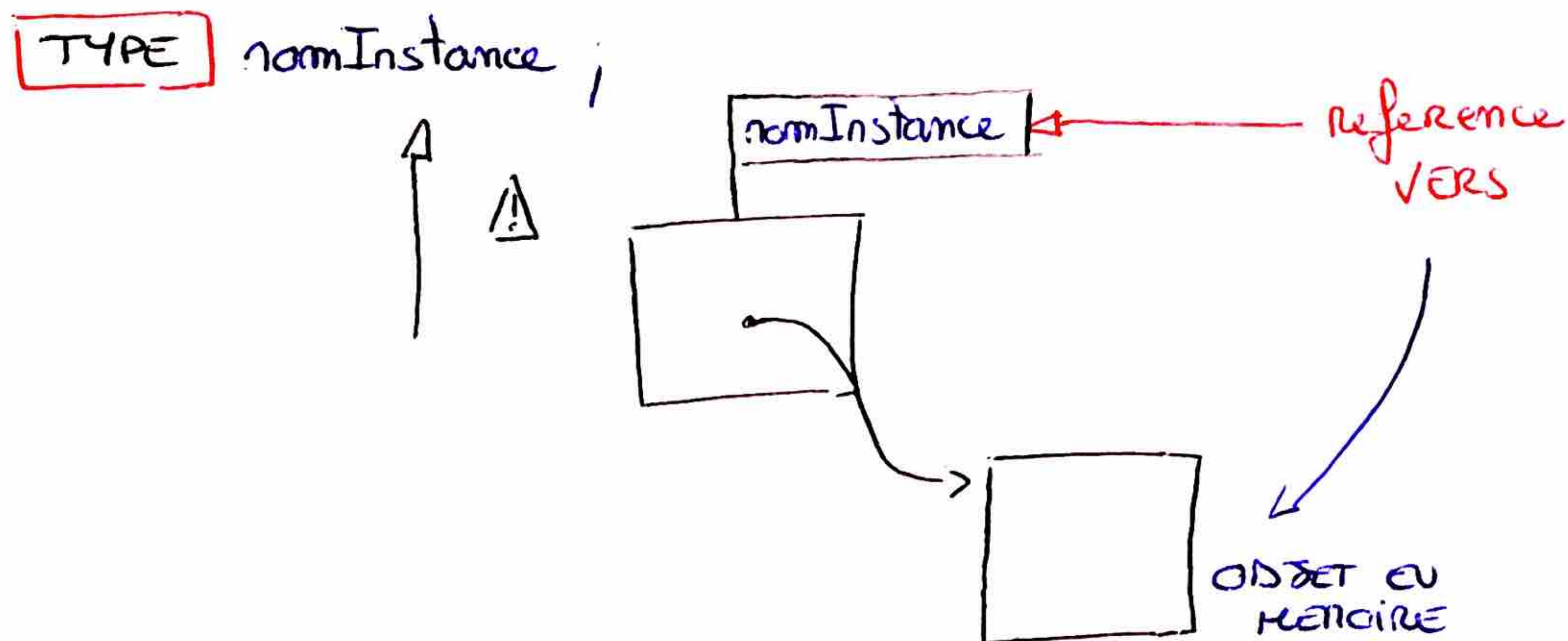


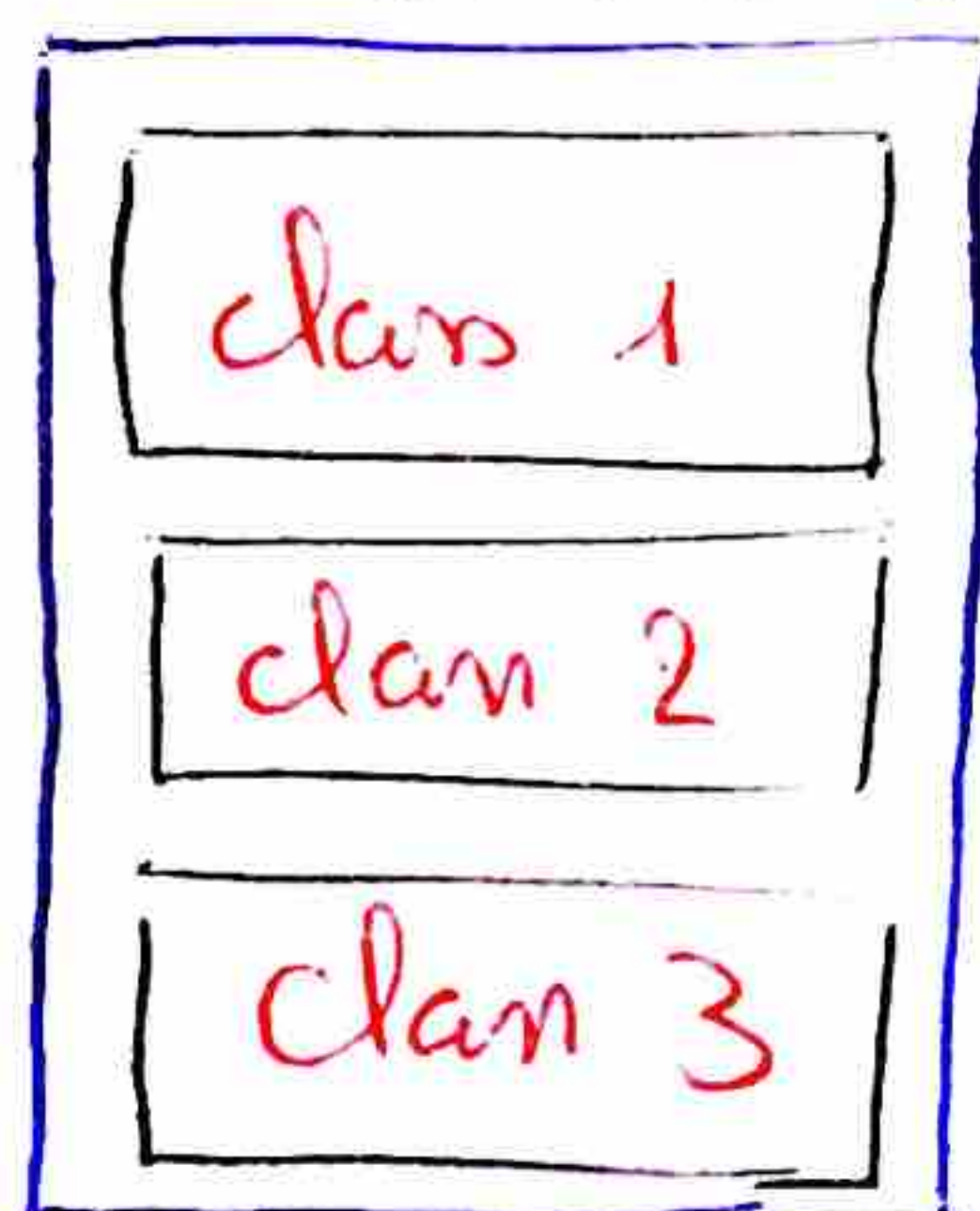
CLASSE → TYPE → VARIABLES

┌ = INSTANCES
└ = OBJETS



POSSIBILITÉS DE DECLARATION DES CLASSES

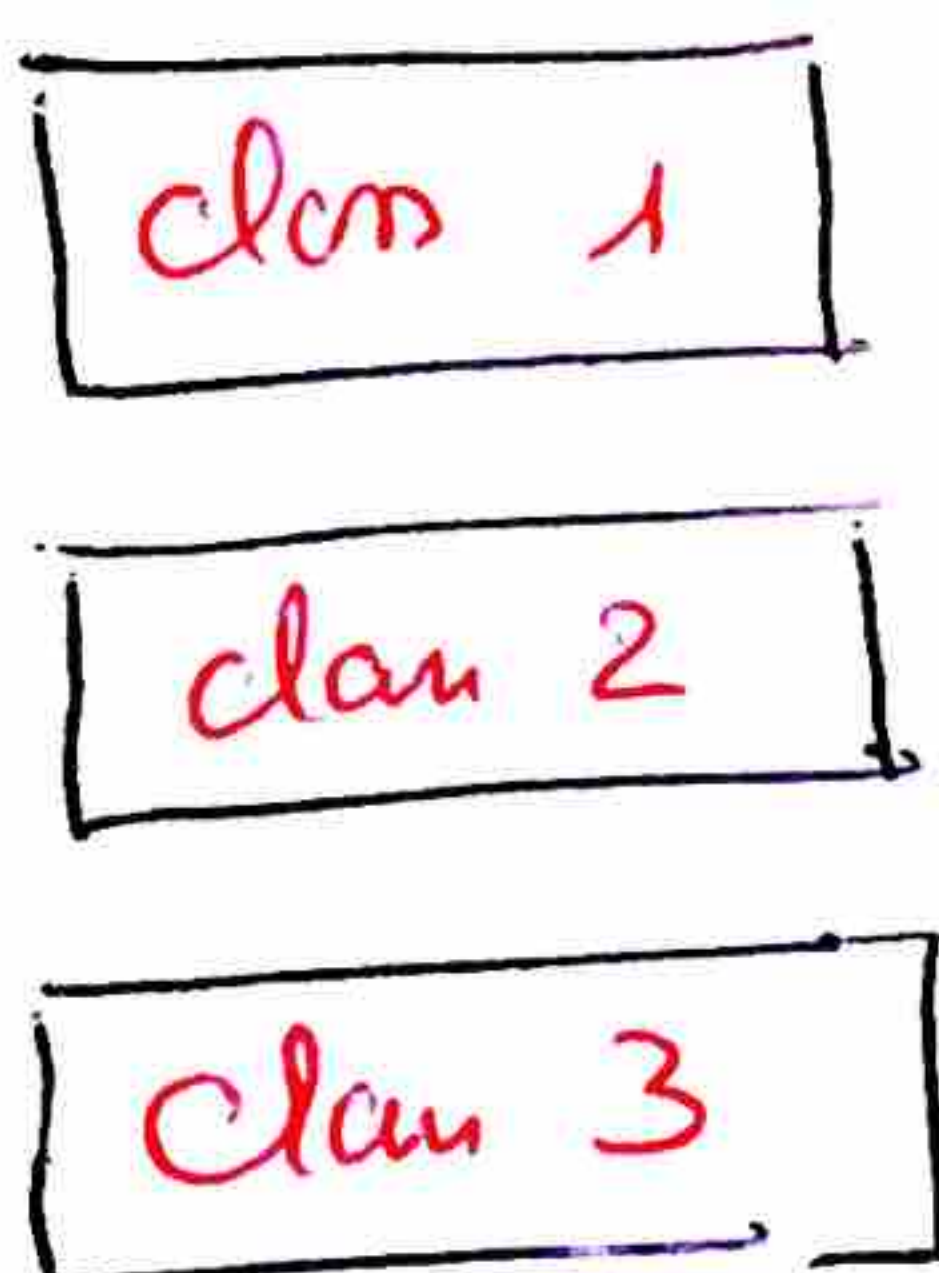
1 FICHIER JAVA



COMPILATION

→ javac fichier.java

OU



COMPILATION VIA IDE

Build

→ javac class1.java

→ javac class2.java

...

⚠ SEUL UNE CLASSE AVEC METHODE MAIN() EST EXECUTABLE

ACCES AUX ATTRIBUTS

instance . attribut

DECLARATION - INITIATION INSTANCE

nom Classe instance = new nomClasse();

└─ VAEUR PAR DEFALT ATTRIBUTS

int	0
double	0.0
boolean	false
objets	null

DECLARATION METHODE

type_retour nom_methode (type parametre 1, nom parametre 1, ...)

{

corps de la methode

}

↓ ⚠ BONNE PRATIQUE
PORTEE DES ATTRIBUTS

CLASSE

class Rect {

int l1

int l2

double surface()

{

return l1 x l2

}

}

PAS DE PASSAGE
D'ARGUMENTS EN PARAMETRE

~~double surface (double l1, double l2)~~

~~{~~

~~return l1 x l2;~~

~~}~~

NON!

