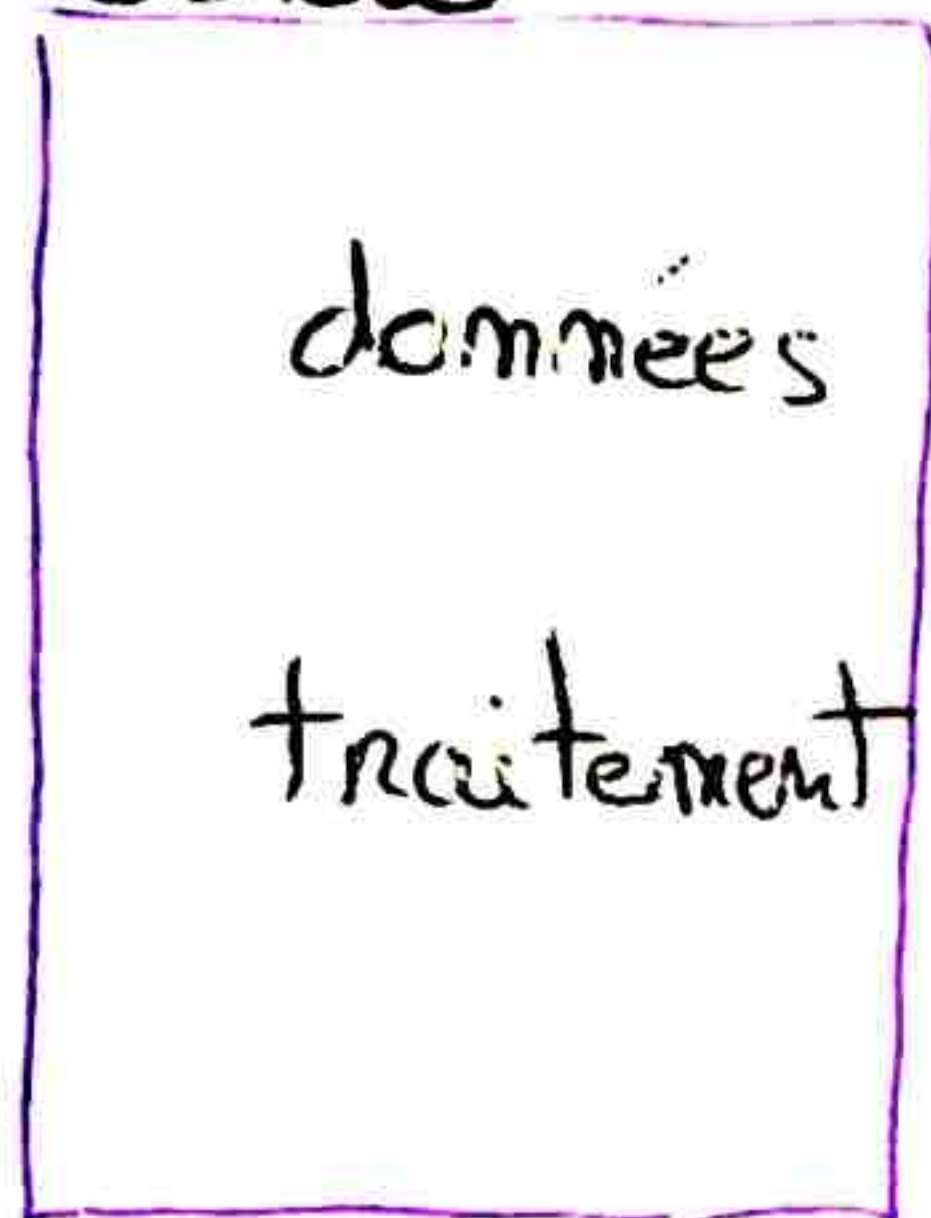


TRANSFORMATION  
EN  
1  
ENTITÉ

⇒ PROGRAMMATION ORIENTÉE OBJET

CLASSE



AVANTAGES

ROBUSTESSE  
MODULARITÉ  
LISIBILITÉ

FACE AU  
CHANGEMENT

FACE AU  
ERREURS DE MANIPULATION

ORGANISATION DU  
PROGRAMME  
COMPLEXE

↓ VIA 4 NOTIONS CLÉS

ENCAPSULATION

ABSTRACTION

HERITAGE

POLYMPHISME

PASSAGE PROGRAMMATION  
PROCÉDURALE

vers

POO

CARACTÉRISTIQUES  
COMMUNES  
= REPÉTITIF!

