

# Programmation orientée objet

## Série 1

Joel Cavat / 2020

### Structure de contrôle

#### 1.1 Exercice

- Ecrivez un code qui simule le comportement d'une boucle `while` à l'aide de la boucle `do while`
- Ecrivez un code qui simule le comportement d'une boucle `do while` à l'aide de la boucle `while`

Utilisez le modèle ci-dessous:

```
// Loop.java
public class Loop {
    public static void whileWithDoWhile(int i) {
        /* essayez par exemple de reproduire l'équivalent suivant:
        while(i < 5) {
            System.out.println(i);
            i++;
        } */
    }

    public static void doWhileWithWhile(int i) {
        /* essayez par exemple de reproduire l'équivalent suivant:
        do {
            System.out.println(i);
            i++;
        } while (i < 5); */
    }

    public static void main(String[] args) {

    }
}
```

## Conversions implicites

### 1.2 Exercice

Indiquez quelles lignes ne compilent pas :

```
1 short s = 10;
2 byte b = s;
3 int i = s;
4 long l = s;
```

### 1.3 Exercice

Indiquez quelles lignes ne compilent pas :

```
1 short s = 10;
2 s = s + s;
3 s = s * 2;
```

### 1.4 Exercice

```
1 short s = Short.MAX_VALUE; // 32767
2 int i = s++;
```

Que vaut i à la fin de l'exécution ?

et ici:

```
1 short s = Short.MAX_VALUE; // 32767
2 int i = ++s;
```

### 1.5 Exercice

On dispose des méthodes suivantes :

```
1 void g(int n, float x) { ... };
2 void h(short s) { ... };
```

Et les déclarations suivantes

```
1 int i;
2 byte b;
3 float f;
4 double d;
```

Spécifiez si les appels suivants sont corrects ou non tout en justifiant vos réponses:

```
1 g(i, f);
2 g(b+1, f);
3 g(b, f);
4 g(i, d);
5 g(i, i);
6 g(i, 2*f);
7 g(i, 2.0*f);
8 h(b);
9 h(b+1);
10 h(5);
11 h(5.0);
```

## Exercices pratiques

### 1.6 Exercice (*args*)

Trouvez deux façons d'afficher les arguments lors de l'appel d'un programme.

```
javac Args.java  
java Arg arg1 arg2 coucou
```

doit afficher

```
arg1  
arg2  
coucou
```