NE JAMAIS PASSER LES ATTRIBUTS
DE LA CLASSE COMME ARGUMENT DE
SES METHODES

PORTEE DE CLASSE

TOUTES LES METHODES DE LA CLASSE ONT ACCES
A SES ATTRIBUTS

EXCEPTION SI ET SEULEMENT SI

ON DOIT PASSER DES PARAMETRES EXTERIEURS A LA
CLASSE → DONE "EXTERIEURS A L'INSTANCE

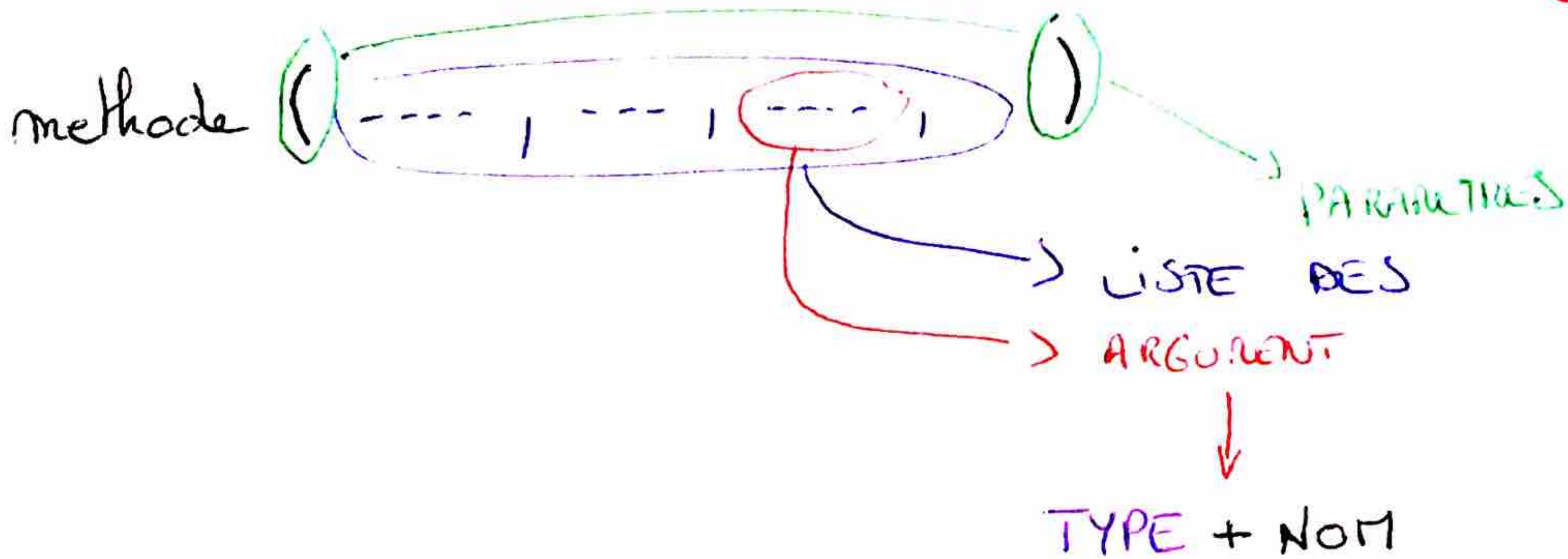
EXEMPLE PASSAGE DE PARAMETRES EXTERIEURS A LA CLASSE

```
class FigureColoree
{
    void colorie (couleur c)
}
```

```
FigureColoree me-figure;
couleur rouge;
me-figure.colorie(rouge);
```

CE N'EST PAS UN
ATTRIBUT DE LA
CLASSE FigureColoree

LANGUAGE :



BONNE METHODE ! (DE PROGRAMMATION)

DANS LA CLASSE MAIN :

{

1. DECLARATION - INSTANCIATION INSTANCE
2. AFFECTATION VALEURS AUX ATTRIBUTS VIA INTERFACE
3. POUR LES AFFICHAGES

SYSO ("texte..." + instance.methode())



