

Bir Aspect'in Anatomisi



Bir Aspect'in Anatomisi (AspectJ İfadeleri ile)



```
public aspect LoggingAspect {  
    private Logger logger =  
        Logger.getLogger(LoggingAspect.class.getName());
```

Pointcut expression, ilgili join point(ler)i kod içinde yakalayan regular expression benzeri ifadedir

```
    pointcut allMethodCalls() : execution(* *(..));
```

```
    Object around() : allMethodCalls() {
```

```
        try {
```

```
            logger.info(thisJoinPoint.getSignature().getName() + "  
entered");
```

```
            return proceed();
```

```
        } finally {
```

```
            logger.info(thisJoinPoint.getSignature().getName() + "  
exited");
```

```
        }
```

```
    }
```

```
}
```

Değişik nesnelere dağılmış
fonksiyonun **tek bir noktada**
toplanmış halidir

Join point: programın çalışma
esnasındaki bir anıdır

Metoda girmeden önceki an
metot çıkışı

Bir exception fırlatılması anı, bir field'a
erişim anı

advice

Belirli bir join point'de
herhangi bir aspect
tarafından
gerçekleştirilen
aksiyondur

Bir Aspect'in Anatomisi (Annotasyonlar ile)

`@Aspect`

```
public class LoggingAspect {
```

```
    private Logger logger = Logger.getLogger(getClass().getName());
```

```
    @Pointcut("execution(* *(..))")
```

```
    public void allMethodCalls() {
```

```
        @Around("allMethodCalls()")
```

```
        public Object log(ProceedingJoinPoint pjp) throws Throwable {
```

```
            logger.info(pjp.getSignature().getName() + " started");
```

```
            Object result = pjp.proceed();
```

```
            logger.info(pjp.getSignature().getName() + " finished");
```

```
            return result;
```

```
        }
```

```
    }
```

Değişik nesnelere dağılmış
fonksiyonun **tek bir noktada**
toplanmış halidir

Pointcut expression, ilgili join
point(ler)i kod içinde yakalayan
regular expression benzeri
ifadedir

advice

Join point: programın çalışma
esnasındaki bir anıdır
Metoda girmeden önceki an
metot çıkışı
Bir exception fırlatılması anı
Bir field'a erişim anı

Belirli bir join point'de
herhangi bir aspect
tarafından
gerçekleştirilen
aksiyondur

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

