Top Rankers (https://rank.sanfoundry.com/top-rankers-spring/)
Practice Test 1 (https://rank.sanfoundry.com/spring-practice-test-1/)
Practice Test 2 (https://rank.sanfoundry.com/spring-practice-test-2/)
Practice Test 3 (https://rank.sanfoundry.com/spring-practice-test-3/)
Practice Test 4 (https://rank.sanfoundry.com/spring-practice-test-4/)
Practice Test 5 (https://rank.sanfoundry.com/spring-practice-test-5/)
Practice Test 6 (https://rank.sanfoundry.com/spring-practice-test-6/)
Practice Test 7 (https://rank.sanfoundry.com/spring-practice-test-7/)
Practice Test 8 (https://rank.sanfoundry.com/spring-practice-test-8/)
Practice Test 9 (https://rank.sanfoundry.com/spring-practice-test-9/)
Practice Test 10 (https://rank.sanfoundry.com/spring-practice-test-10/)
Mock Test 1 (https://rank.sanfoundry.com/spring-mock-test-1/)
Mock Test 2 (https://rank.sanfoundry.com/spring-mock-test-2/)
Mock Test 3 (https://rank.sanfoundry.com/spring-mock-test-3/)
Mock Test 4 (https://rank.sanfoundry.com/spring-mock-test-4/)
Mock Test 5 (https://rank.sanfoundry.com/spring-mock-test-5/)
Mock Test 6 (https://rank.sanfoundry.com/spring-mock-test-6/)
Mock Test 7 (https://rank.sanfoundry.com/spring-mock-test-7/)
Mock Test 8 (https://rank.sanfoundry.com/spring-mock-test-8/)
Mock Test 9 (https://rank.sanfoundry.com/spring-mock-test-9/)
Mock Test 10 (https://rank.sanfoundry.com/spring-mock-test-10/)

« Prev Page (https://www.sanfoundry.com/spring-questions-answers-aspectj-annotation/)
Next Page (https://www.sanfoundry.com/spring-questions-answers-introduction-behaviour-state-beans/) »

Spring Questions and Answers – Pointcut Definitions

This set of Java Spring Multiple Choice Questions & Answers (MCQs) focuses on “Pointcut Definitions”.

1. PointCut definitions can’t be reused again
a) True
b) False
View Answer

Answer: b
Explanation: Like many other AOP implementations, AspectJ also allows you to define a pointcut independently to be reused in multiple advices.

2. Annotation used to refer poincuts?
a) @Pointcut
b) @PointcutExecution
c) @PointcutBefore
d) None of the mentioned
View Answer

Answer: a

Explanation: In an AspectJ aspect, a pointcut can be declared as a simple method with the @Pointcut annotation.

3. what will be the output of the code snippet?

```
package com.apress.springrecipes.calculator;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Pointcut;
@Aspect
public class CalculatorPointcuts
{
     @Pointcut("execution(* *.*(..))")
     public void loggingOperation() {}
}

package com.apress.springrecipes.calculator;
@Aspect
public class CalculatorLoggingAspect
{
...
@Before("CalculatorPointcuts.loggingOperation()")
public void logBefore(JoinPoint joinPoint)
{
...
}
@AfterReturning(
pointcut = "loggingOperation()",
returning = "result")
public void logAfterReturning(JoinPoint joinPoint, Object result)
{
throw new IllegalArgumentException();
}
@AfterThrowing(
pointcut = "CalculatorPointcuts.loggingOperation()",
throwing = "e")
public void logAfterThrowing(JoinPoint joinPoint, IllegalArgumentException e)
{
...
}
@Around("CalculatorPointcuts.loggingOperation()")
public Object logAround(ProceedingJoinPoint joinPoint) throws Throwable
{
...
}
}
```

a) Runtime Error

b) IllegalArgument Exception

c) BeanCreation Exception

d) None of the mentioned

View Answer

Answer: c

Explanation: When you refer to this pointcut, you have to include the class name as well. If the class is not located in the same package as the aspect, you have to include the package name also.

4. Language used to set various kinds of join points

a) AspectJ pointcut language

b) Java pointcut language

c) XML pointcut language

d) None of the mentioned

View Answer

Answer: a

Explanation: The AspectJ pointcut language is a powerful expression language that can match various kinds of join points.

5. Spring AOP only supports method execution join points for the beans in its IoC container

a) True

b) False

View Answer

Answer: a

Explanation: If you use a pointcut expression out of this scope, an IllegalArgumentException will be thrown.

6. Is the following pointcut expression correct?

execution(* ArithmeticCalculator.*(..))

a) Yes

b) No

c) If every target class is in same package

d) Depends where target class is located

View Answer

Answer: c

Explanation: You can omit the package name if the target class or interface is located in the same package as this aspect.

7. The annotations must be added to the implementation class but not the interface

a) True

b) False

View Answer

Answer: a

Explanation: Annotations must be added to the implementation class but not the interface, as they will not be inherited.

8. Which of the following pattern is used to match bean name?

a) bean(*Calculator)

b) bean(Calculator)

c) bean(com.appress.spring.Calculator)

d) None of the mentioned

View Answer

Answer: a

Explanation: The following pointcut expression matches beans whose name ends with Calculator.

9. Bean name patterns are supported by all configurations(XML,Java,AspectJ)

a) True

b) False

View Answer

Answer: b

Explanation: This pointcut type is supported only in XML-based Spring AOP configurations, not in AspectJ annotations.

10. Expressions which returns Parameters of pointcuts?

a) target

b) args

c) none of the mentioned

d) all of the mentioned

View Answer

Answer: d

Explanation: The expressions target() and args() capture the target object and argument values of the current join point and expose them as pointcut parameters.

11. Are logical operators valid in pointcut expressions?

a) Yes

b) No

View Answer

Answer: a
Explanation: In AspectJ, pointcut expressions can be combined with the operators && (and), || (or), and ! (not).

12. Method which checks if all target classes are matched
a) matches()
b) pair()
c) matchTargetClass()
d) none of the mentioned
View Answer

Answer: a
Explanation: If the matches() method always returns true, all target classes will be matched.

13. Spring supports operations on pointcuts:-
a) notably
b) union
c) intersection
d) all of the mentioned
View Answer

Answer: d
Explanation: Union means the methods that either pointcut matches.
Intersection means the methods that both pointcuts match.
Union is usually more useful.

14. Pointcuts can be composed using:-
a) org.springframework.aop.support.Pointcuts class
b) composablePointcut class
c) all of the mentioned
d) none of the mentioned
View Answer

Answer: c
Explanation: Using the static methods in the org.springframework.aop.support.Pointcuts class, or using the ComposablePointcut class in the same package.

15. Pointcut used to parse an AspectJ pointcut expression string
a) org.springframework.aop.aspectj.AspectJExpressionPointcut
b) org.springframework.aop.aspectj.AspectJExpressionPointcutString
c) org.springframework.aop.aspectj.AspectJExpressionString
d) org.springframework.aop.aspectj.AspectJPointcuttoString
View Answer

Answer: a
Explanation: Since 2.0, the most important type of pointcut used by Spring is org.springframework.aop.aspectj.AspectJExpressionPointcut. This is a pointcut that uses an AspectJ supplied library to parse an AspectJ pointcut expression string.