A  
DONNÉE  
DONNÉE  
METHODE (Fonction) OPÈRE SUR

ARGUMENTS DE  
LA FONCTION

B  
DONNÉE  
DONNÉE  
METHODE (Fonction) OPÈRE SUR

MÊME  
1 REPRÉSENTATION GÉNÉRIQUE

CRÉATION D'UN

TYPE

CORRESPONDANT  
À  
REPRÉSENTATION  
ABSTRAITE  
CONSTRUITE

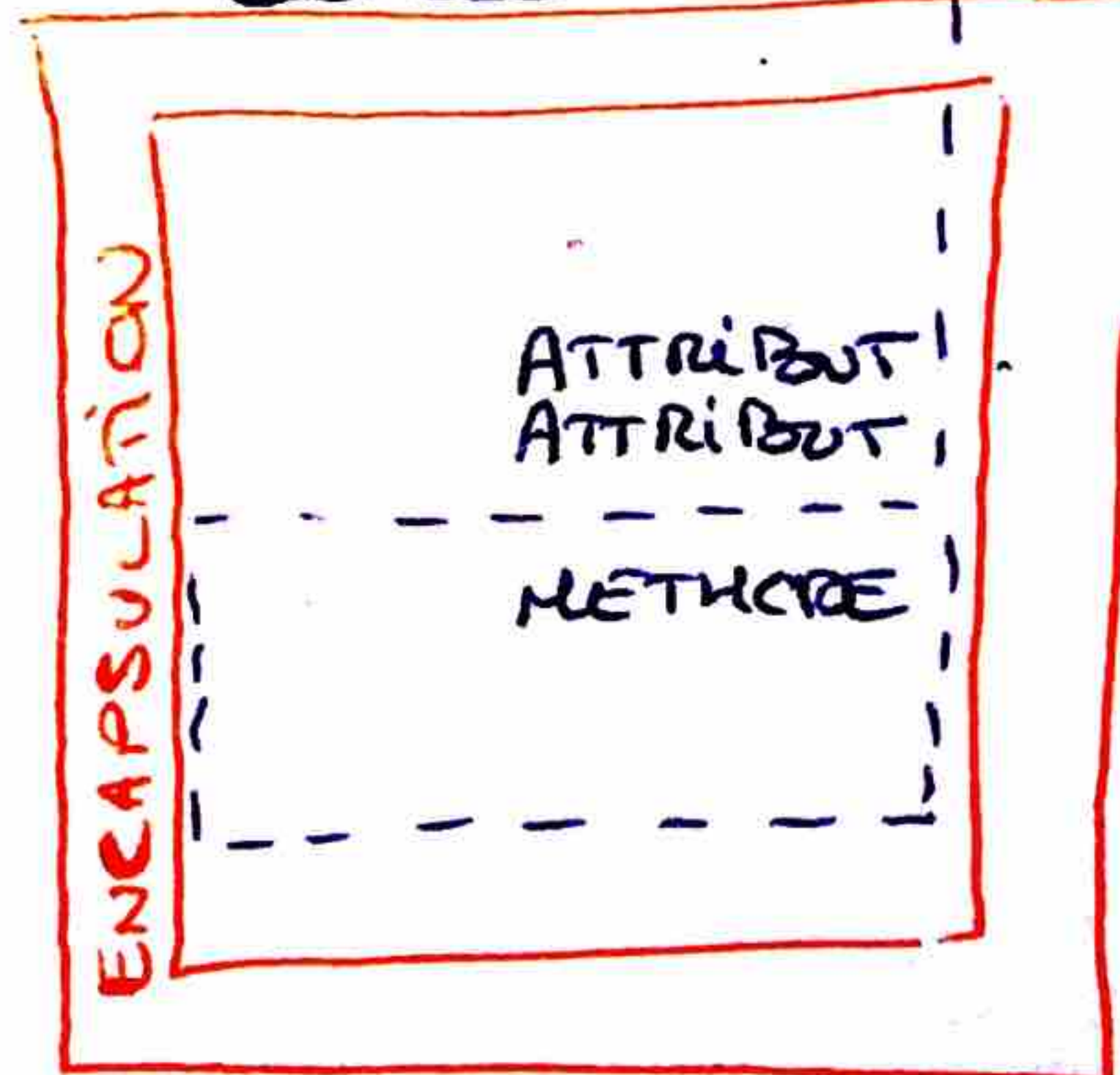
PROCESSUS

ABSTRACTION

=  
JE RÉPÈRE DES  
CHOSSES  
REPÉTITIVES

CALCUL IDENTIQUE EN A ET B  
