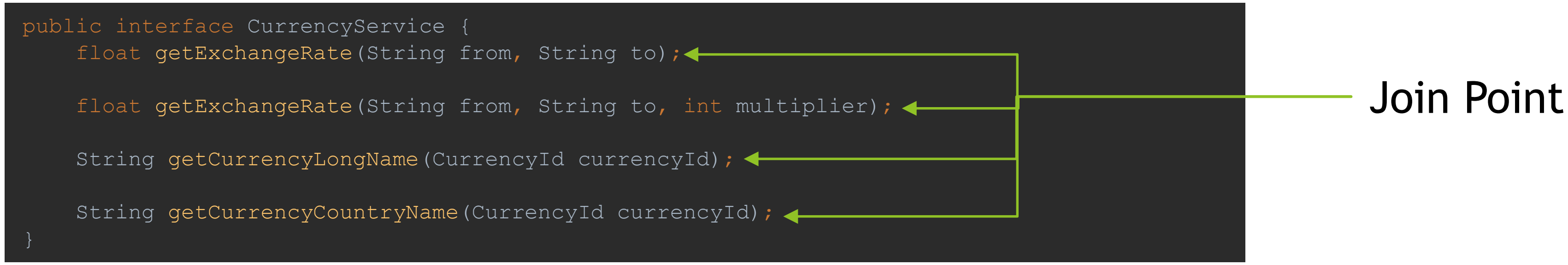
Spring Professional Exam Tutorial v5.0

Question 02

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving?

Join Point in aspect oriented programming is a point in execution of a program in which behavior can be altered by AOP.

In Spring AOP Join Point is always method execution.



```
public interface CurrencyService {
    float getExchangeRate(String from, String to);

    float getExchangeRate(String from, String to, int multiplier);

    String getCurrencyLongName(CurrencyId currencyId);

    String getCurrencyCountryName(CurrencyId currencyId);
}
```

Aspect Oriented Programming concept in general, distinguishes additional Join Points, some of them include:

- Method Execution / Invocation
- Constructor Execution / Invocation
- Reference / Assignment to Field
- Exception Handler
- Execution of Advice
- Execution of Static Initializer / Object Initializer

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving? (cont.)

Pointcut is a predicate used to match join point. Additional code, called Advice is executed in all parts of the program that are matching pointcut. Spring uses the AspectJ pointcut expression language by default.

Example of Pointcut Expressions:

- execution - Match Method Execution

```
execution(* com.spring.professional.exam.tutorial.module02.question02.bls.CurrencyService.getExchangeRate(..))
```

- within - Match Execution of given type or types inside package

```
within(com.spring.professional.exam.tutorial.module02.question02.bls.*)
```

- @within – Match Execution of type annotated with annotation

```
@within(com.spring.professional.exam.tutorial.module02.question02.annotations.Secured)
```

- @annotation – Match join points where the subject of the join point has the given annotation

```
@annotation(com.spring.professional.exam.tutorial.module02.question02.annotations.InTransaction)
```

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving? (cont.)

Example of Pointcut Expressions (cont.):

- bean – Match by spring bean name

```
bean(currency_service)
```

- args – Match by method arguments

```
args(String, String, int)
```

- @args – Match by runtime type of the method arguments that have annotations of the given type

```
@args(com.spring.professional.exam.tutorial.module02.question02.annotations.Validated)
```

- this – Match by bean reference being an instance of the given type (for CGLIB-based proxy)

```
this(com.spring.professional.exam.tutorial.module02.question02.bls.CurrencyService)
```

- target – Match by target object being an instance of the given type

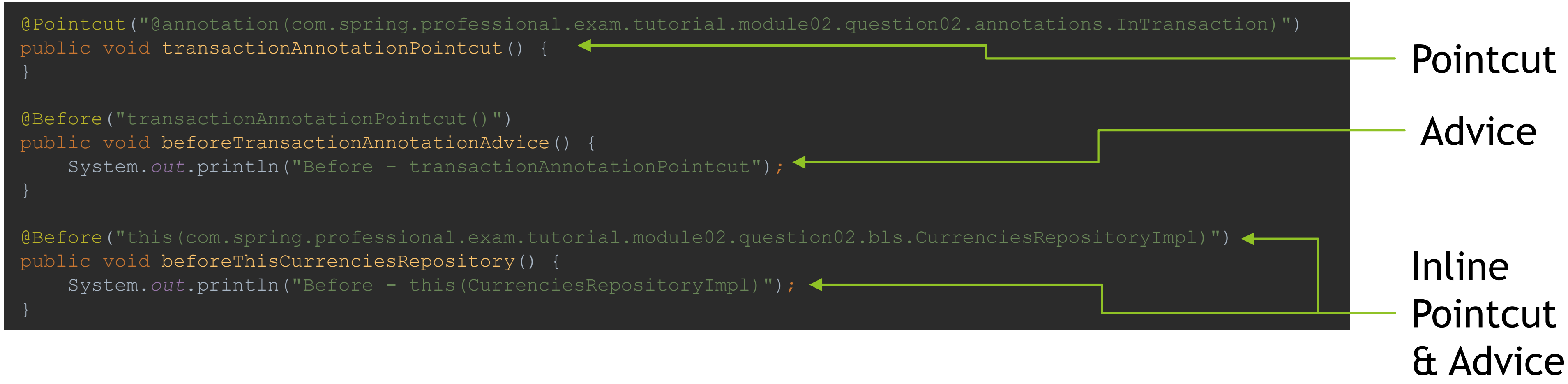
```
target(com.spring.professional.exam.tutorial.module02.question02.bls.CurrencyService)
```

