

## Q7 - Instanciation p 16

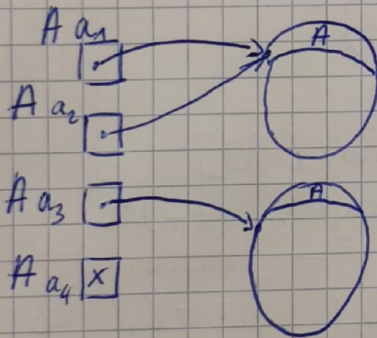
```
main {  
    A a1 = new A();  
    A a2 = a1;  
    A a3 = new A();  
    A a4 = null;  
}
```

→ la classe A() contient un constructeur par défaut (propre au langage JAVA)

→ Un constructeur est utile pour initialiser des variables d'instance.

Méthodes par défaut qui marchent sans être définies:

```
a1.toString();  
a1.clone();
```

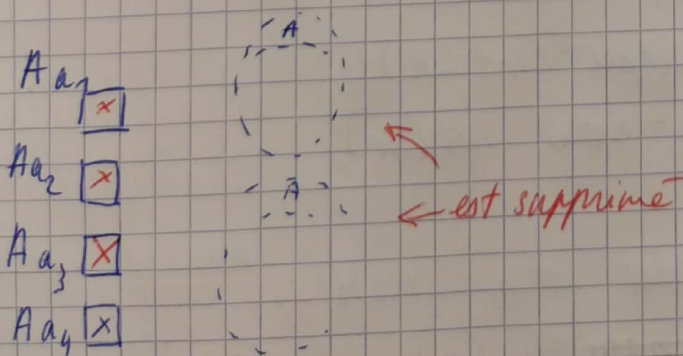


handle: Toute variable susceptible de contenir des références

7.2 2

7.3 4

7.4 En rouge

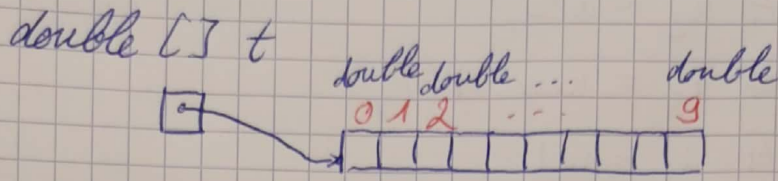


Ex 27

Q 27.1

double[] t = new double[10];  
 ✓ t contient une référence vers un tableau de double

mémoire:



Premier élément: t[0]

i<sup>ème</sup> élément du tableau: t[i-1]

for (int i=0; i < t.length; i++)

t[i] = Math.random();

System.out.println(t.length);  
 → 10

for (int i=0; i < t.length; i++)

System.out.println(t[i]);

Point[] t = new Point[10];

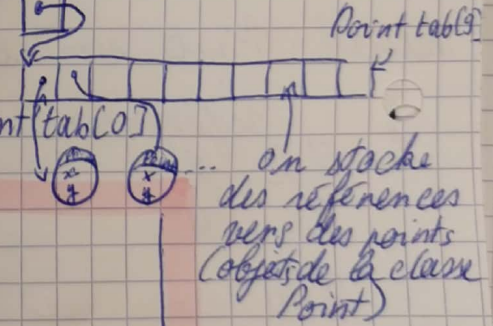
27.2 for (int i=0; i < t.length; i++) {

t[i] = new Point();

System.out.println(t[i]); }

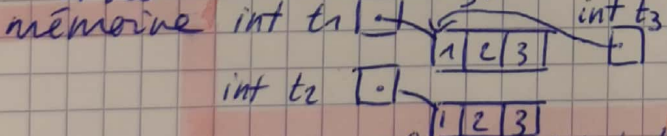
mémoire:

Point[]-tab



27.3 False → (t1 == t2)

true → (t1 == t2)



Ex 28

Q 28.1 ← public class NUp1et {

private int[] tab;

public NUp1et(int n) {

tab = new int[n];

}

public NUp1et(int n, int x) {

this(n);

for (int i=0; i < tab.length; i++)  
 tab[i] = x;

tab = null;  
 → supprime d'abord le tableau puis les objets

Rappel: int[] t1 = {1, 2, 3};  
 équivalent à:  
 { int[] t1 = new int[3];  
 t1[0] = 1; t1[1] = 2;  
 t1[2] = 3; }



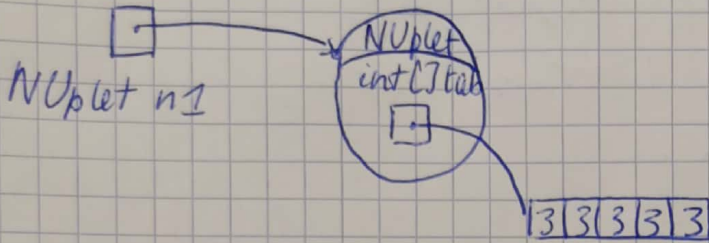
```

public NUplet (int a, int b, int c) {
    this(3);
    tab = {a, b, c}; NON
    tab[0] = a; tab[1] = b; tab[2] = c;
}

```

28.2

mémoire

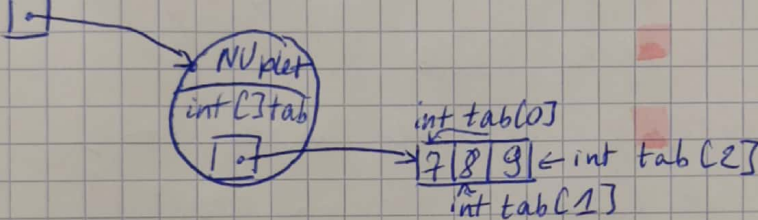


main

`NUplet n1 = new NUplet(5, 3);`

`NUplet n2 = new NUplet(7, 8, 9);`

NUplet n2



Q28.3 public int somme () {

`int res = 0;`

`for (int i = 0; i < tab.length; i++)`

`res += tab[i];`

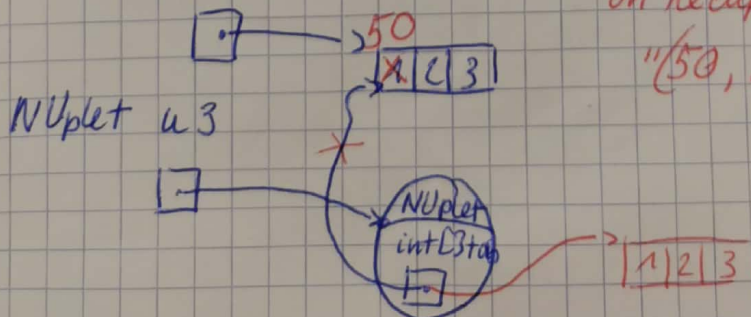
`return res;`

}

Q28.4

mémoire

`int[] t = {1, 2, 3}`



on récupère bien  
"(50, 2, 3)"

On peut écrire :

```

public NUpLet (int [] tab) {
    this.tab = new int [ tab.length ];
    for (int i = 0 ; i < tab.length ; i++) {
        this.tab[i] = tab[i];
    }
}

```

Q28. 6

```

public boolean egal (NUpLet n2);
if (n2 == null) return False;
if (this == n2)
    return true;

if (tab.length != n2.tab.length)
    return False;
for (int i = 0; i < tab.length; i++)
    if (tab[i] != n2.tab[i])
        return False;

return true;
}

```