TP1 : Technologies web avancées

I. Maven

1.3 Exercise 01.C: Logging exercise 1

In the lo4j.properties, change the level from INFO to DEBUG, what happens?

When we change the level from INFO to DEBUG we can see that as when using INFO, we can see the log.info message but also the log.debug message. When logging through DEBUG we are able to receive all the log messages which help the developer when debugging a program.

1.5 Exercise 01.E: Inheritance

Which m() method is called in (1)? *(1) = ((B)this).m();*
Is it the expected result?
Which OOP implementation type fits to that result? (Virtual function overloading)

The method called in (1) is the one from the B class. It is the expected result because we ask the object to act as if it was a B class object. The OOP implementation fitting to that result is the polymorphism principle because of the "Class C extends B".

1.6 Exercise 01.F: toString() overloading

What shows up? Why is the method toString() called without being named?
If you rename the method toString(), what happens?

We can read :

```
C1 => Circle with center (0,0) and radius 5 (Perimter is 31,42)
```

The method toString is called without being named because all Java objects have a toString method which is called when we call the object.
If we rename the method toString it will show :

```
C1 => ex01F.Circle@9d597e1
```

With "ex01F.Circle@9d597e1" being the reference of the object c1 which is given in the default toString method.