INF1 - TD 2

UNIVERSITÉ DE RENNES 1

Exercice 1 : If imbriqués

On considère la suite d'instructions suivante, où a et b sont deux variables de type entier :

```
1 if (a > b) {
2   if (a > 10) System.out.println("1");
3   else if (b < 10) System.out.println("2");
4 }
5 else {
6   if (a == b) System.out.println("3");
7   else System.out.println("4");
8 }
```

1. Pour quelles valeurs de a et b ce programme affiche-t-il respectivement 1, 2, 3 ou 4 ?
2. Existe-t-il des valeurs de a et b pour lesquelles ce programme n'affiche rien ?
3. Peut-on simplifier ce code, et si oui, comment ?

Exercice 2 : Mes premières boucles

1. Écrire un programme qui affiche les entiers de 1 à 10 en utilisant une boucle **for**.
2. Écrire un programme qui affiche les entiers de 1 à 10 en utilisant une boucle **while**.
3. Écrire un programme qui affiche les entiers de 1 à 10 en utilisant une boucle **do while**.
4. Écrire un programme qui affiche les entiers de 10 à 1 en utilisant une boucle **for**.
5. Écrire un programme qui demande à l'utilisateur un entier n et qui affiche les entiers de 1 à n.
6. Écrire un programme qui affiche les entiers pairs compris entre 1 et 100.
7. Écrire un programme qui affiche les entiers impairs compris entre 1 et 100, sans utiliser de **if**.
8. Écrire un programme qui affiche les entiers multiples de 3 mais non multiples de 5, compris entre 1 et 100.

Exercice 3 : Terminaison de boucles

Les programmes suivants s'arrêtent-t-ils ? Dans chaque cas, justifiez votre réponse.

1. Programme 1

```
int i = 10;
while (i >= 5) {
  i--;
}
```

2. Programme 2

```
int i = 10;
int j = 15;
while (i >= j) {
  j++;
}
```

3. Programme 3

```
int i = 0;
int j = 1;
while (i < j) {
  i++;
  i--;
}
```

4. Programme 4

```
int i = 0;
int j = 1;
while (i < j) {
  --i;
  ++j;
}
```

Exercice 4 : Figures d'étoiles

1. Pour chaque figure ci-dessous, écrire un programme qui affiche cette figure. Votre programme doit modéliser la structure sous-jacente de chaque figure de façon à pouvoir être généralisé (cf. question suivante) ; il faut donc trouver mieux qu'un enchaînement de cinq `print(...)` :

```
*                *****            *****
**                ****             *
***                ***              *
****                **              *
*****                *            *****
```

2. Même question, mais la hauteur h des figures (ci-dessus $h = 5$) est demandée à l'utilisateur.