

SPRING FRAMEWORK COURSE

EXERCISE

TALENT CONTEST V5

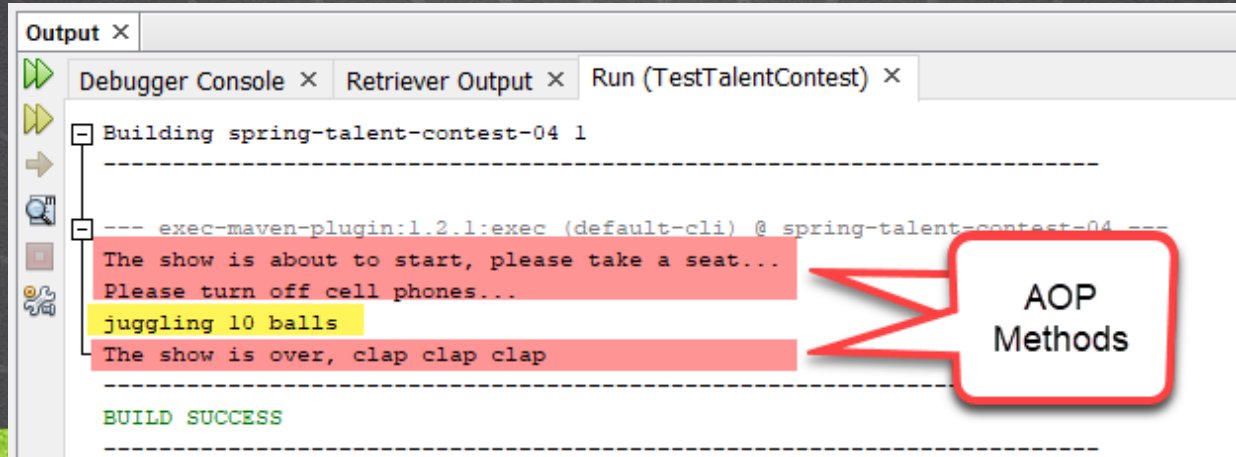


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

- The objective of the exercise is to add AOP functionality to our Talent Challenge project.
- At the end we will see the following output, the result of applying the concepts of AOP:



Output ×

Debugger Console × Retriever Output × Run (TestTalentContest) ×

```
Building spring-talent-contest-04 1
-----
--- exec-maven-plugin:1.2.1:exec (default-cli) @ spring-talent-contest-04 ---
The show is about to start, please take a seat...
Please turn off cell phones...
juggling 10 balls
The show is over, clap clap clap
-----
BUILD SUCCESS
-----
```

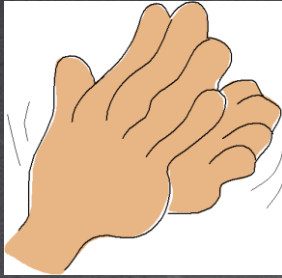
AOP Methods

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

PUBLIC

- Our talent show would be nothing without the great audience that visits this unprecedented show night after night.

