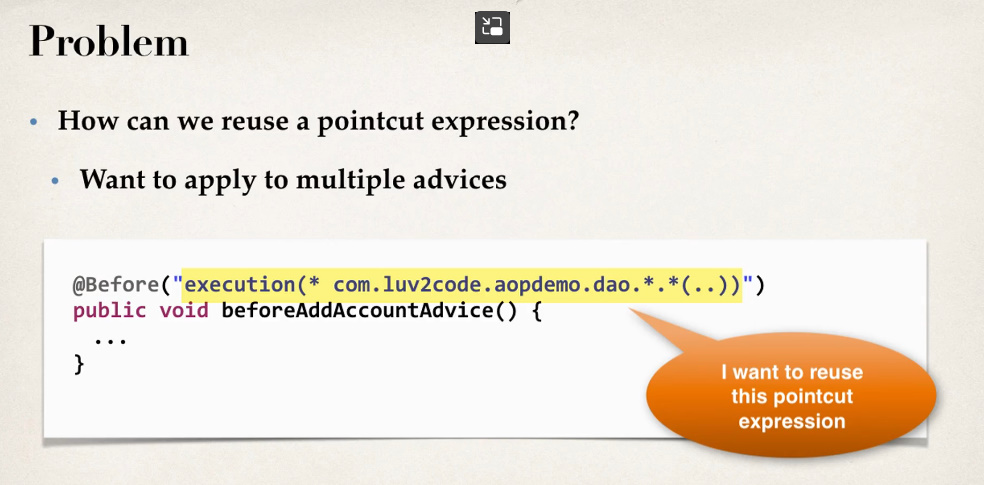
**Necesitate**

Uneori s-ar putea sa avem nevoi sa folosim un point cut expression pentru mai multe advices, si copierea lor nu e o idee prea buna, caci s-ar putea sa modificam prototypul metodei si asa va trebui sa modificam si pointcut expression peste tot.



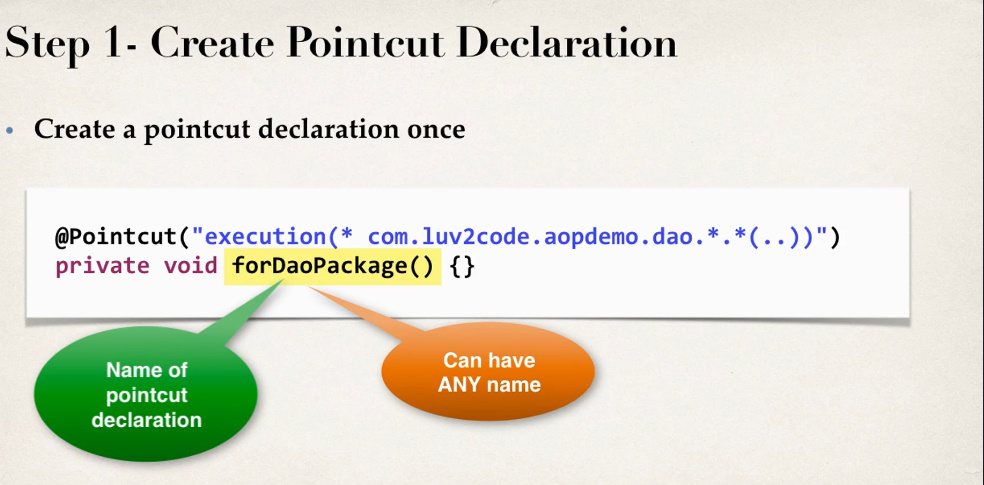
Solutie – cream un pointcut declaration si o folosim oriunde vrem.

**@PointCut**

* Pentru a rezolva problema de mai sus, cream un pointcut declaration. Pentru asta trebuie sa facem urmatorii pasi:

1. Folosim @PointCut(pointCut expression) in Component Aspect class
2. Cream o metoda ce returneaza void fara corp, doar cu {}:

public void oriceNume() {}



Metoda e aici ca un fel de variabila. Spring face in asa fel incat la executarea acestei metode sa returneze acest PointCut expression, desi metoda are void.

