**AOP - Pointcut Expressions Overview**

**Pointcut**:

Pointcut is a predicate expression for where advice should be applied.

**Pointcut Expression Language**:

* Spring AOP uses AspectJ's poincut expression language.
* We will start with execution pointcuts
  + Applies to execution of methods

**Match on Method Name**:

**Pointcut Expression Language**:

**execution(modifiers-pattern? return-type-pattern declaring-type-pattern?**

**method-name-pattern(param-pattern) throws-pattern?)**

* The pattern is optional if it has “?”
* Patterns can use wildcards
* For the patterns, can use wildcard: \* (matches on everything)

**Pointcut Expression Examples**:

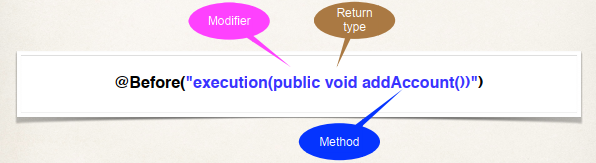
1. Match only **addAccount()** method in **AccountDAO class**

**@Before**("execution(public void com.luv2code.aopdemo.dao.AccountDAO.addAccount())"**)**



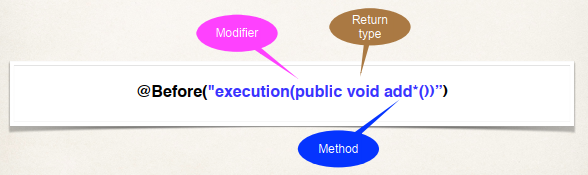
1. Match any **addAccount()** method in **any** class

**@Before**("execution(public void addAccount())"**)**



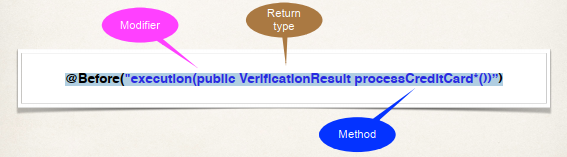
1. Match methods **starting** with **add** in any class

**@Before**("execution(public void add\*())"**)**



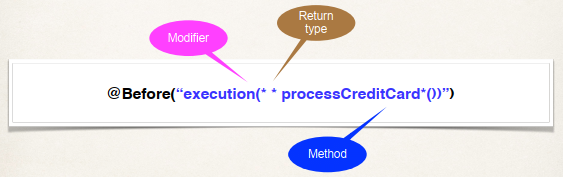
1. Match methods starting with **processCreditCard** in any class

**@Before**("execution(public VerificationResult processCreditCard\*())")



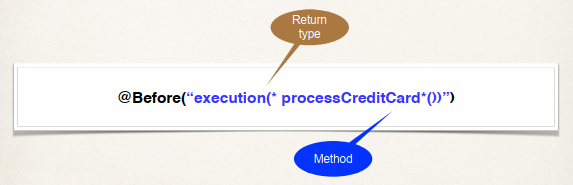
1. Use wildcards on modifier and return type

**@Before**("execution(\* \* processCreditCard\*())")



1. Modifier is optional … so we don’t have to list it

**@Before**("execution(\* processCreditCard\*())")



AOP - Pointcut Expressions Overview