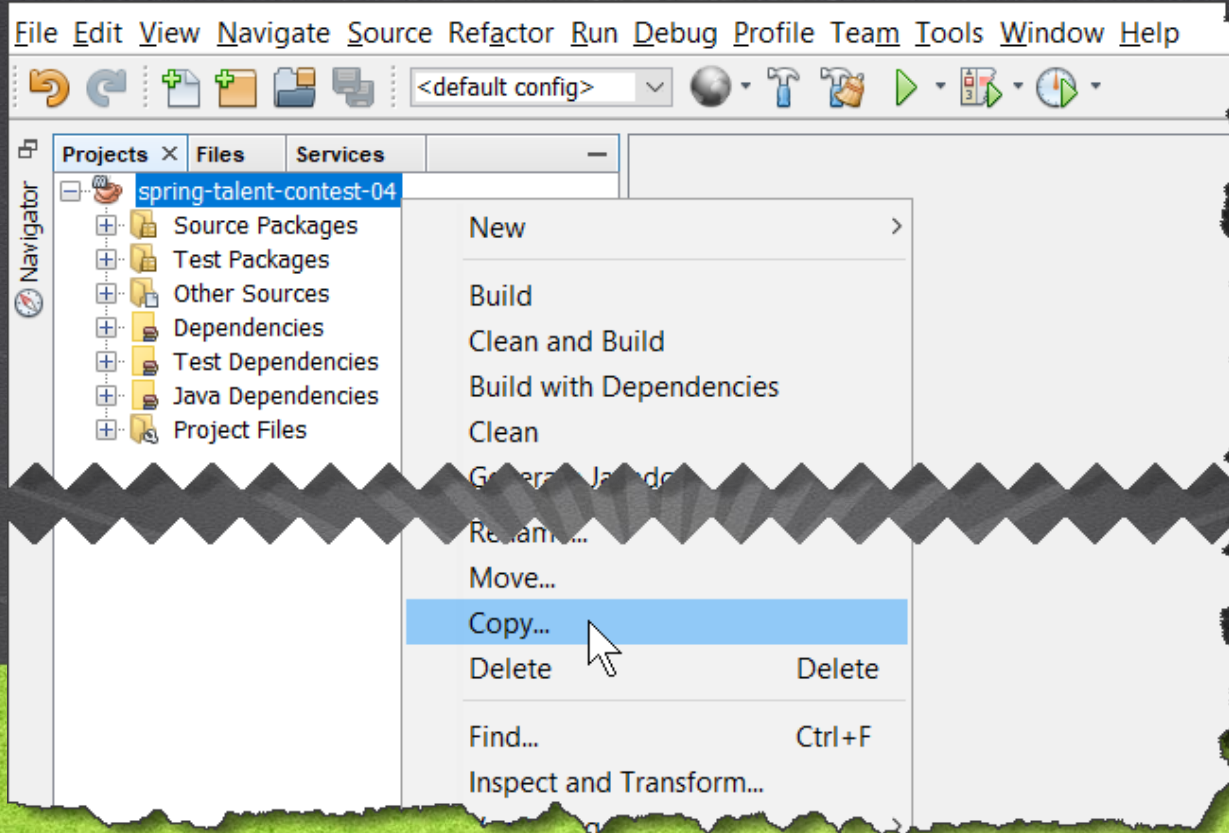


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

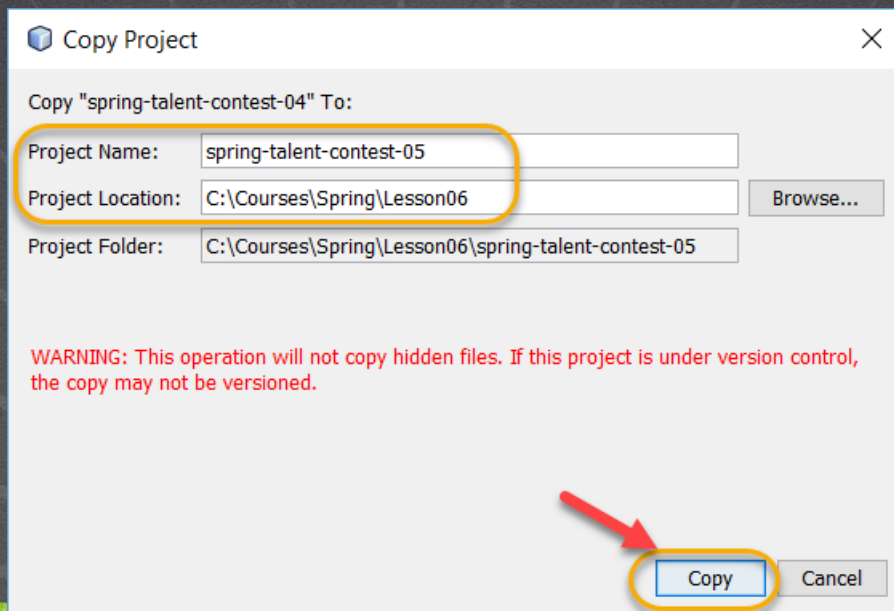
1. COPY THE PROJECT

We copy the project spring-concurso-talents-04:



1. COPY THE PROJECT

We changed the name of the project to spring-talent-contest-05 :

