

# Spring Professional Exam Tutorial v5.0

## Question 05

Question 05 - How many advice types does Spring support. Can you name each one? What are they used for? Which two advices can you use if you would like to try and catch exceptions?

## Question 05 - How many advice types does Spring support. Can you name each one? [...]

Spring supports following advice types:

- ▶ @Before - executed before joint point matched by pointcut is executed
- ▶ @After - executed after joint point matched by pointcut is executed
- ▶ @AfterThrowing - executed when exception is thrown from joint point matched by pointcut
- ▶ @AfterReturning - executed after joint point matched by pointcut is executed successfully without any exception
- ▶ @Around - allows you to take full control over joint point matched by pointcut, most powerful advice, allows you to implement all advices from above, you need to call `ProceedingJoinPoint::proceed()` to execute original code

# Question 05 - [...] What are they used for? [...]

Some examples of usage for each Advice type:

- ▶ @Before
  - ▶ Authorization, Security
  - ▶ Logging
  - ▶ Data Validation
- ▶ @After
  - ▶ Logging
  - ▶ Resource Cleanup
- ▶ @AfterThrowing
  - ▶ Logging
  - ▶ Error Handling
- ▶ @AfterReturning
  - ▶ Logging
  - ▶ Data Validation for method result
- ▶ @Around
  - ▶ Transactions
  - ▶ Distributed Call Tracing
  - ▶ Authorization, Security

## Question 05 - [...] Which two advices can you use if you would like to try and catch exceptions?

To catch exceptions you can use two advices:

- ▶ `@AfterThrowing` with `throwing` field set and exception passed as argument
- ▶ `@Around` with `try ... catch` block implemented