DONNÉES (TYPE) IDENTIQUE EN A ET B

CLASSE



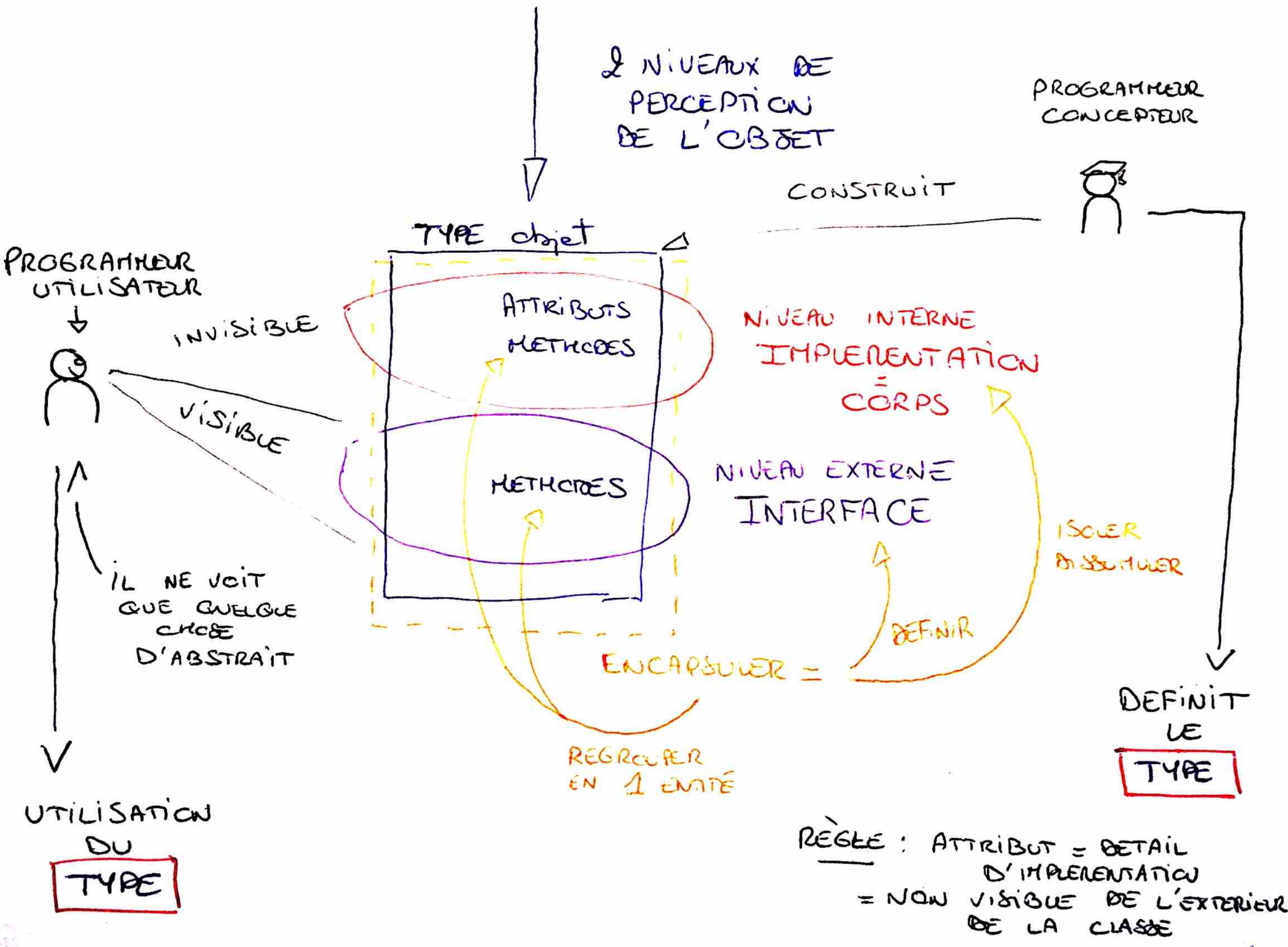
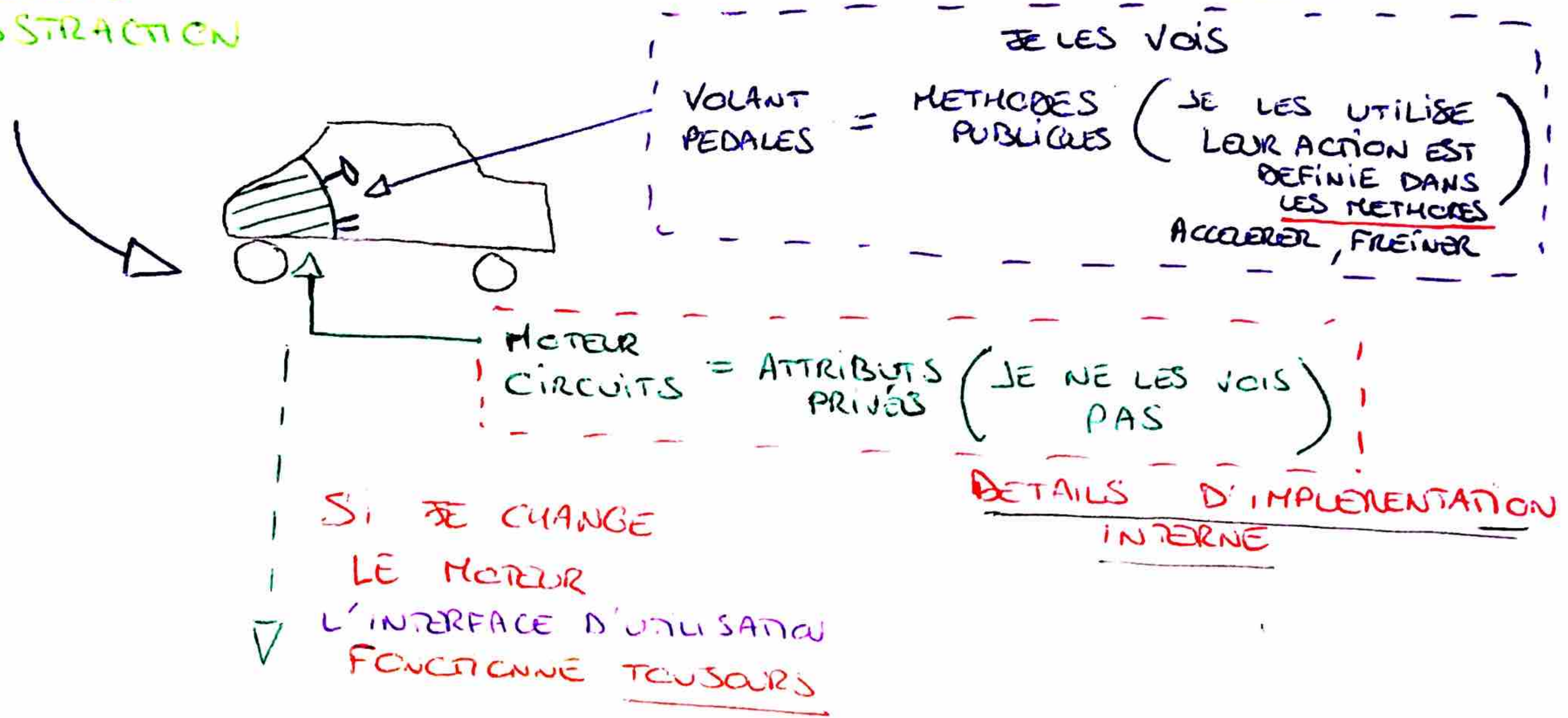
JE NE VOIS PLUS DE L'EXTÉRIEUR  
QUE L'INTERFACE D'UTILISATION  
METHODES