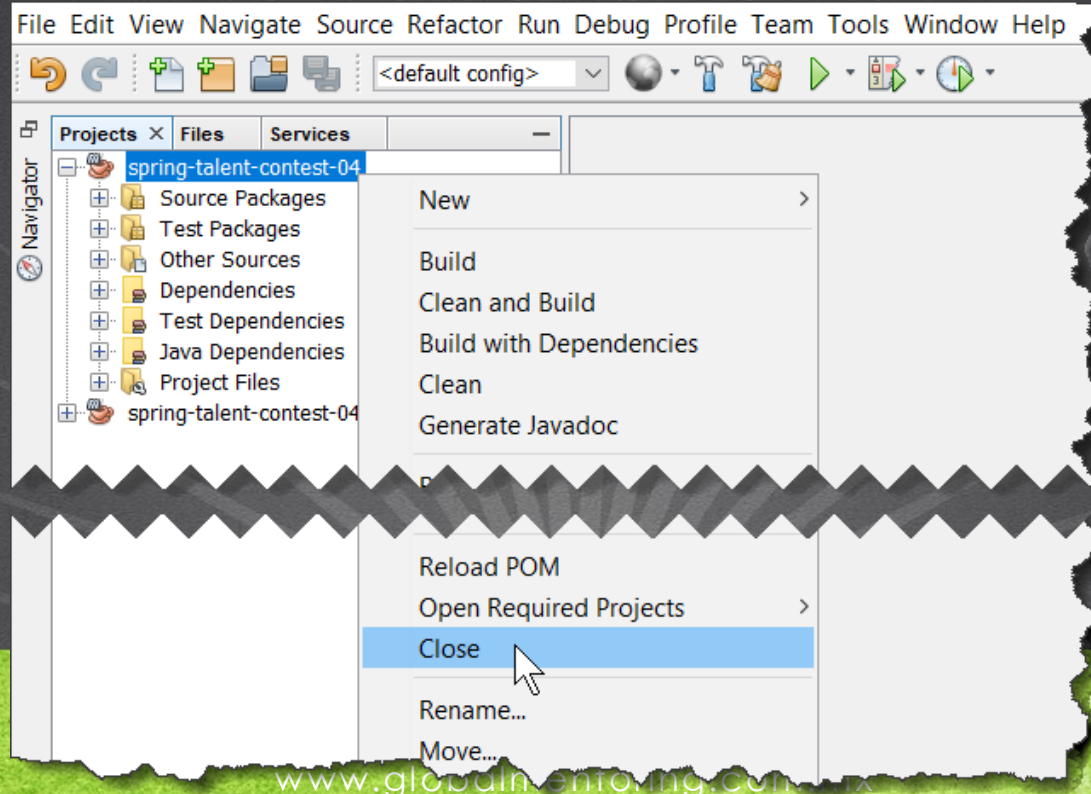


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

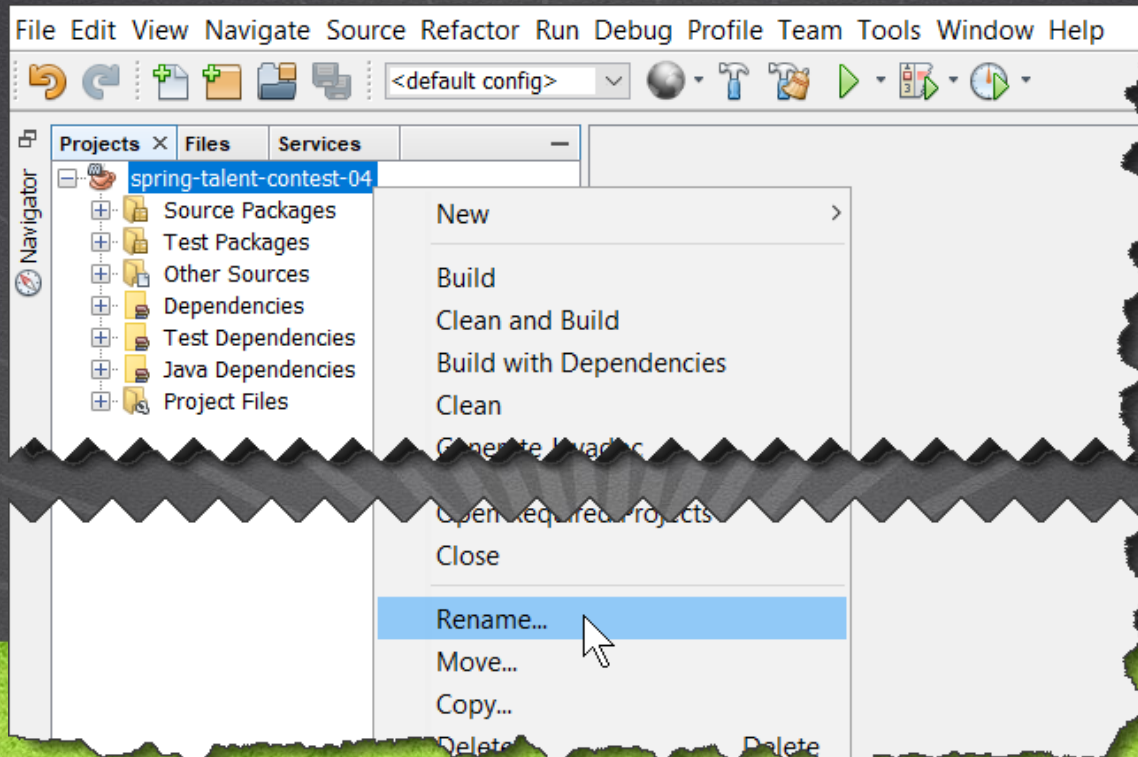
2. CLOSE THE PROJECT

We closed the previous project and we are left with the new one:



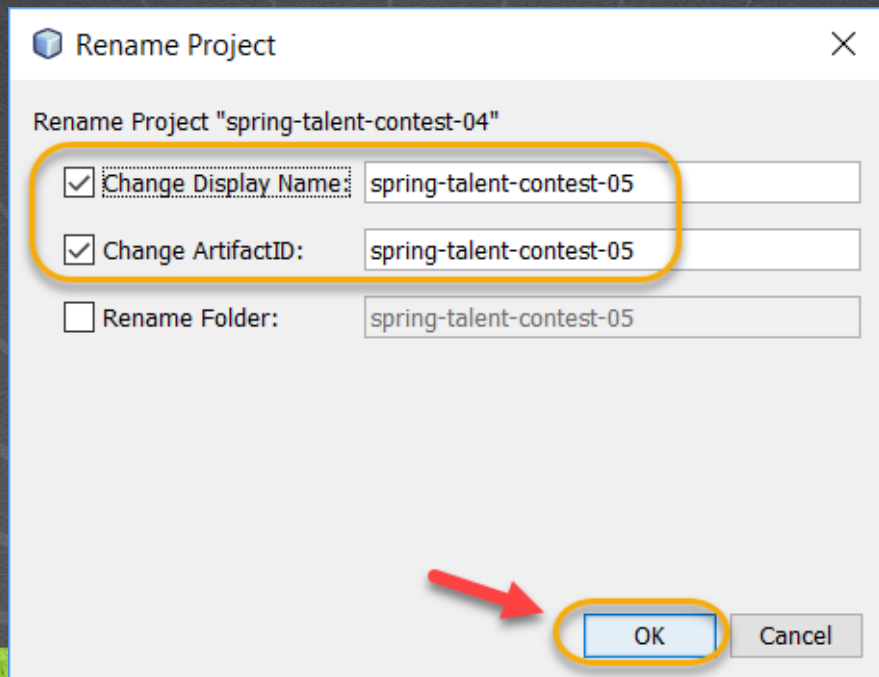
3. RENAME THE PROJECT

We renamed the project to spring-talent-contest-05:



3. RENAME THE PROJECT

We renamed the project to spring-talent-contest-05:



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

4. MODIFICATION OF THE APPLICATIONCONTEXT.XML FILE

Modify the applicationContext.xml file to add the AOP namespace:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<beans xmlns = "http://www.springframework.org/schema/beans"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation =
        "http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop.xsd">
```

4. MODIFICATION OF THE APPLICATIONCONTEXT.XML FILE

Add the configuration of audience behavior before and after finishing the execution of a talent.

