

# TP02 - M2 GIL - POA

09 octobre 2017

## Exercices 01:

```
public aspect TestAspect{
    private int count = 0;
    pointcut Q01():

        //Exo01 Q01
        preinitialization(fiche2.classes.*.new(..)) ||
        initialization(fiche2.classes.*.new(..)) ||
        execution(fiche2.classes.*.new(..)) ||
        call(fiche2.classes.*.new(..))||

        //Exo01 Q02+03
        call(* fiche2.classes.*.*(..))||
        //Exo01 Q03
        execution(* fiche2.classes.Test.main(..))||
        //Exo01 Q04
        set(* fiche2.classes.*.*);

    before() : Q01(){
        print();
        System.out.println("Entree : " + thisJoinPointStaticPart);
        count++;
    }

    after() : Q01(){
        count--;
        print();
        System.out.println("Sortie : " + thisJoinPointStaticPart);
    }
    public void print(){
        for(int i = 0; i < count; i++)
            System.out.print("| ");
    }
}
```

---

```

Entree : execution(void fiche2.classes.Test.main(String[]))
| Entree : call(fiche2.classes.B())
| | Entree : preinitialization(fiche2.classes.B())
| | | Entree : call(int fiche2.classes.C.m())
| | | Sortie : call(int fiche2.classes.C.m())
| | | Entree : call(int fiche2.classes.C.p(int))
| | | Sortie : call(int fiche2.classes.C.p(int))
| | Sortie : preinitialization(fiche2.classes.B())
| | Entree : preinitialization(fiche2.classes.A(int))
| | Sortie : preinitialization(fiche2.classes.A(int))
| | Entree : initialization(fiche2.classes.A(int))
| | | Entree : execution(fiche2.classes.A())
| | | | Entree : set(String fiche2.classes.A.s)
| | | | Sortie : set(String fiche2.classes.A.s)
| | | Sortie : execution(fiche2.classes.A())
| | | Entree : execution(fiche2.classes.A(int))
| | | | Entree : set(int fiche2.classes.A.i)
| | | | Sortie : set(int fiche2.classes.A.i)
| | | Sortie : execution(fiche2.classes.A(int))
| | Sortie : initialization(fiche2.classes.A(int))
| | Entree : initialization(fiche2.classes.B())
| | | Entree : execution(fiche2.classes.B(int))
| | | | Entree : set(int fiche2.classes.B.j)
| | | | Sortie : set(int fiche2.classes.B.j)
| | | Sortie : execution(fiche2.classes.B(int))
| | | Entree : execution(fiche2.classes.B())
| | | Sortie : execution(fiche2.classes.B())
| | Sortie : initialization(fiche2.classes.B())
| | Sortie : call(fiche2.classes.B())
| -----
| Entree : call(fiche2.classes.A())
| | Entree : preinitialization(fiche2.classes.A())
| | Sortie : preinitialization(fiche2.classes.A())
| | Entree : initialization(fiche2.classes.A())
| | | Entree : execution(fiche2.classes.A())
| | | | Entree : set(String fiche2.classes.A.s)
| | | | Sortie : set(String fiche2.classes.A.s)
| | | Sortie : execution(fiche2.classes.A())
| | Sortie : initialization(fiche2.classes.A())
| | Sortie : call(fiche2.classes.A())
| -----
Sortie : execution(void fiche2.classes.Test.main(String[]))

```

## **Exercices 02 :**

1. Le point de jonction de pré initialisation d'une expression  $E_i$  dépend du constructeur qui va être appelé.  
Exp : si on appelle C1 alors on va parcourir tous le code pour arriver à C3 ou on trouve super().
2. Le point de jonction de l'initialisation d'une expression  $E_i$  dépend du constructeur qui va être appelé.  
Exp : si on appelle C3 alors on va avoir  $x = i$  ;  $y = j$  ; et  $y = i$ .

3. Le point de jonction de l'exécution d'une expression  $E_i$  contient toutes les instructions trouvant à l'intérieur du constructeur appelé.

Exp : si on appelle C2 alors on va avoir  $y = i$ .