

- cross-cutting concerns (сквозная функциональность) - это действия, выполняемые во многих местах системы для решения одних и тех же задач (напр. логгирование, обработка исключений и пр.).
- advice (совет) - какие действия нужно выполнить, когда их нужно выполнить (до вызова метода, после вызова метода, до и после, только если выброшено искл и пр.).
- join points - это все точки выполнения программы, где можно применить совет (вызов метода, изменение поля и т.п.).
- pointcuts - это набор join point в которых применяется данный совет.

- аспект = совет + pointcuts

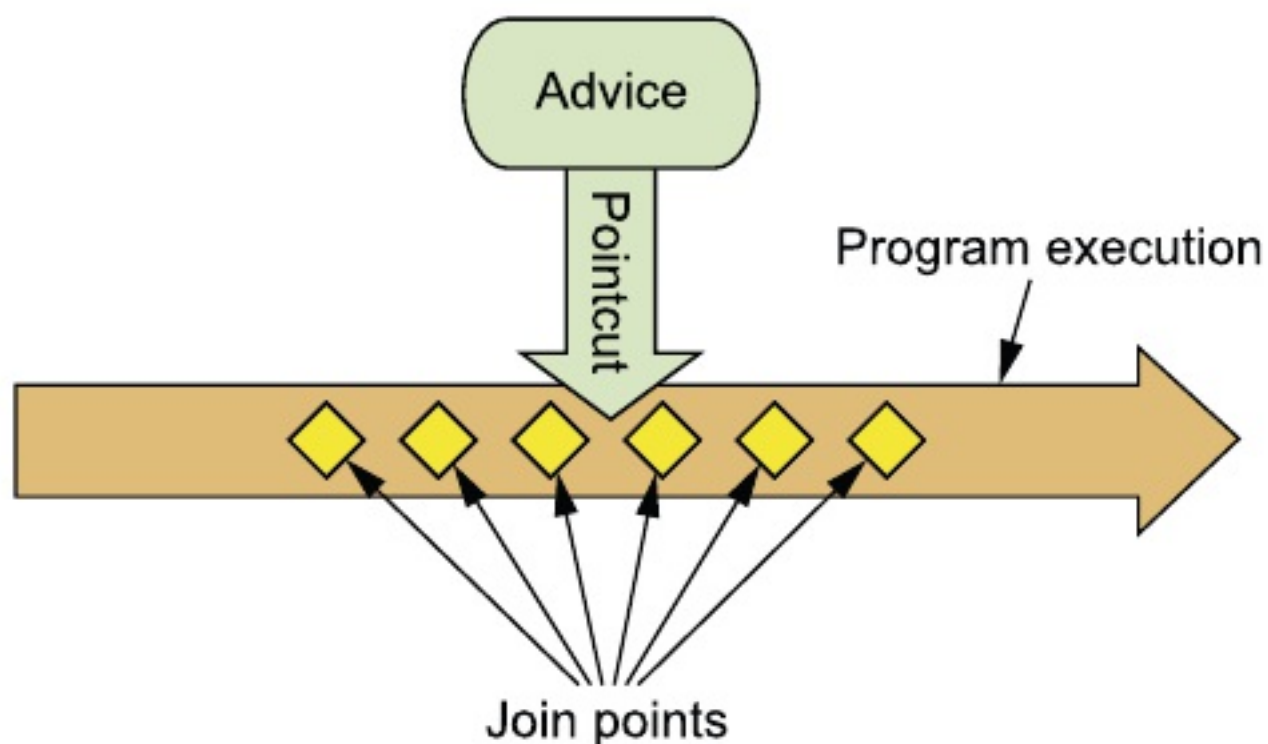


Figure 4.2 An aspect's functionality (advice) is woven into a program's execution at one or more join points.

- introduction - это добавление новых методов и атрибутов в существующие классы.
- weaving - это процесс применения аспектов к объектам, в spring создается новый прокси-объект, содержащий аспект и собственно рабочий объект.