```
<!-- AOP configuration-->

<aop:config>
  <aop:aspect ref="audience">
    <aop:pointcut expression="execution(* competitors.Competitor.execute(..))" id="show" />
    <aop:before pointcut-ref="show" method="sit" />
    <aop:before pointcut-ref="show" method="turnOffCells"/>
    <aop:after-returning pointcut-ref="show" method="clap" />
    <aop:after-throwing pointcut-ref="show" method="refund"/>
  </aop:aspect>
</aop:config>
```


5. MODIFY THE CODE

applicationContext.xml:

Click to download

```
<?xml version = "1.0" encoding = "UTF-8"?>
<beans xmlns = "http://www.springframework.org/schema/beans"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xmlns:context="http://www.springframework.org/schema/context"
      xmlns:aop="http://www.springframework.org/schema/aop"
      xsi:schemaLocation =
        "http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop.xsd">

    <context:component-scan base-package="competitors" />

    <!-- AOP configuration-->
    <aop:config>
        <aop:aspect ref="audience">
            <aop:pointcut expression="execution(* competitors.Competitor.execute(..))" id="show" />
            <aop:before pointcut-ref="show" method="sit" />
            <aop:before pointcut-ref="show" method="turnOffCells"/>
            <aop:after-returning pointcut-ref="show" method="clap" />
            <aop:after-throwing pointcut-ref="show" method="refund"/>
        </aop:aspect>
    </aop:config>
```

5. MODIFY THE CODE

[applicationContext.xml:](#)

[Click to download](#)

```
<!-- change of instrument -->
<bean id="saxophonist" class="competitors.Musician">
  <property name="song" value="Equinox"/>
  <!-- inner bean -->
  <property name="instrument">
    <bean class="competitors.Saxophone"/>
  </property>
</bean>

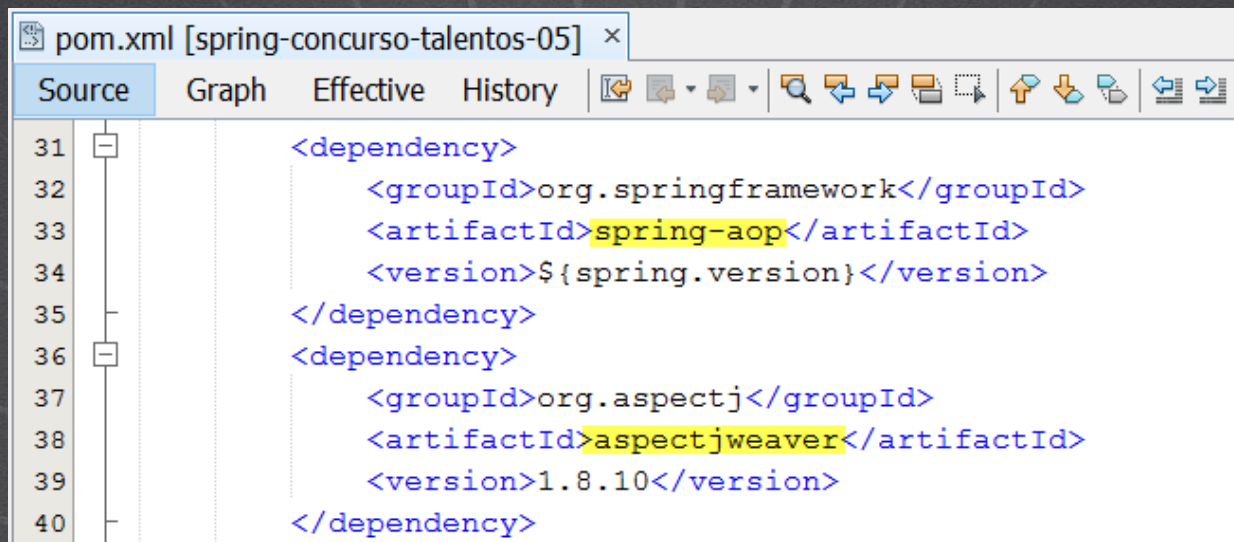
</beans>
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

6. MODIFICATION FILE POM.XML

Modify the pom.xml file to add the AOP libraries to the project:



