

## TD 2 - Encapsulation, surcharge

NB QZ 2.1:

float val = 1.12;

↳ Erreur car 1.12 est de type double

Différence entre float (32 bits) et double (64 bits)

↳ cf p 55 TD/TNE

CAST

float val = 1.12;  $\Rightarrow$  float val = (float) 1.12;

32 bits

64 bits

perte d'information

$\Rightarrow$  Refus de Java

↳ dit à Java: J'autorise la conversion même si on perd de l'info

Conversion

EXPLICITE

float val = 1.12;

↳ On demande une conversion implicite

$\Rightarrow$  float val = 1.12f

↳ On stock 1.12 directement en float

QZ 2.2:

int x = 2, y = 5; double z = x / y;

↳ La valeur de z est 0

car / représente la division entière et comme x et y sont des entiers, le résultat de la division de l'un par l'autre est un entier

2 solutions {  
- double z = (x \* 1.0) / y;  
- double z = ((double) x) / y;

QZ 2.3 :

- a)  $1 + "1" \Rightarrow "11"$   
 (conversion)  $\swarrow$  + concatenation de String  
 $"1" + "1"$
- b)  $\underbrace{1}_{\text{int}} + \underbrace{'1'}_{\text{char}} \Rightarrow 50$  (type int)  
 (code ASCII)  $\swarrow$   
 $1 + 49$
- c)  $'1' + "1" \Rightarrow "11"$

QZ 2.4 :

- a)  $1 + '1' + "1" \Rightarrow "501"$   
 $\underbrace{1 + '1'}_{50} + "1"$   
 $\underbrace{"50" + '1'}_{"501"}$
- b)  $"1" + '1' + 1 \Rightarrow "111"$   
 $\underbrace{"1" + '1'}_{\text{false}} + 1$   
 $\underbrace{"11" + 1}_{\text{false}}$

QZ 2.5 :  $\underbrace{\text{true} \&\& \text{false}}_{\text{true}} == \underbrace{\text{false}}_{\text{true}} \Rightarrow \text{true}$

Type : booléen

car **Priorité des Opérateurs**

$\rightarrow ==$  est prioritaire vis à vis de  $\&\&$



QZ 2.6 :

System.out.println("Bonjour \n vous \t tous !");

le caractère  
"passage à la ligne"  
le caractère  
"tabulation"

↳ Bonjour  
vous      tous !  
            tab

Ex 15

Q1

- 2 constructeurs : lignes 4 et 7
- L'un est défini en prenant en compte un argument volume de type double et l'autre est défini sans prendre en compte d'argument.
- Bouteille b1 = new Bouteille(1.);  
Bouteille b2 = new Bouteille();

Q2

- this.volume : référence à la variable d'instance volume de l'objet courant sur lequel on applique le constructeur
- volume : représente le paramètre du constructeur Bouteille utilisé.

Q3

- this(1.5); ⇒ this() : Appel à un constructeur de la classe courante. Ici, c'est le précédent constructeur Bouteille auquel on fait appel, en prenant un argument de type double

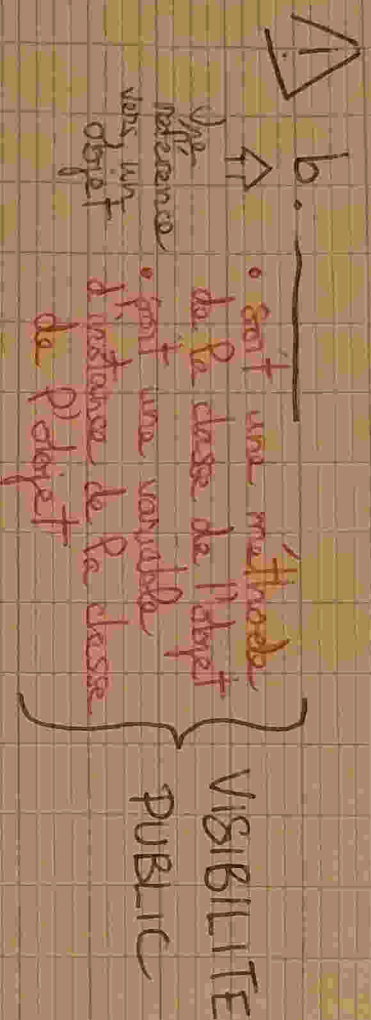
Q4

public void remplir(Bouteille b) {

this.volume += b.volume;

b.volume = 0;

}



Q5

↳ surcharge de la méthode possible

public void remplir (double v) {

this.volume += v;

}

Q6

5 → "Volume du liquide dans la bouteille = 10.0"

6 → "Volume du liquide dans la bouteille = 4.5"

Entre temps : 7 → b1.remplir(b2),

↳ Ainsi le volume de la bouteille b2 (contenu dans b2) est transféré dans la bouteille courante b1

8 → "Volume du liquide dans la bouteille = 14.5"

9 → "Volume du liquide dans la bouteille = 0.0"