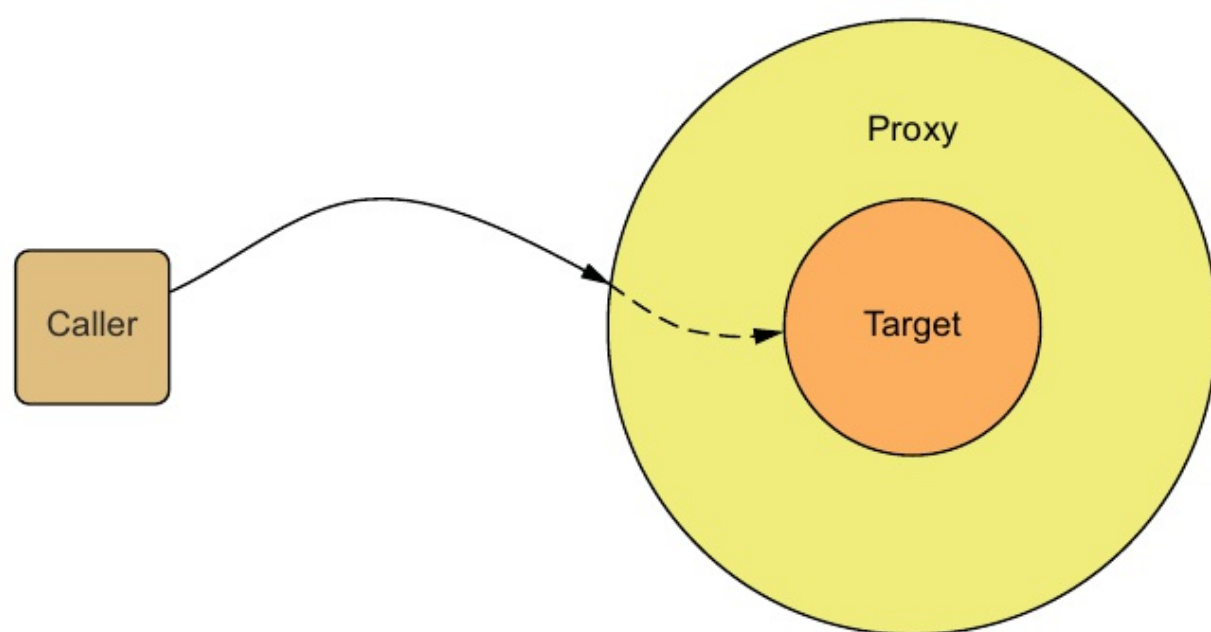
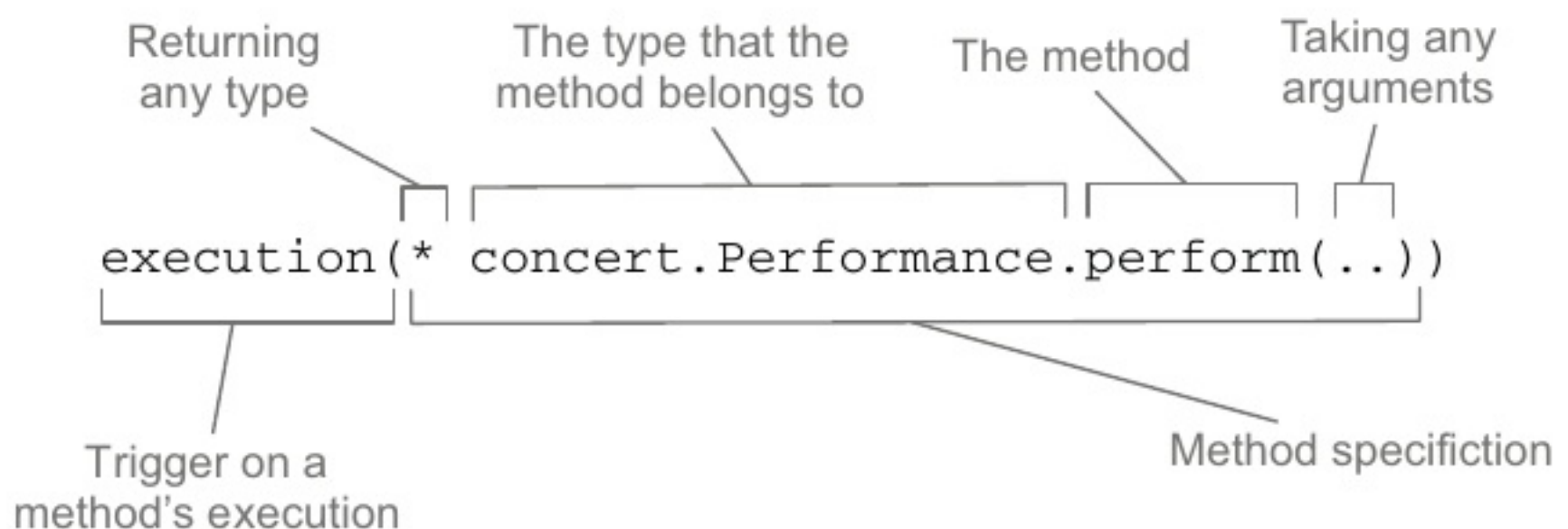


