

Terminology

Aspect	Modularization unit, comparable with Class
Introduction	introduce new method
Joinpoint	Condition to trigger some action
Pointcut	Collection of Joinpoints
Advice	Behavior introduced at Pointcut

Weaving Aspects get weaved into the result bytecode

Aspect

```
public aspect LogAspect { ... }
```

Holds Introductions, Advices and Pointcuts

Introduction

```
public aspect HashableComplex {
    // introduce an interface
    declare parents: Complex implements Comparable;
    // don't forget the class name!
    public int Complex.compareTo( Object o ) {
        ...
    }
    @Override
    public boolean Complex.equals(Object obj) {
        ...
    }
}
```

Can Introduce new Methods and Declarations

Advice

before

after **alternative:** after returning, after throwing

around **use proceed()** to execute wrapped joinpoint

Pointcut Args

target(p) bind cut target

args(vals) bind Pointcut arguments

```
pointcut setter(Point p, int newval):
    target(p) &&args(newval)
```

Joinpoint

```
package hello;
public aspect WorldAspect{
    public pointcut mainOperation():
        execution (void hello.World.main(String[]));
        //execution (void.*());
    before(): mainOperation() {
        println( "Hello aspect world!" );
    }
    after(): mainOperation() {
        println( "Bye aspect world!" );
    }
}
```

Pointcut Syntax

Wildcards

* any characters, but not "."

.. any characters, also "."

+ Any Subclass / Interface

Logical

||, &&, !

Primitive Pointcuts

call(void Foo.m(int)) call of a Method

execution(Foo(..) throws IOException) execution of any Method throwing an Exception

call Foo.new(..) any Constructors of Foo

initialization(Foo.new(-int)) initialisation by a special constructor

handler(IOException+) handling of any IOException

get(int Foo.bar) or set(int Foo.bar)

within(Foo) limit to foo class

* call vs execution: think of "caller" vs "executing Object"

thisJoinPoint

getArgs(): Object[] args

getSignature()

getSourceLocation()

thisJoinPointStaticPart equals jp.getStaticPart()

thisJoinPoint is available inside the Advice