



AspectJ: Adendos

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[Adendo]

- Especifica o código que será executado quando um ponto de corte for atingido
- O adendo pode ser executado
 - Antes do ponto de corte (*before*)
 - Depois do ponto de corte (*after*)
 - Antes e depois do ponto de corte (*around*)

[Definindo um Adendo]

```
pointcut logCredito() :  
    call (* Conta.*creditar(double));  
  
after() : logCredito() {  
    System.out.println("ocorreu um credito");  
}
```

O código do adendo será executado após cada chamada ao método creditar

Variações do Adendo *After*

- **After returning**

- É interceptado quando o método retorna normalmente sua execução

- **After throwing**

- É interceptado quando o método retorna uma situação excepcional (*exception*)

[Exemplos de adendo *After*]

```
poincut debitos():  
    call(* Conta.debitar(..));  
  
after() returning: debitos(){  
    System.out.println("debito deu certo!");  
}  
  
after() throwing: debitos(){  
    System.out.println("debito deu errado!");  
}
```

[Expondo o Contexto]

- Pontos de corte podem expor alguns valores para a execução do adendo
 - **args()** parâmetros de chamadas de métodos
 - **target()** objeto que recebe a chamada do método ou acesso ao atributo
 - **this()** objeto que efetua a chamada ao método ou acesso ao atributo

[Exemplo de Uso (*target*)]

```
pointcut logCredito(Conta c) :  
    call (* Conta.creditar(double)) &&  
    target(c);
```

Conta que recebeu a chamada

```
after(Conta c) : logCredito(c) {  
    System.out.println("ocorreu credito");  
    System.out.println("num: "+c.getNumero());  
}
```

Imprime o número da conta

[Exemplo de Uso (*args*)]

```
pointcut logCredito(Conta c, double v) :  
    call (* Conta.creditar(double)) &&  
    target(c) && args(v);
```

Valor do crédito (parâmetro de creditar)

```
after(Conta c, double v) : logCredito(c, v) {  
    System.out.println("ocorreu credito");  
    System.out.println("num: " + c.getNumero());  
    System.out.println("valor: " + v);  
}
```

Imprime o valor creditado

[Adendo do tipo *Around*]

- Substitui o ponto de junção interceptado
 - **proceed()** permite executar o ponto interceptado

```
pointcut logDebito(Conta c,double v):
    call (* Conta.debitar(double)) &&
    target(c) && args(v);

void around(Conta c,double v): logDebito(c,v) {
    if(v > c.getSaldo())
        System.out.println("Sem saldo!");
    else
        proceed(c,v);
}
```

[Exemplo de Aspecto *Logging*]

```
public aspect LogContas {  
    pointcut logCredito():  
        call (* Conta.creditar(double));  
  
    pointcut logDebito():  
        call (* Conta.debitar(double));  
  
    after (): logCredito() {  
        System.out.println("ocorreu um credito");  
    }  
  
    after () returning: logDebito() {  
        System.out.println("ocorreu um debito");  
    }  
}
```

Bibliografia Principal

- R. LADDAD. **AspectJ in Action**, 2^a Ed. 2010.
 - Part 1 Understanding AOP and AspectJ
- Sergio Soares. **Programação Orientada a Aspectos com AspectJ**. Minicurso CBSOft 2010.