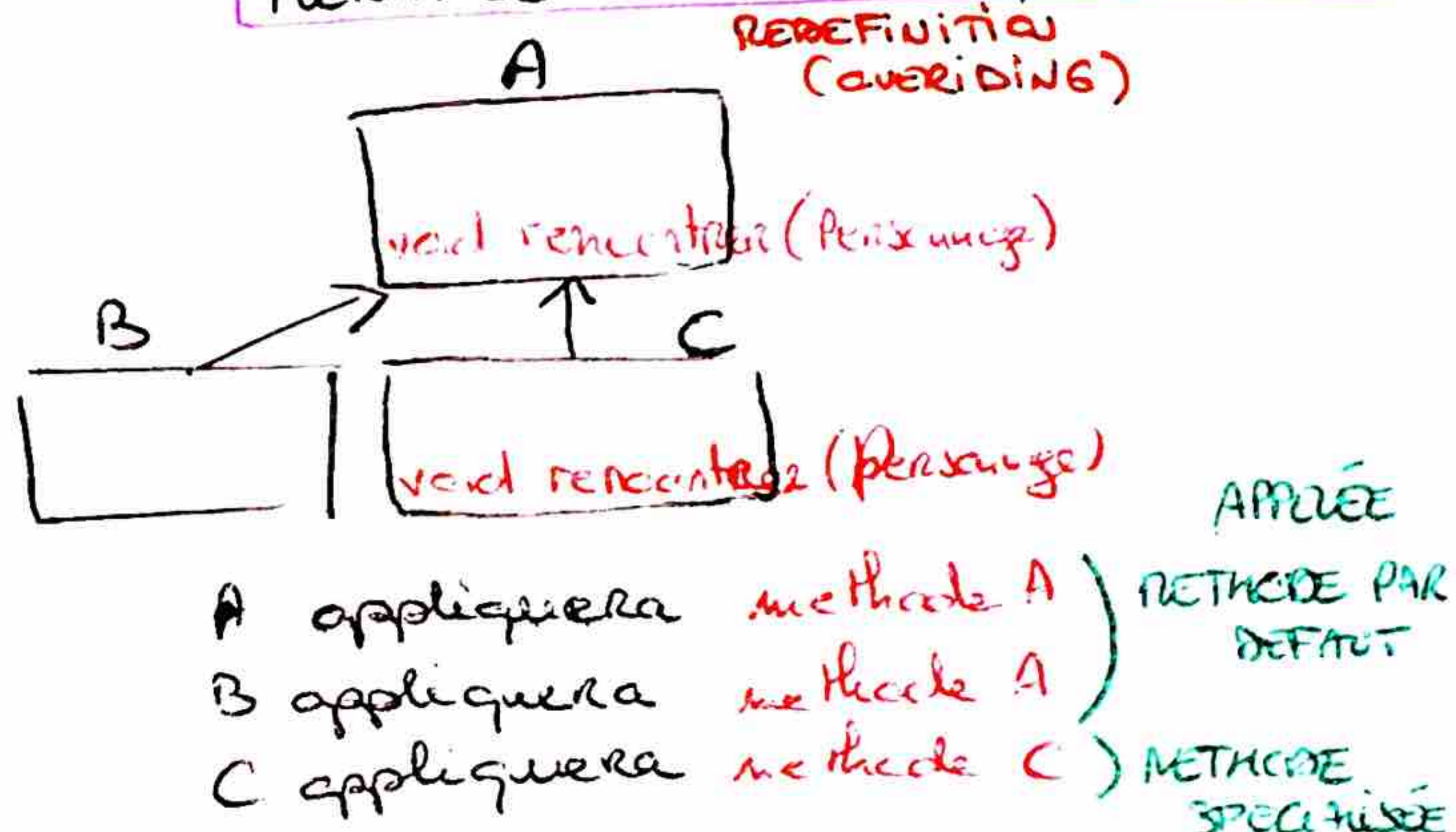
ACCES PROTEGE : BONNE PRATIQUE



UTILISER LES GETTERS / SETTERS DANS LES CLASSES FIRES.
(INTERFACE A BIEN DEFINIR POUR LA CLASSE PERE)