1. In @Before, de ex, intre “” executam metoda setata din @PointCut



@Aspect  
@Component  
public class MyDemoLoggingAspect {  
   
 @Pointcut("execution(\* com.java.org.DAO.\*.\*(..))")  
 public void myPointCut() { }  
   
 @Before("myPointCut()")  
 public void beforeAddAccountAdvice(){  
 System.*out*.println("\n=========>>>> Executing @Before advice on addAccount");  
 }

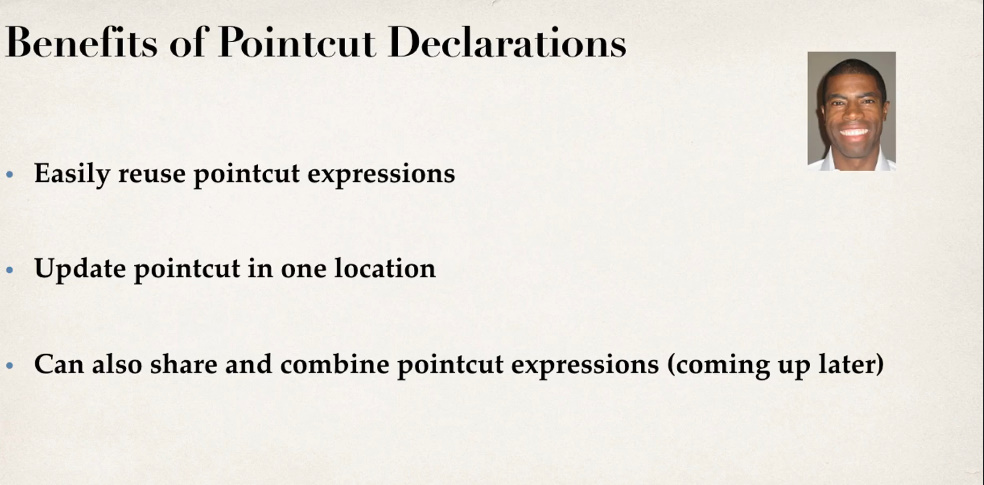


@Before("myPointCut()")  
 public void performApiAnalytics() {  
 System.*out*.println("\n======>>>> Performing API analytics");  
}

}

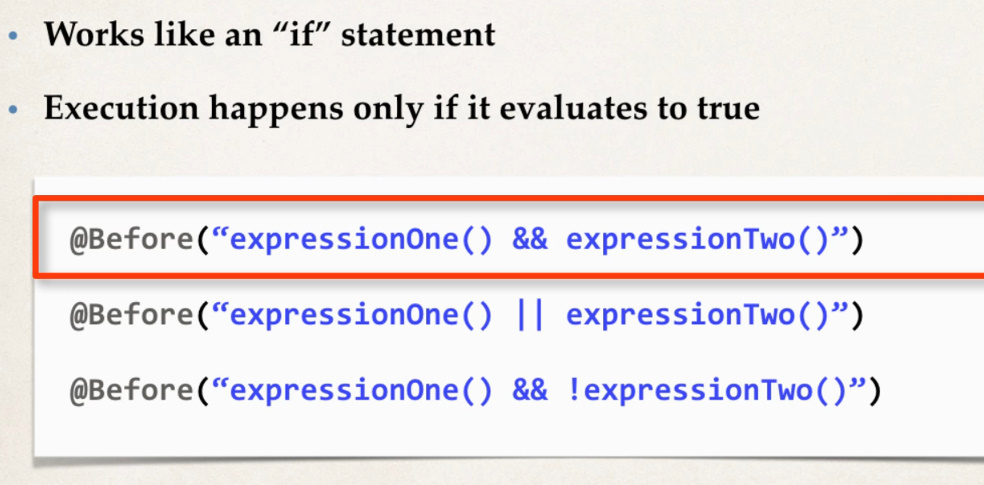
Cand sunt mai multe Advices ce folosesc acelasi PointCut declaration, ordinea executiei lor nu se stie care poate fi.

**Beneficii la PointCuts declaration**



**Combine PointCuts**

* S-ar putea sa avem nevoie sa aplicam mai multe PointCuts expression la un singur advice.
* S-ar putea sa vrem sa executam un advice numai in anumite conditii specifice.
* De ex, poate am vrea ca un advice sa fie executat pentru toate metodele in afara de getter/setter
* Pentru a combina PointCut expresions, folosim operatori logici: && || !



* Totusi, cel mai bine e ca conditia sa fie scrisa tot intr-un @PointCut
* Mai jos cream un serviciu care executa toate metodele cu exceptia celor care incep cu get si set
* @Aspect  
  @Component  
  public class MyDemoLoggingAspect {  
    
   @Pointcut("execution(\* com.java.org.DAO.\*.\*(..))")  
   private void myPointCut() { }  
    
   @Pointcut("execution(\* com.java.org.DAO.\*.set\*(..))")  
   private void setter() { }  
    
   @Pointcut("execution(\* com.java.org.DAO.\*.get\*(..))")  
   private void getter() { }  
    
   @Pointcut("myPointCut() && !(setter() || getter())")  
   private void pointCut() { }  
    
    
   @Before("pointCut()")  
   public void beforeAddAccountAdvice(){  
   System.*out*.println("\n=========>>>> Executing @Before advice on addAccount");  
   }  
    
   @Before("pointCut()")  
   public void performApiAnalytics() {  
   System.*out*.println("\n======>>>> Performing API analytics");  
   }  
  }