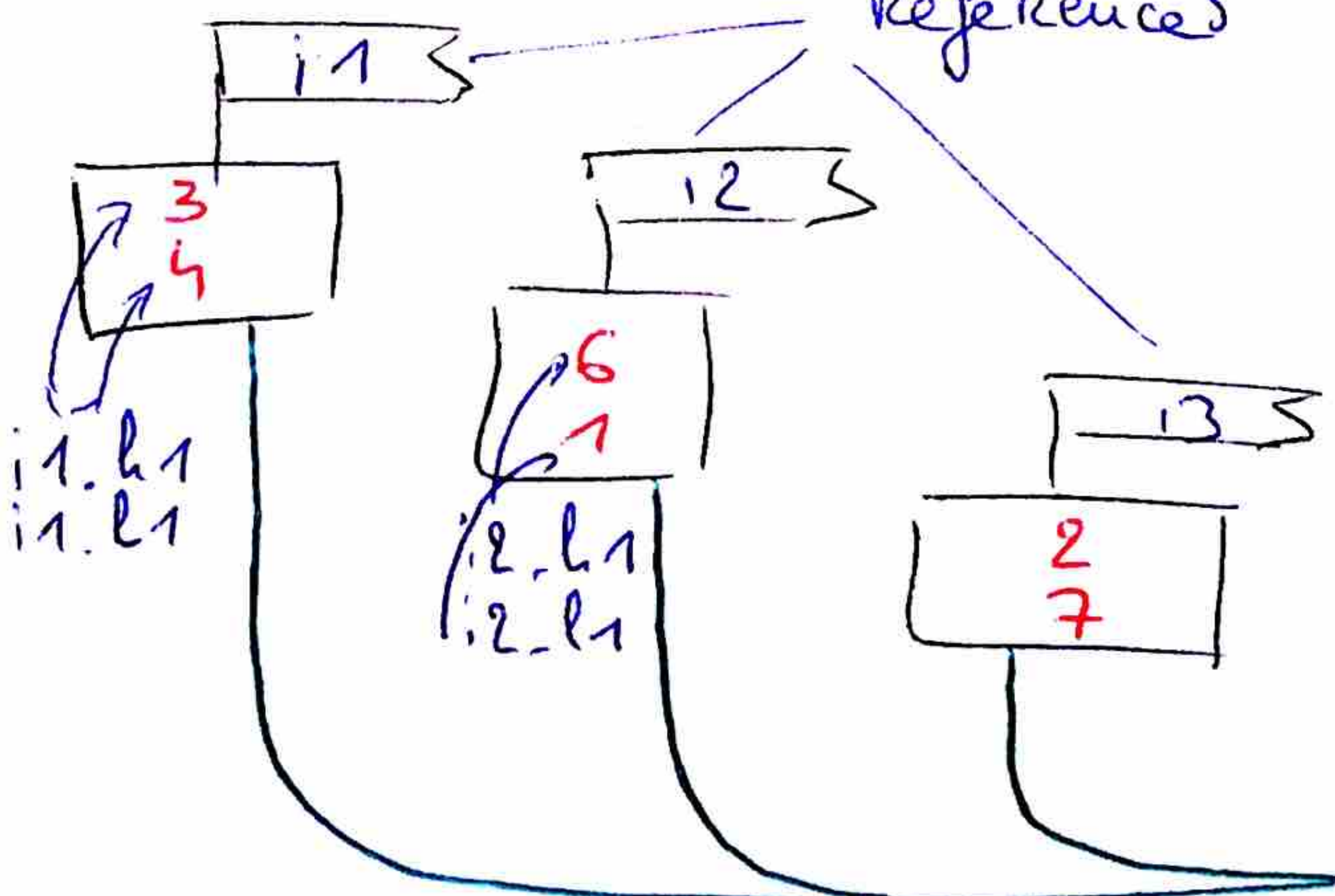
var line System.out.println

VIDE

de type int
string
double
boolean
avec un return

ET UNE PORTÉE
DE CLASSE !

References



Δ VARIABLE = REFERENCE
VERS OBJET EN MEMOIRE.

⇒ CHAQUE INSTANCE A SES
PROPRIES ATTRIBUTS.

3 VARIABLES DIFFERENTES A DES
ENDROITS ≠ EN MEMOIRE