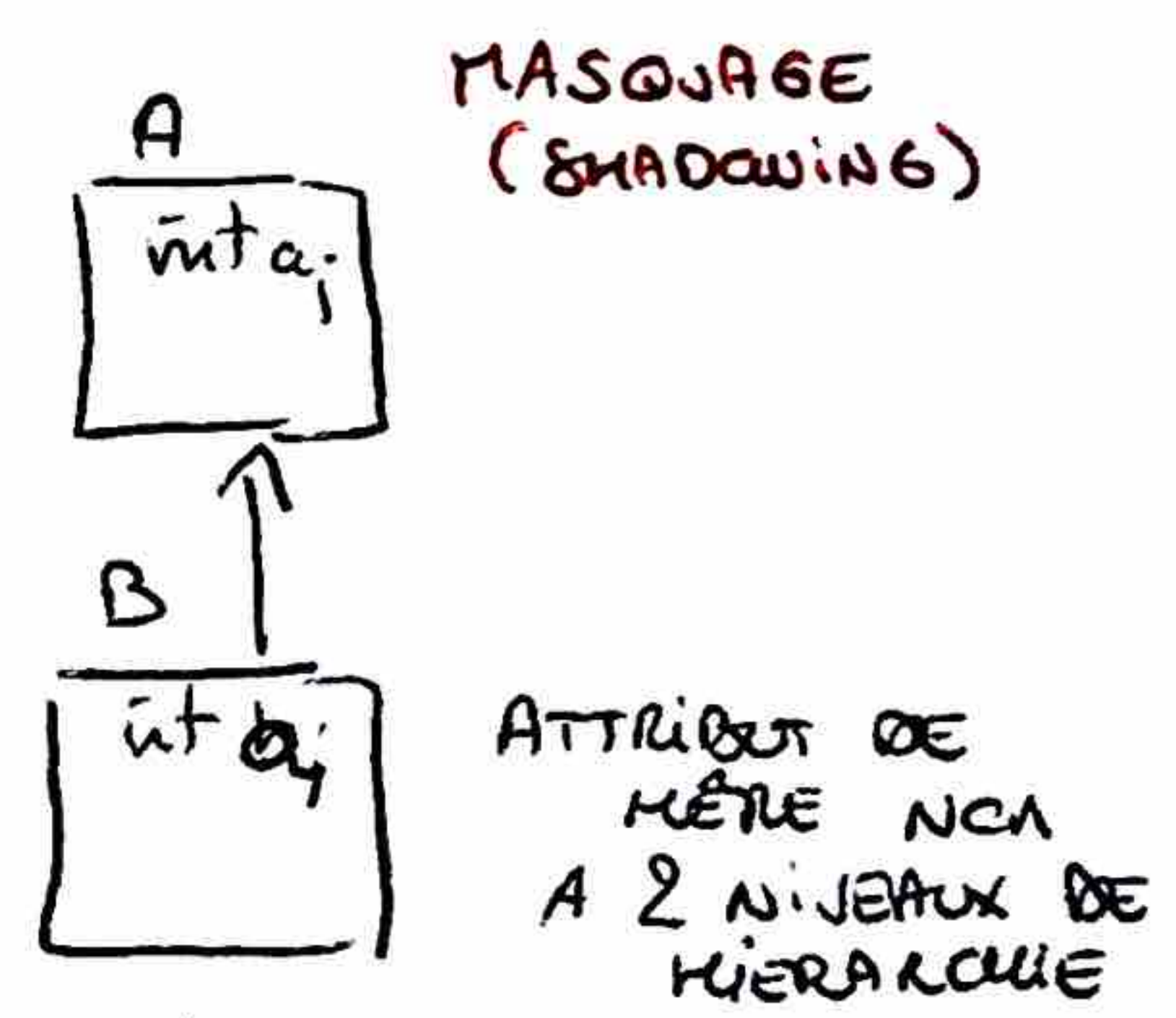
- MEMBRES PRIVES -> programmeur de la classe
- MEMBRES PROTEGES -> programmeur d'extensions
- MEMBRES PUBLIC -> programmeur utilisateur.

HERITAGE : MASQUAGE, REDEFINITION



COMMENT APPLIQUER D'ABORD UNE METHODE GENERALE PUIS UNE METHODE SPECIALISEE.

