

Aspect-oriented Programming in Spring

Tilo Fischer

25. Mai 2011

Beispiel in OOP

Aspect-oriented
Programming in
Spring

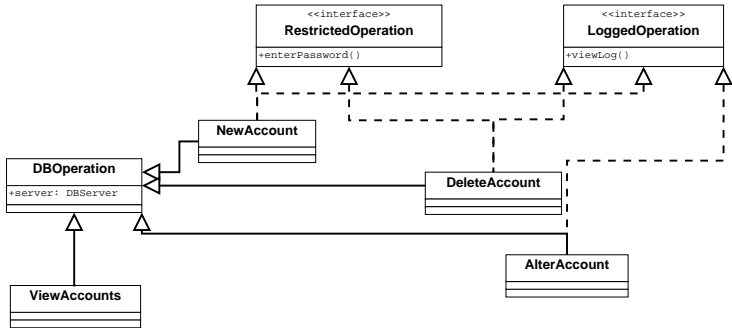
Tilo Fischer

Was ist
AOP?

AOP in
Spring

Konzepte
Code-
Beispiele
Technik

Referenzen



Ist das ganze noch modular?

- Änderungen am Logging oder an der Authentifizierung ziehen Änderungen in mehreren Klassen nach sich.
- Crosscutting Concerns → Anforderungen, die sich über mehrere Module (Klassen) erstrecken
- modulare Implementierung von Concerns kann mit Aspekt-orientierter Programmierung erreicht werden.

Was kann man als Aspekt realisieren?

- Caching
- Transaction-Management
- Persistenz
- Logging
- Fehlerbehandlung
- Profiling
- Authentifizierung
- etc...

Welche Implementierungen gibt es für AOP?

- AspectJ (Java mit modifiziertem Compiler)
- Aspect# (C#)
- AspectL (Lisp)
- FeatureC++
- FLOW3 (PHP)
- Spring AOP (Java, nur für Spring Beans)

1 Was ist AOP?

2 AOP in Spring

- **Konzepte**
- Code-Beispiele
- Technik

3 Referenzen

Terminologie 1

- **Aspekt**
 - Implementierung einer Anforderung
 - eigene Klasse
 - Verknüpft PointCuts mit Advices
- **JoinPoint**
 - Punkt im Programmablauf, der für einen Aspekt geeignet ist
 - in Spring Aufruf einer Methode

Terminologie 2

- PointCut
 - Bestimmt eine Untermenge der JoinPoints
- Advice
 - Code der vor und/oder nach einem JoinPoint ausgeführt wird
- Introduction
 - Variablen und Methoden des Aspekts

Welche Advice-Arten gibt es?

- Before Advice
- After returning Advice
- After throwing Advice
- After (finally) Advice
 - Wird nach unabhängig davon ausgeführt, ob die Methode normal oder mit Exception endet
- Around Advice
 - Einziger Advice, der die Ausführung der Original-Methode verhindern kann

Welche Möglichkeiten bieten PointCuts?

- PointCutDesignators
 - execution, within, this, target, args
 - Wahlweise nur mit Annotationen
- Regex auf Klassennamen
- Regex auf Methodennamen
- geschachtelte PointCuts

1 Was ist AOP?

2 AOP in Spring

- Konzepte
- **Code-Beispiele**
- Technik

3 Referenzen

Aspekte, PointCuts, Advices

```
@Aspect
public class AroundExample {
    @Around("com.xyz.myapp.SystemArchitecture.
        businessService()")
    public Object doBasicProfiling(
        ProceedingJoinPoint pjp) throws
        Throwable {
        // start stopwatch
        Object retVal = pjp.proceed();
        // stop stopwatch
        return retVal;
    }
}
```

Mehr PointCuts

```
@Pointcut("execution(public * *(..))")  
private void anyPublicOperation() {}
```

```
@Pointcut("within(com.xyz.someapp.  
trading..*))")  
private void inTrading() {}
```

```
@Pointcut("anyPublicOperation() &&  
inTrading()")  
private void tradingOperation() {}
```

Spring AOP in XML

```
<aop:config>
  <aop:aspect id="myAspect" ref="aBean">
    <aop:pointcut id="businessService"
      expression="execution(*_com.xyz.
        myapp.service.*.*(..))_and_ this
        (service)"/>
    <aop:before pointcut-ref="
      businessService" method="monitor"/>
    ...
  </aop:aspect>
</aop:config>
```

Wie kann man die Reihenfolge der Aspekte bestimmen?

```
@Aspect
public class ConcurrentOperationExecutor
    implements Ordered {
    private int order = 1;
    public int getOrder() {
        return this.order;
    }
    public void setOrder(int order) {
        this.order = order;
    }
    // ...
}
```

Wie kann man die Reihenfolge der Aspekte bestimmen?

Die Original-Methode wird von dem Advice mit der geringsten Präzedenz (`getOrdered()`) direkt aufgerufen.

Diese Advice-Methode wird von dem Advice mit der nächsthöheren Präzedenz aufgerufen, usw...

- `HighOrderedAdviceBefore()`
 - `LowOrderedAdviceBefore()`
 - `OriginalMethode()`
 - `LowOrderedAdviceAfter()`
- `HighOrderedAdviceAfter()`

1 Was ist AOP?

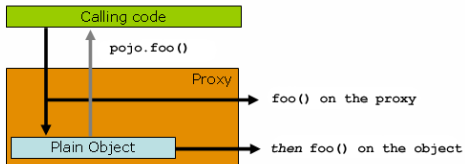
2 AOP in Spring

- Konzepte
- Code-Beispiele
- **Technik**

3 Referenzen

Wie wird Spring AOP umgesetzt?

- Spring Main()
 - Proxy-Klasse für Beans
 - Plain Old Java Object



Wo kann man mehr über Spring AOP erfahren?

Aspect-
oriented
Program-
ming in
Spring

Tilo Fischer

Was ist
AOP?

AOP in
Spring

Konzepte
Code-
Beispiele
Technik

Referenzen

Spring AOP

<http://static.springsource.org/spring/docs/2.5.x/reference/aop.html>

AspectJ Pointcut Expressions

<http://www.eclipse.org/aspectj/doc/released/progguide/index.html>