

Nicolas SIMOES
Julien VOYAT

Web Advanced – Lab01

Exercise 01C :

```
mvn exec:java -Dexec.mainClass="ex01C.Log4jBasics" -  
Dlog4j.configuration=file:./src/main/resources/log4j.properties
```

En mode INFO :

```
[INFO] --- exec-maven-plugin:1.2.1:java (default-cli) @ hello ---  
17:01:19,907 INFO Log4jBasics:11 - INFO: Cool !  
[INFO] -----  
[INFO] BUILD SUCCESS
```

En mode DEBUG :

```
[INFO] --- exec-maven-plugin:1.2.1:java (default-cli) @ hello ---  
16:58:05,238 DEBUG Log4jBasics:10 - DEBUG: Cool !  
16:58:05,239 INFO Log4jBasics:11 - INFO: Cool !  
[INFO] -----
```

Exercise 01D :

```
mvn exec:java -Dexec.mainClass="ex01D.C"
```

package ex01D;

public class C

```
{  
    //void method1(int i, StringBuffer s)  
    static void method1(int i, StringBuffer s)  
    {  
        i++;  
        s.append("d");  
    }  
  
    public static void main(String [] args)  
    {  
        int i = 0;  
        StringBuffer s = new StringBuffer("abc");  
        //methode1(i, s);  
        method1(i, s);  
        //System.out.println(i=" + i + ", s=" + s); // i=0, s=abcd  
        System.out.println("i=" + i + ", s=" + s); // i=0, s=abcd  
    }  
}
```

Exercice 01E :

```
mvn exec:java -Dexec.mainClass="ex01E.C"
```

```
Je suis dans la méthode m d'une instance de C
```

La méthode m appelée est celle de la classe C.

De prime abord on peut penser que : ((B)this).m(); appellera la méthode m() de la classe B. Ce n'est donc pas le résultat attendu. On constate donc que le cast avec une autre classe ne permet pas d'atteindre ses méthodes mais seulement ses attributs.

Cela correspond au "Multi-level inheritance" (héritage à niveaux multiples).

Exercice 01F :

```
mvn exec:java -Dexec.mainClass="ex01F.ToStringOverloading"
```

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

La méthode toString() par défaut est appelé lorsque l'on fait un System.out.println sur un objet d'une classe en écrivant une méthode toString() dans la classe Circle.java on surcharge celle par défaut c'est pour cela qu'elle s'exécute sans être appelée.

Si l'on renome la méthode toString() il n'y a plus surcharge de la méthode par défaut de ce fait la classe de l'objet et son adresse mémoire son affichées à la place :

C1 => ex01F.Circle@2bb6e6dc