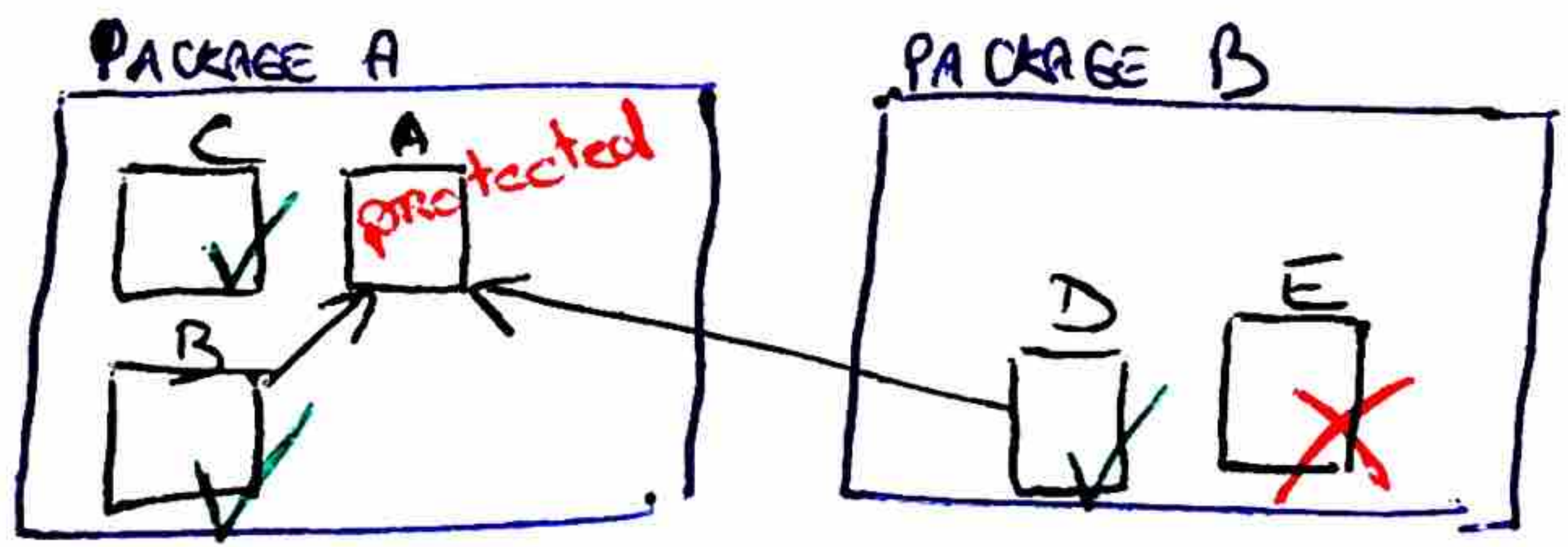
Super. methode ou attribut



B dispose de deux attributs

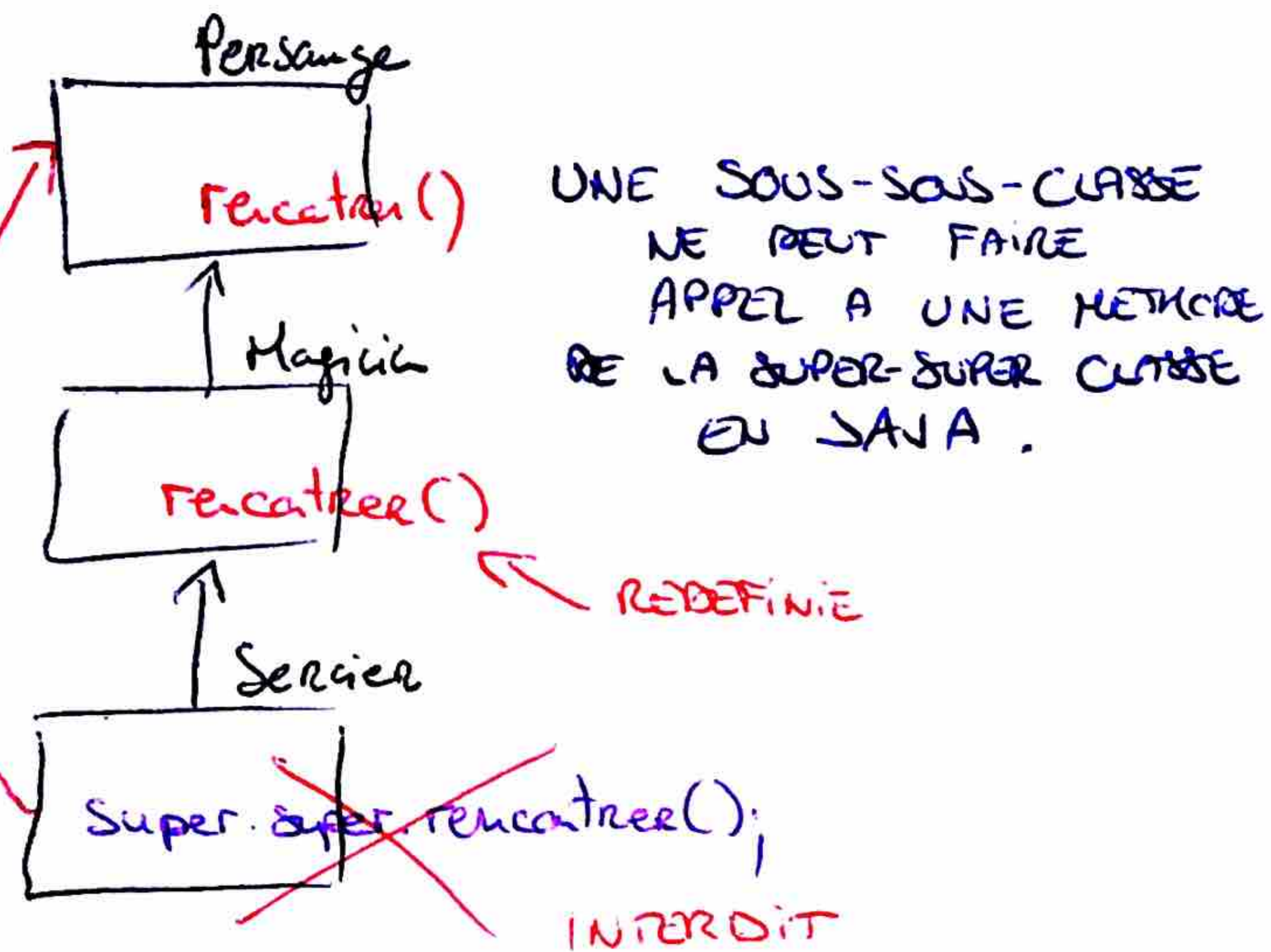
AMBIGU = PEU COURANT

PRECISION

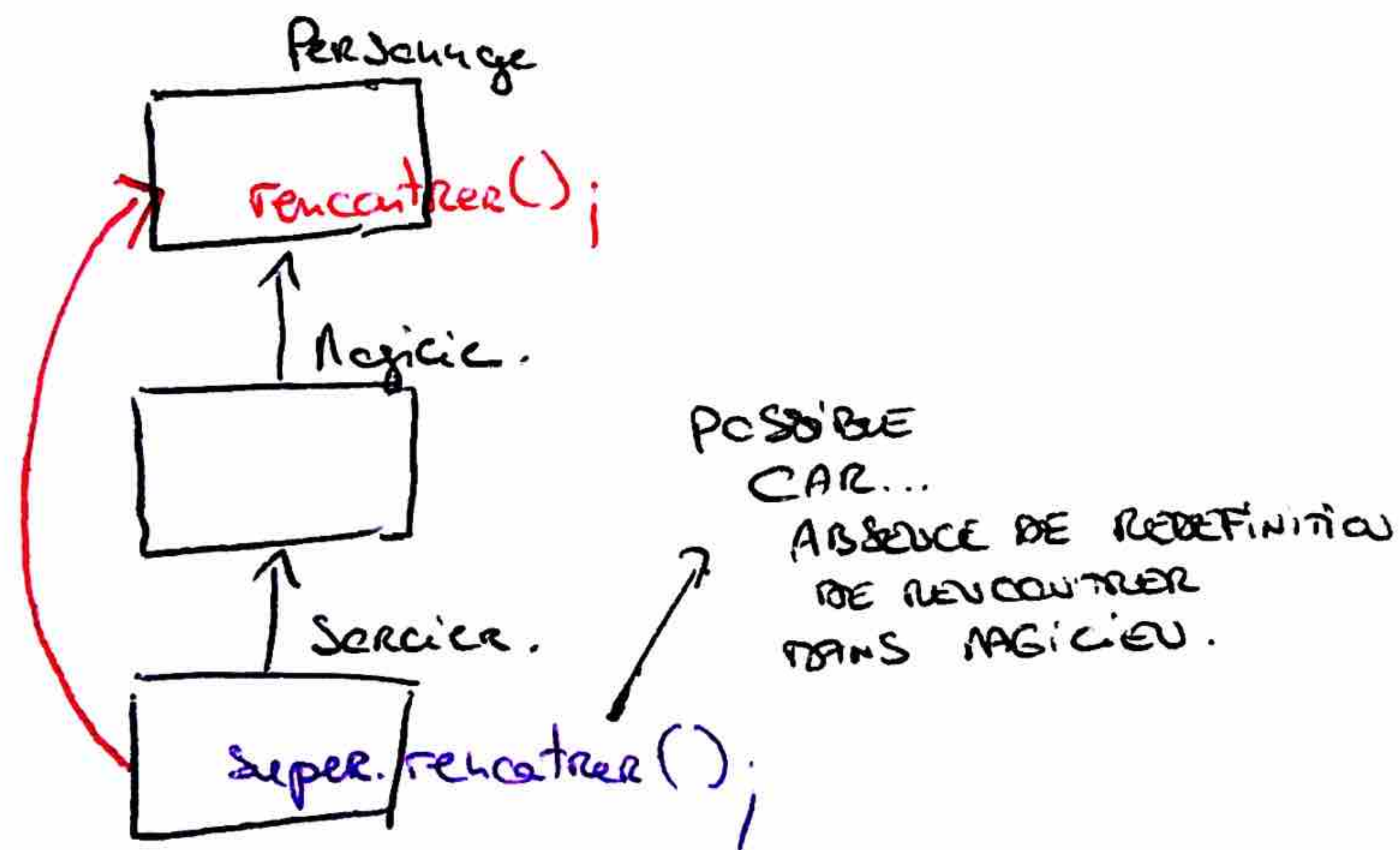


DANGEREUX POUR L'ENCAPSULATION
Si on modifie A, cela impacte B et D
SOLUTION: ~~protected~~ default

REDEFINITION : ACCES A METHODE MASQUEE SUR NIVEAUX MULTIPLES D'HERITAGE



TOUTEFOIS...



HERITAGE