PROCESSUS  
D'ABSTRACTION

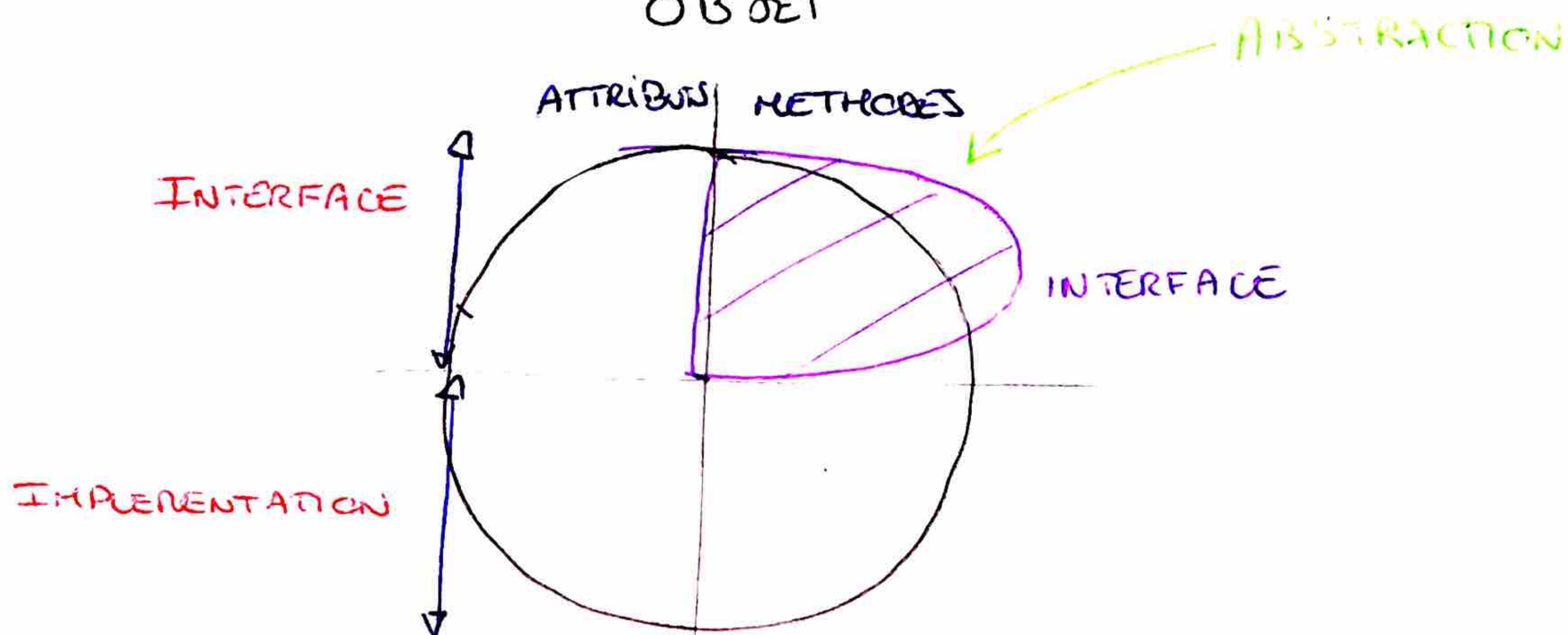
ANALOGIE : INTERFACE  
D'UNE VOITURE

INTERFACE  
D'UTILISATION





# DEFINIR UN OBJET



PROGRAMMEUR  
CONCEPTEUR

PROGRAMME

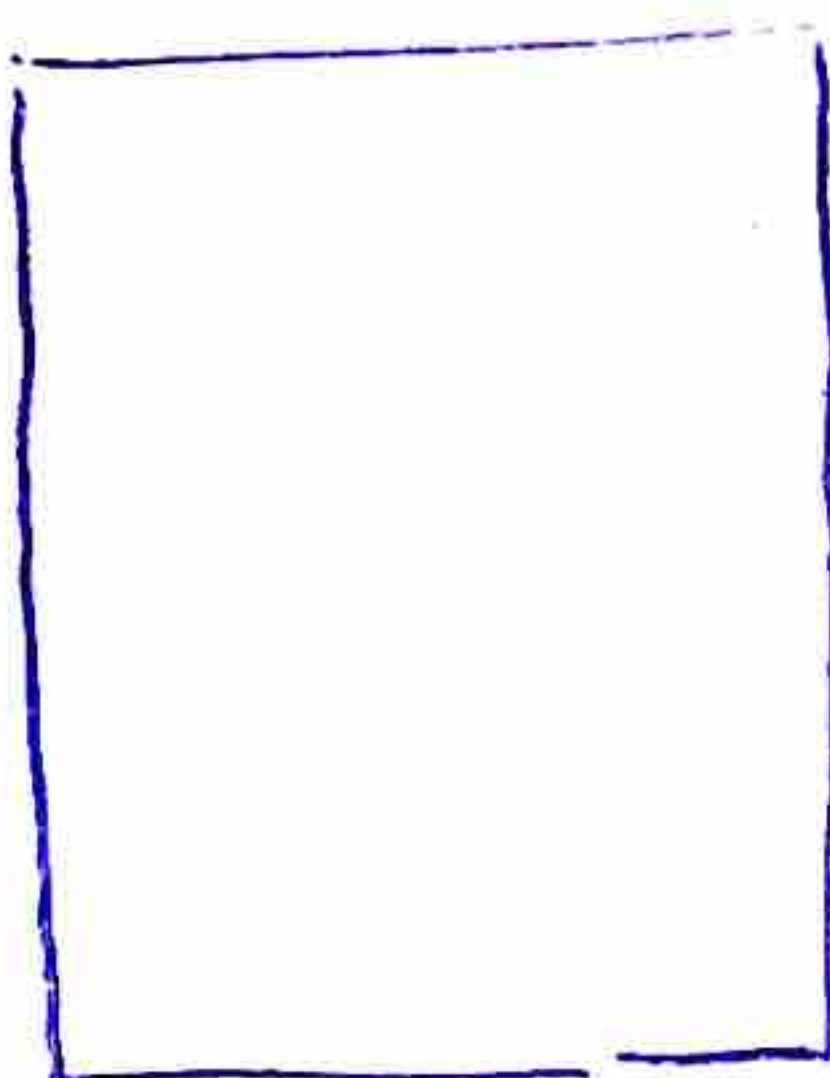
OU

ECRIURE

PROCESSUS

ABSTRACTION  
ENCAPSLATION

