

Pointcut Deep Dive

Eberhard Wolff
<http://ewolff.com>
eberhard.wolff@gmail.com



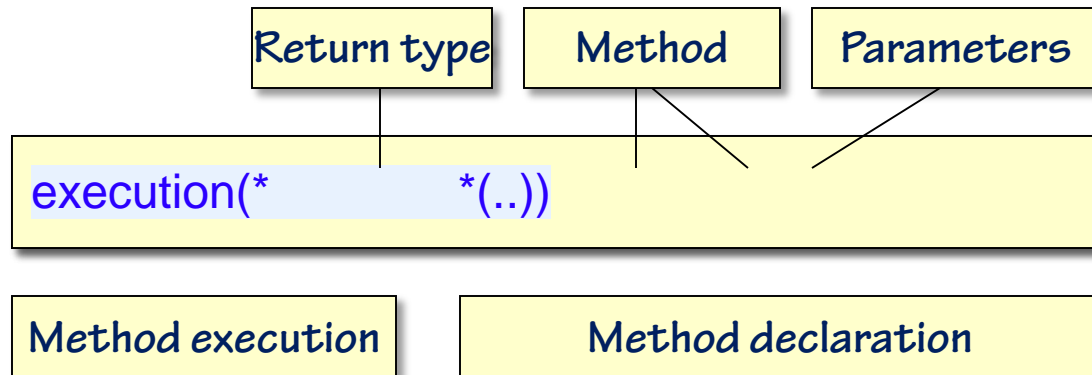
pluralsight 
hardcore developer training

Pointcuts

- So far: add aspects to very specific methods
- ...or all methods
- How can you add advices to specific parts of the system?

Pointcuts

- Define where an aspect is added to the code
- Pointcut language = “SQL for code”
- Recap



- Real power comes through wildcards

Pointcuts: Wildcards

```
execution(* hello())
```

Execution of method hello, no parameters, any return type

```
execution(* hello(int, int))
```

Execution of method hello, two ints as parameters, any return type

Pointcuts: Parameter Wildcards

```
execution(* hello(*))
```

Execution of method hello, one parameter of any type, any return type

```
execution(* hello(..))
```

Execution of method hello, any number of parameters of any type, any return type

Pointcuts: Packages and Classes

```
execution(int com.ewolff.Service.hello(int))
```

Execution of method hello
in class Service
in package com.ewolff
one ints as parameters, int return type

```
execution(* com.ewolff..*Service.*(..))
```

Execution of any method
class name ends in Service
in package com.ewolff or subpackage
any parameters
any return type

Pointcuts: Packages and Classes

```
execution(* *.*(..))
```

Execution of any method

Any parameters

in any class

in the default package

```
execution(* *.*(..))
```

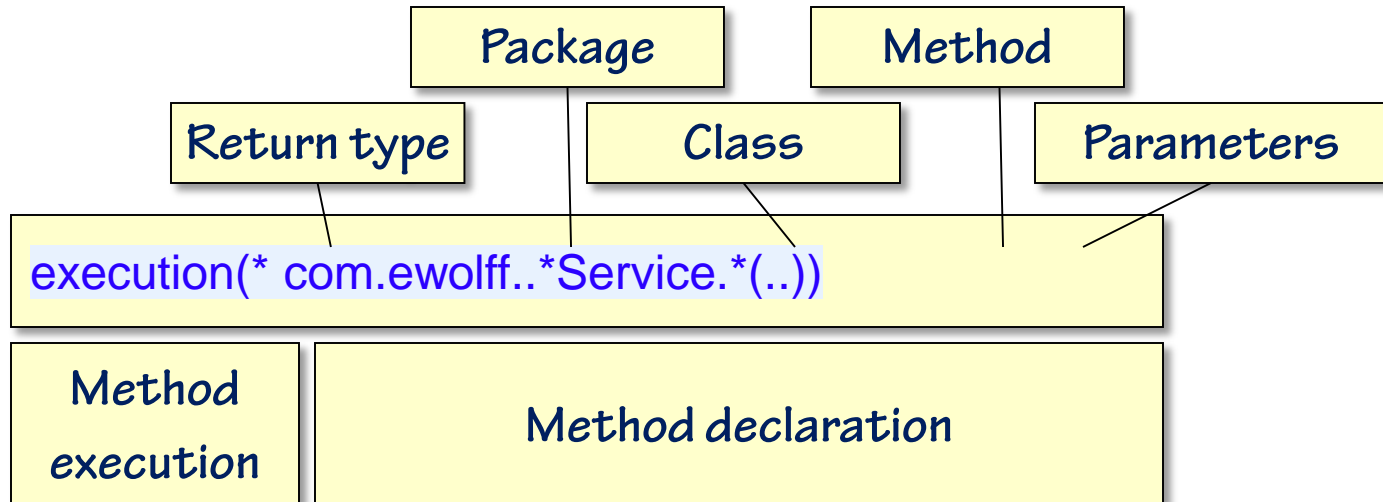
Execution of any method

Any parameters

in any class

in any package
or subpackage

Method Execution Pointcut Recap



Pointcuts: Annotations

```
execution(@com.ewolff.Annotation * *(..))
```

Method must be annotated

Note: fully qualified class name of the Annotation

```
execution(* (@com.ewolff.Annotation *).*(..))
```

Class must be annotated

Spring Bean Names as Pointcuts

```
bean(*Service)
```

Method in Spring Bean

Bean name ends in ...Service

Bean name default: class name (small caps)

Java Config: @Bean method name

Annotation: parameter to @Component, @Service, @Repository

XML: name / id attribute of bean element

Pointcuts and Boolean Operators

- Pointcuts can be combined using boolean operators
- i.e. `||`, `&&`, `!`

```
execution(* service.*(..)) || execution(* repository.*(..))
```

@Pointcut

- Problem: Pointcut expression repeated every time a pointcut is used
- How can they be reused?

```
public class MyPointcuts {  
  
    @Pointcut("execution(@annotation.Trace * *(..))")  
    public void traceAnnotated() {  
    }  
    Use @Pointcut annotation  
}  
    Note: Method's purpose is just to be annotated
```

```
@Around("MyPointcuts.traceAnnotated()")  
public void trace(ProceedingJoinPoint proceedingJP )  
    throws Throwable {  
}
```

Sum Up

- **Pointcuts: Where should an advice be added?**
- **Method Execution Pointcut**
 - return type
 - package name
 - class name
 - method name
 - parameters
- **Power through wildcards**
- **Can also use annotations**
- **Bean name pointcut**
- **Boolean operators**
- **Pointcuts can be reused (@Pointcut)**