Figure 4.3 Spring aspects are implemented as proxies that wrap the target object. The proxy handles method calls, performs additional aspect logic, and then invokes the target method.

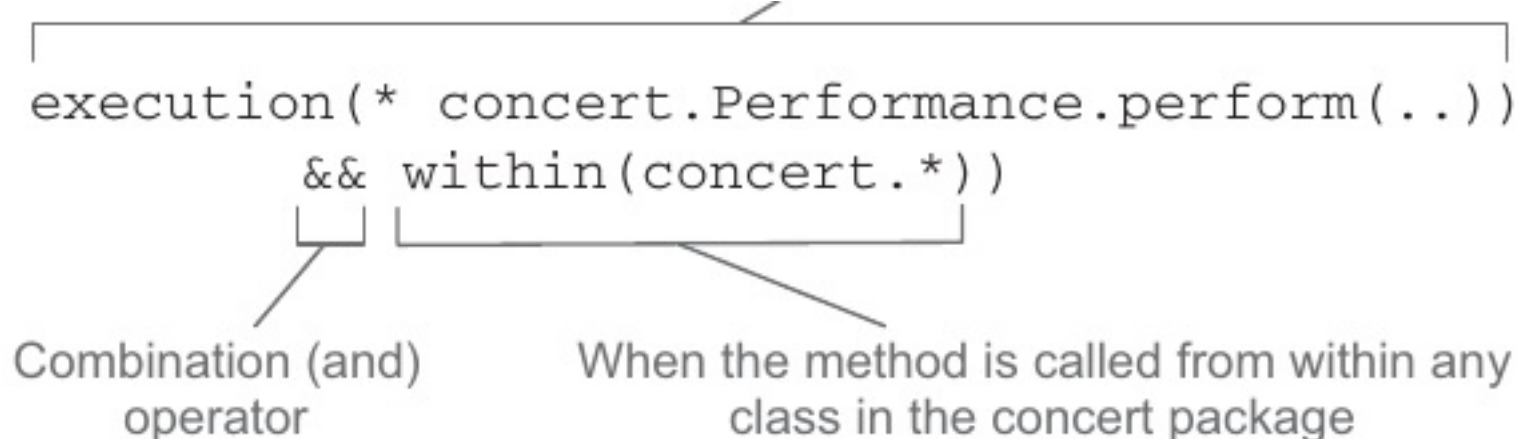
Pointcuts (выборки join points)

AspectJ designator	Description
<code>args()</code>	Limits join-point matches to the execution of methods whose arguments are instances of the given types
<code>@args()</code>	Limits join-point matches to the execution of methods whose arguments are annotated with the given annotation types
<code>execution()</code>	Matches join points that are method executions
<code>this()</code>	Limits join-point matches to those where the bean reference of the AOP proxy is of a given type
<code>target()</code>	Limits join-point matches to those where the target object is of a given type
<code>@target()</code>	Limits matching to join points where the class of the executing object has an annotation of the given type
<code>within()</code>	Limits matching to join points within certain types
<code>@within()</code>	Limits matching to join points within types that have the given annotation (the execution of methods declared in types with the given annotation when using Spring AOP)
<code>@annotation</code>	Limits join-point matches to those where the subject of the join point has the given annotation

- П: выберем точку внедрения по:
 1. Полному имени класса (`concert.Performance`)
 2. Имени метода
 3. Возвращаемый тип - любой
 4. Тип аргументов - любой



- П: используя указатель `within` ограничим выборку точек внедрения, будем перехватывать метод только если он вызывается из любого класса в пакете `concert`:



- В aspectJ выборках можно использовать булевые операторы:

```
&& and
|| or
! not
```

