

# Initiation

## Exercice 1

Les variables  $a$ ,  $b$  et  $c$  sont de type *int* ou *boolean*.

Indiquer la valeur de  $a$ ,  $b$  et  $c$  pour chaque ligne.

Écrire le programme qui permettra d'afficher ces valeurs.

a	b	c	Expression	a ==	b ==	c ==
6	7	4	$c = (a \ \& \ b)$	a==	b ==	c ==
true	5	3	$b = (!a ? b * = c : c * = b)$	a==	b ==	c ==
5	2	4	$b = ((b \mid c) != ++a ? 5 : 1)$	a==	b ==	c ==
0	1	2	$a = (a++) * ((++b) * (c++))$	a==	b ==	c ==
0	1	2	$a = (++a) * ((++b) * (c++))$	a==	b ==	c ==
8	true	4	$b = (false \parallel ((c * = 2) > 0))$	a==	b ==	c ==
8	false	4	$b = (true \parallel ((c * = 2) > 0))$	a==	b ==	c ==
5	4	3	$c = ((a > b) \&\& (b > c) ? b : c)$	a==	b ==	c ==
6	false	7	$a \&= ((true \&\& b) ? -a : c -= 3)$	a==	b ==	c ==
4	8	1	$b += (a++) + (++c)$	a==	b ==	c ==

## Exercice 2

Écrire le programme donnant les limites minimums et maximums de chacun des types entiers et réels.

## Exercice 3

Étudier le programme suivant :

```
public class Limites {
    public static void main (String [] args) {
        int i1 = 1000000000;
        int i2 = 2 * i1;
        int i3 = 3 * i1;
        int i4 = Integer.MAX_VALUE;
        System.out.println("i1_:_ " + i1);
        System.out.println("i2_:_ " + i2);
        System.out.println("i3_:_ " + i3);
        System.out.println("i4_:_ " + i4);
    }
}
```

1. Ce programme compile-t-il ?
2. Que se passe-t-il à l'exécution ? Pourquoi ?

## Exercice 4

En utilisant les 3 types de boucles vus en cours, écrire un programme qui :

- affiche les  $n$  premiers entiers,
- calcule et affiche la somme des  $n$  premiers entiers,
- calcule et affiche  $n!$ .

## Exercice 5

On définit dans cet exercice les classes *Segment* et *Point* pour représenter des segments de droites dans un plan définis par deux points du plan.

1. Écrire la classe *Point*. Écrire la classe *Segment*. Chaque segment contient deux points nommés start et stop. Ajouter les getters et les setters.

2. Écrire deux constructeurs. L'un prendra en argument deux objets *Point*, et l'autre quatre variables entières.
3. Écrire une classe *SegmentTest* destinée à tester les classes *Segment* et *Point*. Elle contiendra une méthode *main* où on crée un segment *s1* avec le premier constructeur et un segment *s2* avec le second constructeur.
4. Écrire dans la classe *Segment* une méthode *toString* permettant d'afficher les deux points d'un segment. On utilisera la méthode *toString* de *Point*.
5. Écrire dans la classe *Segment* une méthode *translate* permettant de déplacer un segment dans le plan. On utilisera une méthode *translate* de *Point*.
6. Écrire une méthode qui teste si deux segments ont une extrémité (un point) en commun. Dans quelle classe faut-il la mettre? Quel est le type de retour de cette méthode?
7. Tester vos méthodes dans *SegmentTest*.
8. On peut également imaginer la classe *Triangle*. Codez cette classe, qui permettra notamment de déterminer si un triangle est rectangle, isocèle ou équilatéral. On pourra aussi calculer les coordonnées du centre du cercle circonscrit à ce triangle.