PREVIOUS ARTICLE
SAP PP Interview Questions and Answers for 2018

NEXT ARTICLE
Codeigniter Interview Questions and Answers for 2 Year Experience

Spring AOP Interview Questions and Answers

BY ANURADHA

March 26, 2018 in INFORMATION TECHNOLOGIES (IT) No Comments 370

1.What is meant by Aspect Oriented Programming (AOP) in Spring?

Aspect Oriented Programming works like Object Oriented Programming. In Object Oriented Programming,

the unit of modularity is Object But in Aspect Oriented Programming the unit of modularity is Aspect. Aspect works as the modularization of concerns known as crosscutting concerns in AOP. AOP framework is pluggable in spring. AOP provides declarative enterprise service and allows users to implement custom aspects.

2.Why is it uses?

Suppose we want to log every method entry and exit. This can be achieved by writing log statements in every method at the start and end. But this will require lot of code work. There are various such tasks like Security which need to be applied across all methods or classes. These are known as cross cutting concerns. AOP addresses the problem of cross-cutting concerns, which would be any kind of code that is repeated in different methods and cannot normally be completely refactored into its own module, like with logging or verification.

3.What are the different implementations of Spring AOP ?

The different implementations of Spring AOP are-
AspectJ
Spring AOP
JBoss AOP

4.Explain different AOP terminologies??

The different AOP terminologies are
Joinpoint: A joinpoint is a candidate point in the Program Execution of the application where an aspect can be plugged in. This point could be a method being called, an exception being thrown, or even a field being modified. These are the points where your aspect's code can be inserted into the normal flow of your application to add new behavior.
Advice: This is an object which includes API invocations to the system wide concerns representing the action to perform at a joinpoint specified by a point.
Pointcut: A pointcut defines at what joinpoints, the associated Advice should be applied. Advice can be applied at any joinpoint supported by the AOP framework. Of course, you don't want to apply all of your aspects at all of the possible joinpoints. Pointcuts allow you to specify where you want your advice to be applied. Often you specify these pointcuts using explicit class and method names or through regular expressions that define matching class and method name patterns. Some AOP frameworks allow you to create dynamic pointcuts that determine whether to apply advice based on runtime decisions, such as the value of method parameters.
Aspect:The key unit of modularity in OOP is the class, whereas in AOP the unit of modularity is the aspect. Aspects enable the modularization of concerns such as transaction management that cut across multiple types and objects.
Weaving:In Spring AOP makes it possible to modularize and separate logging, transaction like services and apply them declaratively to the components Hence programmer can focus on specific concerns. Aspects are wired into objects in the spring XML file in the way as JavaBean. This process is known as 'Weaving'.

5.What is the difference between Spring AOP and AspectJ AOP?

AspectJ is the industry-standard implementation for Aspect Oriented Programming whereas Spring implements AOP for some cases. Main differences between Spring AOP and AspectJ are:

- Spring AOP is simpler to use than AspectJ because we don't need to worry about the weaving process.
- Spring AOP supports AspectJ annotations, so if you are familiar with AspectJ then working with Spring AOP is easier.
- Spring AOP supports only proxy-based AOP, so it can be applied only to method execution join points. AspectJ support all kinds of pointcuts.
- One of the shortcoming of Spring AOP is that it can be applied only to the beans created through Spring Context.

6.What are the different types of Spring Advice ?

The different types of Spring Advice are-
Before advice : Advice that executes before a join point.
After returning advice : Advice to be executed after a join point completes normally.
After throwing advice : Advice to be executed if a method exits by throwing an exception.
After advice : Advice to be executed regardless of the means by which a join point exits.
Around advice : Advice that surrounds a join point such as a method invocation.

7.What are the the types of advice in Spring AOP.

In Spring AOP, types of advice are
Before: Advice that runs before a join point.
After returning: Advice that runs after a join point normal completion.
After throwing: Advice which runs when a methods exits by throwing an exception.
After: Advice that runs after the join point exit by any way.
Around: Advice that runs surrounding to join point. Example method invocation.

8.Define Run-time AOP vs Compile-time AOP ?

The different types of AOPs depending on when they are loaded are-
Source code weaving: Aspect code is injected as source code statements into your application source code. This is some kind of preprocessor approach. No AOP framework in the Java world uses this approach nowadays, but there used to be some in the early days of AOP.
Compile-time weaving: Aspect code is woven into your application by a special compiler.
Binary weaving: Aspect code is woven into existing class files after compilation rather than during compilation.
Load-time weaving (LTW): A weaving agent/library is loaded early when your VM/container is started. It gets a configuration file with rules describing which aspects should be woven into which classes.
Proxy-based LTW: This special LTW form is used by Spring AOP while AspectJ does the previous 3 forms

listed above. It works by creating dynamic proxies (i.e. subclasses or interface implementations) for aspect targets.

9.How to enable @AspectJ Support?
Include the below XML code in application XML
<aop:aspectj-autoproxy/>

10.How to declare aspect in Spring AOP?
Using the below XML snippet
<bean id="myAspect" class="com.javaconnect.MyAspect">
<!– configure properties of aspect here –>
</bean>

11.What are the differences between OOP and AOP ?

Object Oriented Programming (OOP)
1.OOP looks at an application as a set of collaborating objects. OOP code scatters system level code like logging, security etc with the business logic code.
2.OOP nomenclature has classes, objects, interfaces etc.
3.Provides benefits such as code reuse, flexibility, improved maintainability, modular architecture, reduced development time etc with the help of polymorphism, inheritance and encapsulation.

Aspect Oriented Programming (AOP)

1. AOP looks at the complex software system as combined implementation of multiple concerns like business logic, data persistence, logging, security, multithread safety, error handling, and so on. Separates business logic code from the system level code. In fact one concern remains unaware of other concerns.

2. AOP nomenclature has join points, point cuts, advice, and aspects.

3. AOP implementation coexists with the OOP by choosing OOP as the base language. For example: AspectJ uses Java as the base language.

4. AOP provides benefits provided by OOP plus some additional benefits which are discussed in the next question.

12.How to declare a pointcut in Spring AOP?
Find the below code snippet.
@Pointcut("execution(* update(..))")
private void accountUpdate {}

13.What are the supported AspectJ pointcut designators in Spring AOP?

Followings are the AspectJ pointcut designators in Spring AOP. Execution
This
Target
Args

@target
@args
@within
@annotation

14.When to use Spring AOP and when to use full AspectJ?

If we only need to advice the execution of operations on Spring beans then we should use Spring AOP. Spring AOP is simpler than AspectJ. Full AspectJ requires the AspectJ complier in the build process.
In case if we advice objects not to be managed by Spring Container, use AspectJ.

15.What do you understand by Load-time weaving (LTW) in Spring?

Load-time weaving (LTW) is a process of weaving AspectJ aspects into an applications class file when the classes are being loaded in JVM.

Summary	
Reviewer	Gurpreeth
Review Date	2018-03-26
Reviewed Item	Excellent!! Spring AOP Interview Questions and Answers Thank you mytectra for sharing. keep updating more Spring AOP Interview Questions and Answers.
Author Rating	★★★★★

SPRING AOP INTERVIEW QUESTIONS AND ANSWERS