- Указатель bean позволяет определить аспект только при вызове метода бином с определенным id:

```
execution(* concert.Performance.perform())
    and bean('woodstock')
```

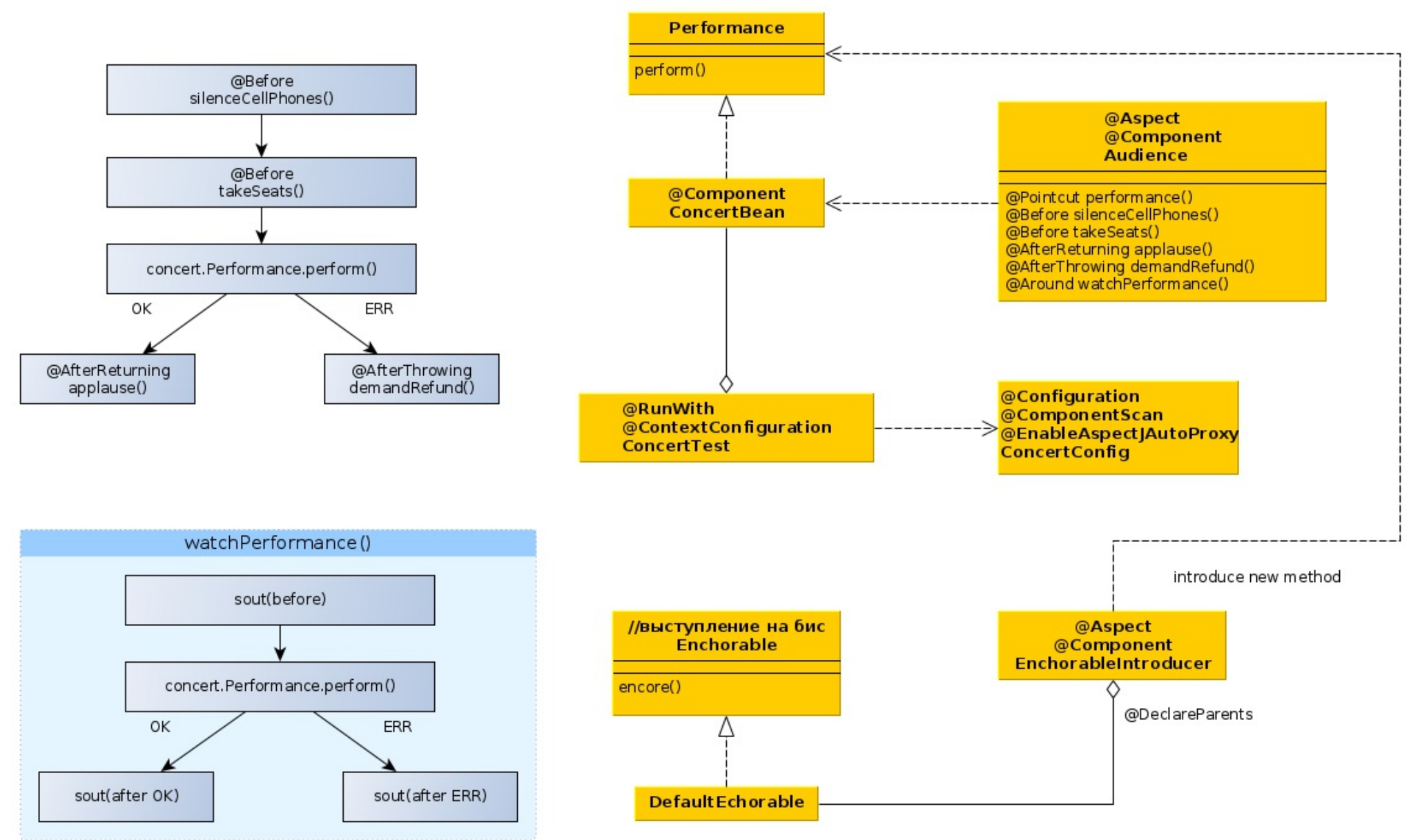
Советы

Table 4.2 Spring uses AspectJ annotations to declare advice methods.

Annotation	Advice
@After	The advice method is called after the advised method returns or throws an exception.
@AfterReturning	The advice method is called after the advised method returns.
@AfterThrowing	The advice method is called after the advised method throws an exception.
@Around	The advice method wraps the advised method.
@Before	The advice method is called before the advised method is called.