- @target – Match by class of the executing object having an annotation of the given type

```
@target(com.spring.professional.exam.tutorial.module02.question02.annotations.Secured)
```

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving? (cont.)

Advice is additional behavior that will be inserted into the code, at each join point matched by pointcut.



```
@Pointcut("@annotation(com.spring.professional.exam.tutorial.module02.question02.annotations.InTransaction)")
public void transactionAnnotationPointcut() {
}

@Before("transactionAnnotationPointcut()")
public void beforeTransactionAnnotationAdvice() {
    System.out.println("Before - transactionAnnotationPointcut");
}

@Before("this(com.spring.professional.exam.tutorial.module02.question02.bls.CurrenciesRepositoryImpl)")
public void beforeThisCurrenciesRepository() {
    System.out.println("Before - this(CurrenciesRepositoryImpl)");
}
```

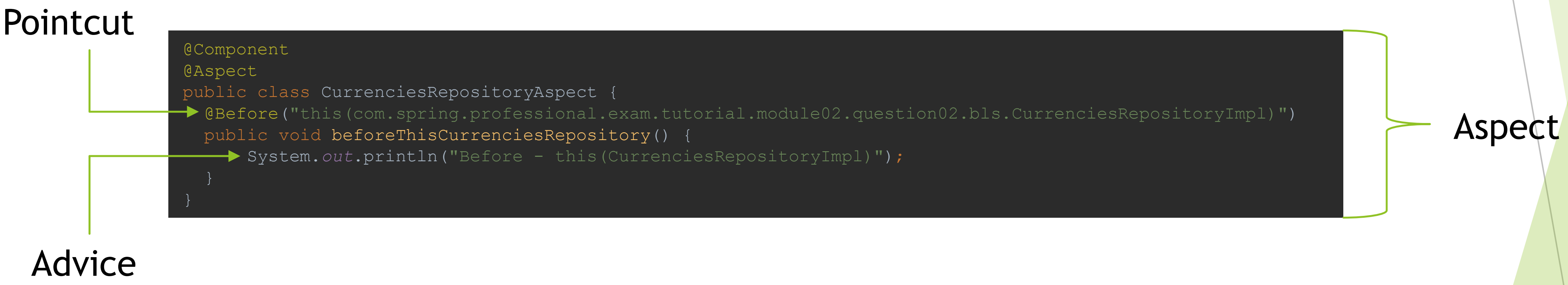
Pointcut

Advice

Inline Pointcut & Advice

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving? (cont.)

Aspect brings together Pointcut and Advice. Usually it represents single behavior implemented by advice that will be added to all join points matched by pointcut.



Pointcut

```
@Component
@Aspect
public class CurrenciesRepositoryAspect {
  @Before("this(com.spring.professional.exam.tutorial.module02.question02.bls.CurrenciesRepositoryImpl)")
  public void beforeThisCurrenciesRepository() {
      System.out.println("Before - this(CurrenciesRepositoryImpl)");
  }
}
```

Aspect

Advice

Question 02 - What is a pointcut, a join point, an advice, an aspect, weaving? (cont.)

Weaving is the process of applying aspects, which modifies code behavior at join points that have matching pointcuts and associated advices. During weaving aspects and application code is combined which enables execution of cross-cutting concerns.

Types of weaving:

- Compile Time Weaving – byte code is modified during the compilation, aspects are applied, code is modified at join points matching pointcuts by applying advices
- Load Time Weaving – byte code is modified when classes are loaded by class loaders, during class loading aspects are applied, code is modified at join points matching pointcuts by applying advices
- Runtime Weaving – used by Spring AOP, for each object/bean subject to aspects, proxy object is created (JDK Proxy or CGLIB Proxy), proxy objects are used instead of original object, at each join point matching pointcut, method invocation is changed to apply code from advice