HERITAGE SUR CONSTRUCTEURS

L'INITIALISATION DES ATTRIBUTS HERITES SE FAIT EN INVOQUANT LES CONSTRUCTEURS DES SUPER-CLASSES

SousClasseConstructeur (.....)

Argument
Super (Argument)
+ initialisation des attributs de la sous-classe si il y a.

⚠ Si SUPERCLASSE A UN CONSTRUCTEUR PAR DEFAUT INVOCAION DU CONSTRUCTEUR DE LA SEUL CLASSE NON OBLIGATOIRE

RESUME

- ① CHAQUE CONSTRUCTEUR D'UNE SEUL CLASSE doit appeler super
- ② ARGUMENTS DE SUPER SONT AU MOINS UN DES CONSTRUCTEURS DE LA SUPER CLASSE
SIGNATURE METHODE SUPERCLASSE
↓
SousClasse super (SIGNATURE METHODE SUPERCLASSE)
- ③ L'APPEL DE SUPER EST LA 1ERE INSTRUCTION
- ④ ERREUR si APPEL + TARD ou 2 FOIS
- ⑤ AUCUNE AUTRE METHODE NE PEUT APPeler super(..)

↓ EVITER PROBLEMES HIERARCHIE DE CLASSE

TOUJOURS DECLARER AU MOINS UN CONSTRUCTEUR.
TOUJOURS FAIRE APPEL A super(..).