- @EnableAspectJAutoProxy - это аннотация, включающая автопоиск аспектов, аналог в xml:

```
<aop:aspectj-autoproxy/>
```

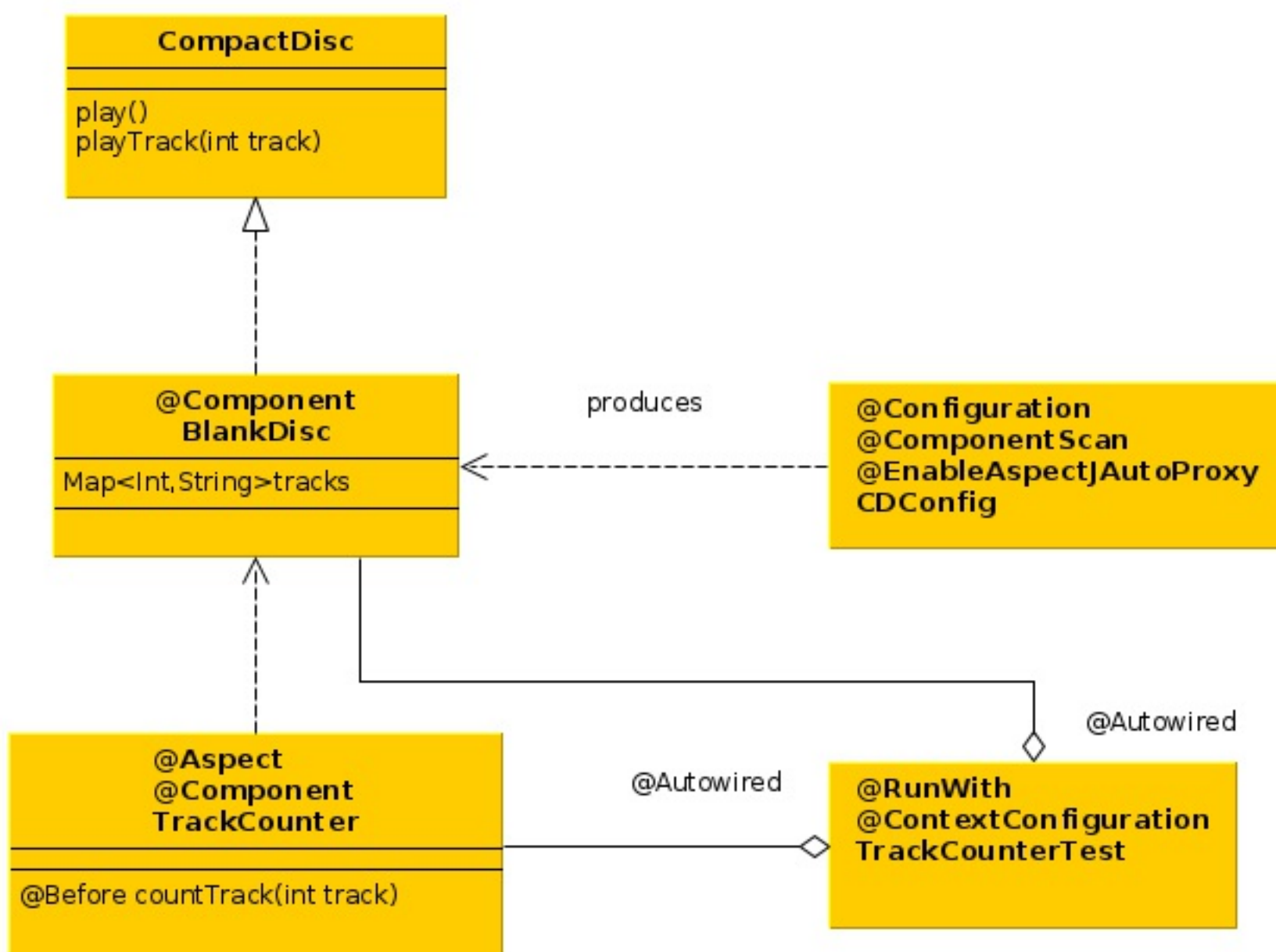


- В советы можно передавать параметры:

Returning any type The type that the method belongs to The method Taking an int argument

```
execution(* soundsystem.CompactDisc.playTrack(int))
&& args(trackNumber)
```

Arguments specification



Добавление новых методов и свойств

- Spring AOP позволяет добавлять новые методы в существующие бины, при вызове дополнительного метода прокси-объект пробрасывает вызов на другой объект, т.о. получается, что один бин состоит из множества объектов.