RESULTAT :



ECRIURE  
D'UNE  
CLASSE

DEFINIT  
UN  
NOUVEAU

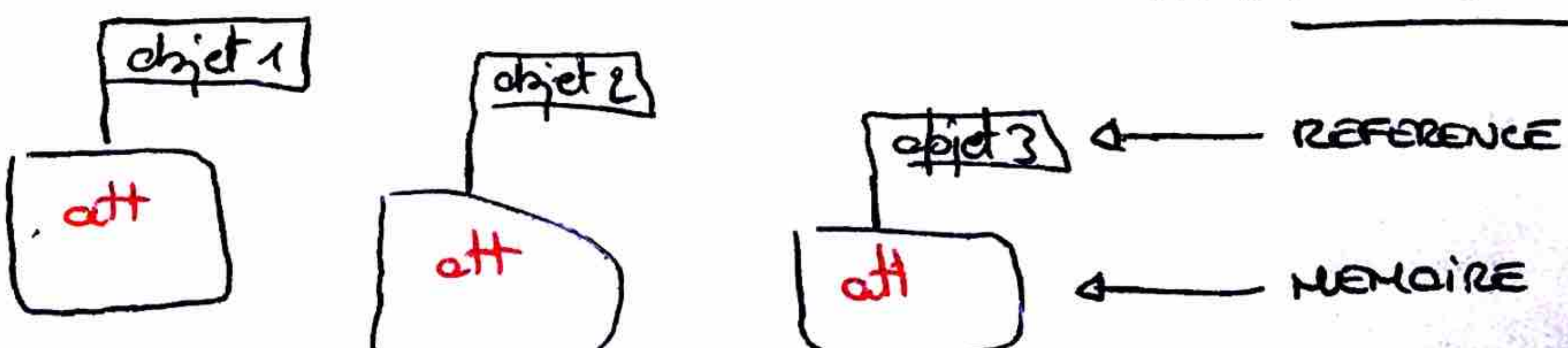
TYPE

UTILISATION

TYPE variable = new ...

DECLARATION  
D'UN  
OBJET  
DE NOM variable

L'OBJET EST  
MANIPULE  
VIA VARIABLE



PROGRAMMEUR  
UTILISATEUR

EXECUTION DU PROGRAMME