Timothée GINDRE, Cédric VIMONT

Advanced Web Technologies : Questions LAB 1

Question 1.3 :

When INFO is set on log4j.properties, only « INFO » tool displays :

```
MacBook-Air-de-Cedric:LAB1 Cedric$ java -cp target/hello-0.0.1-SNAPSHOT-jar-with]
-dependencies.jar ex01C.Log4jBasics
23:23:35,780  INFO Log4jBasics:9 - INFO: Cool !
MacBook-Air-de-Cedric:LAB1 Cedric$
```

When DEBUG is set, "DEBUG" and "INFO" tools are shown on terminal :

```
MacBook-Air-de-Cedric:LAB1 Cedric$ java -cp target/hello-0.0.1-SNAPSHOT-jar-with]
-dependencies.jar ex01C.Log4jBasics
23:24:25,828 DEBUG Log4jBasics:8 - DEBUG: Cool !
23:24:25,831  INFO Log4jBasics:9 - INFO: Cool !
MacBook-Air-de-Cedric:LAB1 Cedric$
```

Question 1.4 :

The m() method called in (1) is the m() method of class C, as we can see in the second line :

```
MacBook-Air-de-Cedric:LAB1 Cedric$ java -cp target/hello-0.0.1-SNAPSHOT-jar-with]
-dependencies.jar ex01E.C
Je suis dans la méthode m d'une instance de B
Je suis dans la méthode m d'une instance de C
MacBook-Air-de-Cedric:LAB1 Cedric$
```

Normally, "((B)this).m()" should return a method m() in class B.

But as C is the only instantiated class here, ((B)this) refers to C and not to B : that is **overloading**.

Question 1.5 :

The terminal displays the characteristics of the circle C1 :

```
MacBook-Air-de-Cedric:LAB1 Cedric$ java -cp target/hello-0.0.1-SNAPSHOT-jar-with]
-dependencies.jar ex01F.ToStringOverloading
C1 => Circle with center (0,0) and radius 5 (Perimter is 31,42)
MacBook-Air-de-Cedric:LAB1 Cedric$
```

The method toString() is called without even being named because of the println() method.
In fact, println(object x) automatically call the toString() method on the object x in parameter.
If we rename the method toString(), the result is C1 is not converted into string anymore :

```
MacBook-Air-de-Cedric:LAB1 Cedric$ java -cp target/hello-0.0.1-SNAPSHOT-jar-with]
-dependencies.jar ex01F.ToStringOverloading
C1 => ex01F.Circle@7b23ec81
MacBook-Air-de-Cedric:LAB1 Cedric$
```