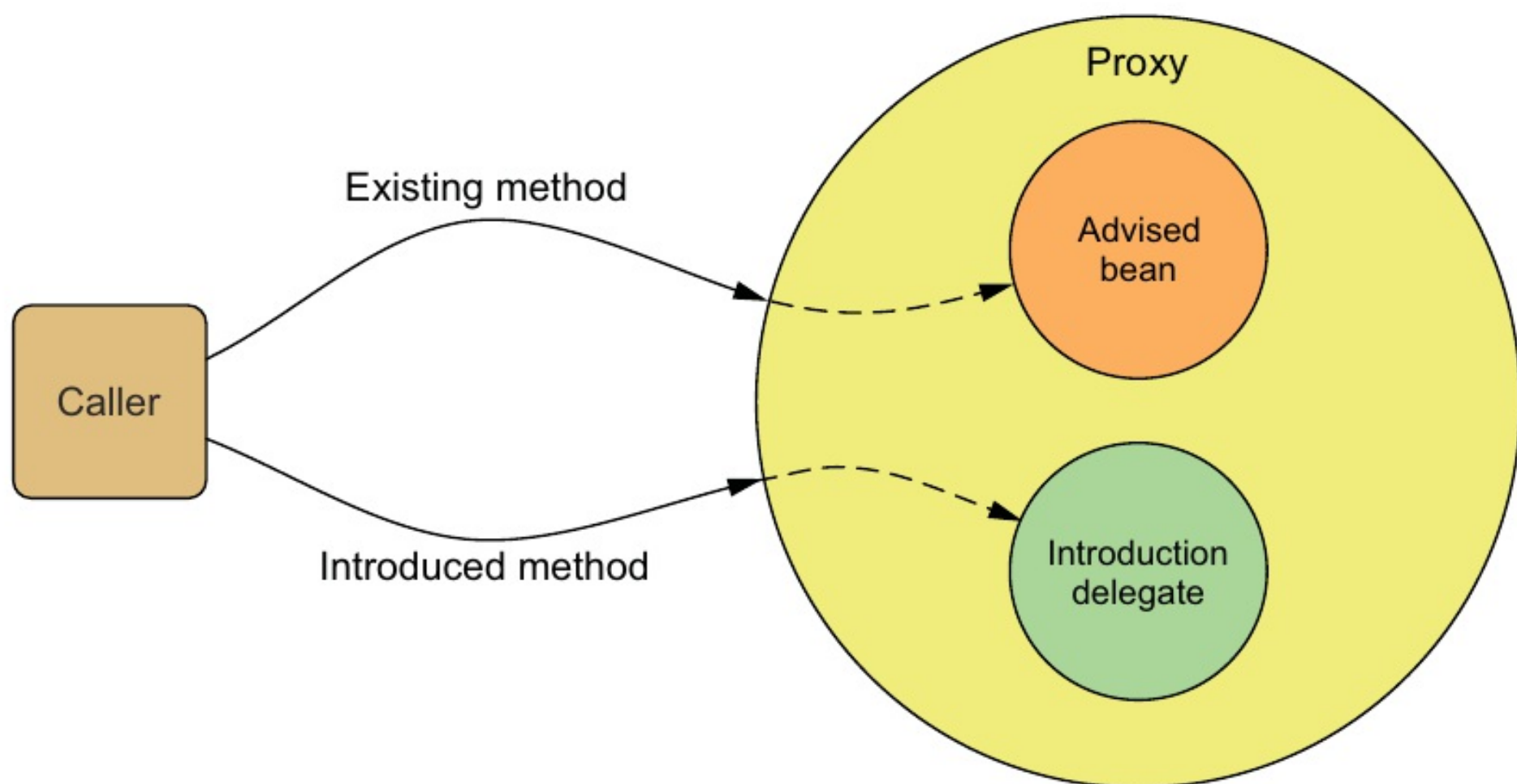


Figure 4.7 With Spring AOP, you can introduce new methods to a bean. A proxy intercepts the calls and delegates to a different object that provides the implementation.