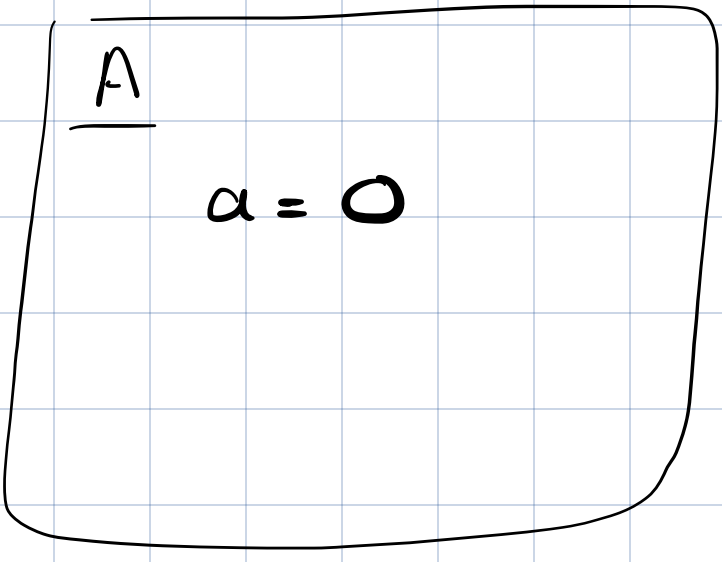
Exercice 2



A est initialisé avec le constructeur par défaut (donc a vaut 0 au début).

Puis m(a) ne modifie pas a, il modifie simplement la variable locale a.

Le sout va renvoyer 0.

(2) public enum SemaphoreColor {

RED = SemaphoreColor("rouge", false);
GREEN = SemaphoreColor("vert", true);
YELLOW = SemaphoreColor("jaune", false);
boolean canPass; String frenchName;

public SemaphoreColor(String color, boolean canPass) {
this.frenchName = color;
this.canPass = canPass;
}

public boolean canPass() {}

public String frenchName() {}

}

③ Un objet de type B a ~~3~~ 2 attributs :

~~super.a (défaut 1)~~
a (défaut 2)
b (défaut 3)

⑥ Elle a 2 constructeurs :
- public A() ;
- public A(int a) ;

⑥ 1 (constructeur par défaut)

⑥ A : 1
B : 3 2

B : 3 2

⑥ B : 3 1

⑧ Oui, le constructeur par défaut de A sera appelé

(9) Le code ne va plus compiler. Il faudra que B appelle le constructeur public A(int a) pour fonctionner.

(4) La class B n'implémente pas la fonction m sans paramètre abstraite de A.

L. 10, a n'est pas un attribut de B, mais de A, qui est privée donc cette instruction ne compile pas.

A ne peut pas être instanciée.

L. 10
```
public B(int unA, int unB) {
        b = unB;
}
```

L. 12
```
public void m() {
        System.out.println("plz be nice
                avec le constructeur");
}
```

9 + 5 + 4 = 18

Exercice 3

(1) Cannot be sent: Suspicious
@ @ @ @ @
Sent

(2) Cannot be sent: Suspicious
@ @ @ @ @
Cannot be sent: Suspicious

(3) Non, RuntimeException est trop precis (la fonction peut renvoyer des Exception tout court).

(4) RuntimeException, car null.toto ne fonctionne pas.