The screenshot shows an IDE window titled 'pom.xml [spring-concurso-talentos-05]'. The 'Source' tab is active, displaying XML code. The code is structured with line numbers 31 to 40 on the left. It contains two dependency blocks. The first block (lines 31-35) is for 'spring-aop' with a version placeholder. The second block (lines 36-40) is for 'aspectjweaver' with version '1.8.10'. The artifact IDs 'spring-aop' and 'aspectjweaver' are highlighted in yellow.

```
31 <dependency>
32     <groupId>org.springframework</groupId>
33     <artifactId>spring-aop</artifactId>
34     <version>${spring.version}</version>
35 </dependency>
36 <dependency>
37     <groupId>org.aspectj</groupId>
38     <artifactId>aspectjweaver</artifactId>
39     <version>1.8.10</version>
40 </dependency>
```

7. MODIFY THE CODE

[pom.xml:](#)

Click to download

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>beans</groupId>
  <artifactId>spring-talent-contest-04</artifactId>
  <version>1</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <spring.version>5.1.0.RELEASE</spring.version>
    <log4j.version>2.11.1</log4j.version>
    <junit.version>5.3.1</junit.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>${spring.version}</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
    </dependency>
  </dependencies>
</project>
```


7. MODIFY THE CODE

[pom.xml:](#)

Click to download

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>${log4j.version}</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>${log4j.version}</version>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${spring.version}</version>
  <scope>test</scope>
  <type>jar</type>
</dependency>
```

7. MODIFY THE CODE

[pom.xml:](#)

Click to download

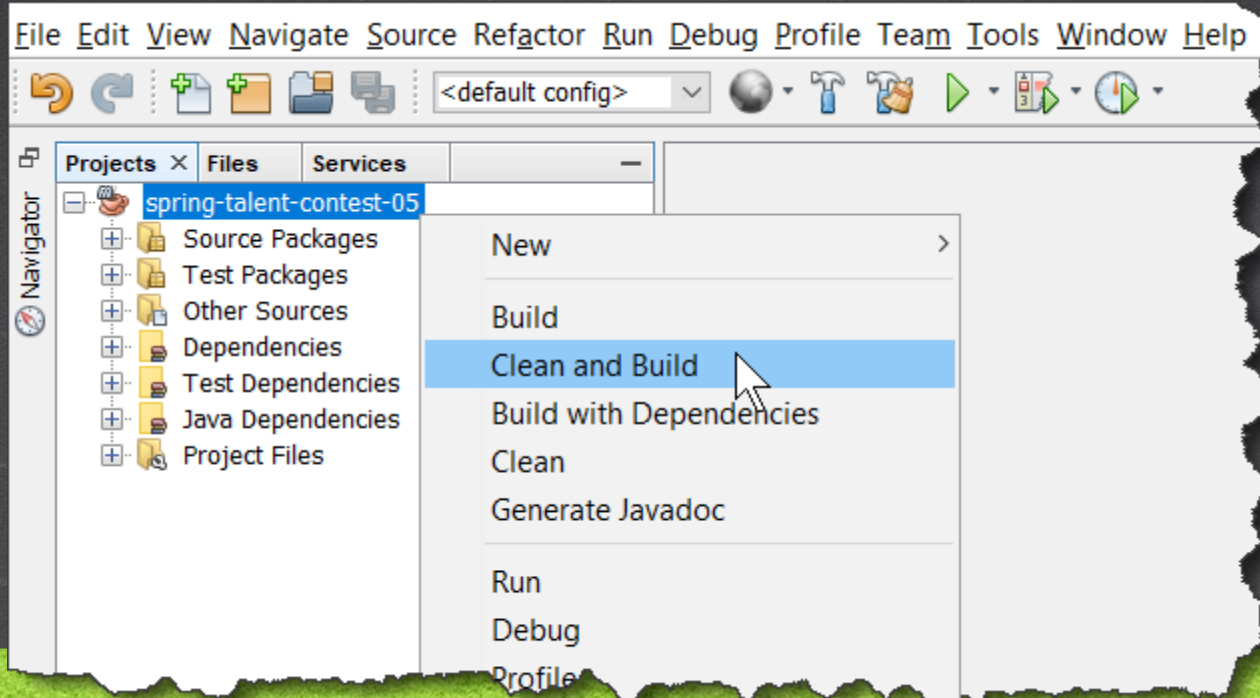
```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aop</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjweaver</artifactId>
  <version>1.9.1</version>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.22.0</version>
    </plugin>
  </plugins>
</build>
<name>spring-talent-contest-05</name>
</project>
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

8. EXECUTE CLEAN & BUILD

We do clean & build to update the pending libraries:

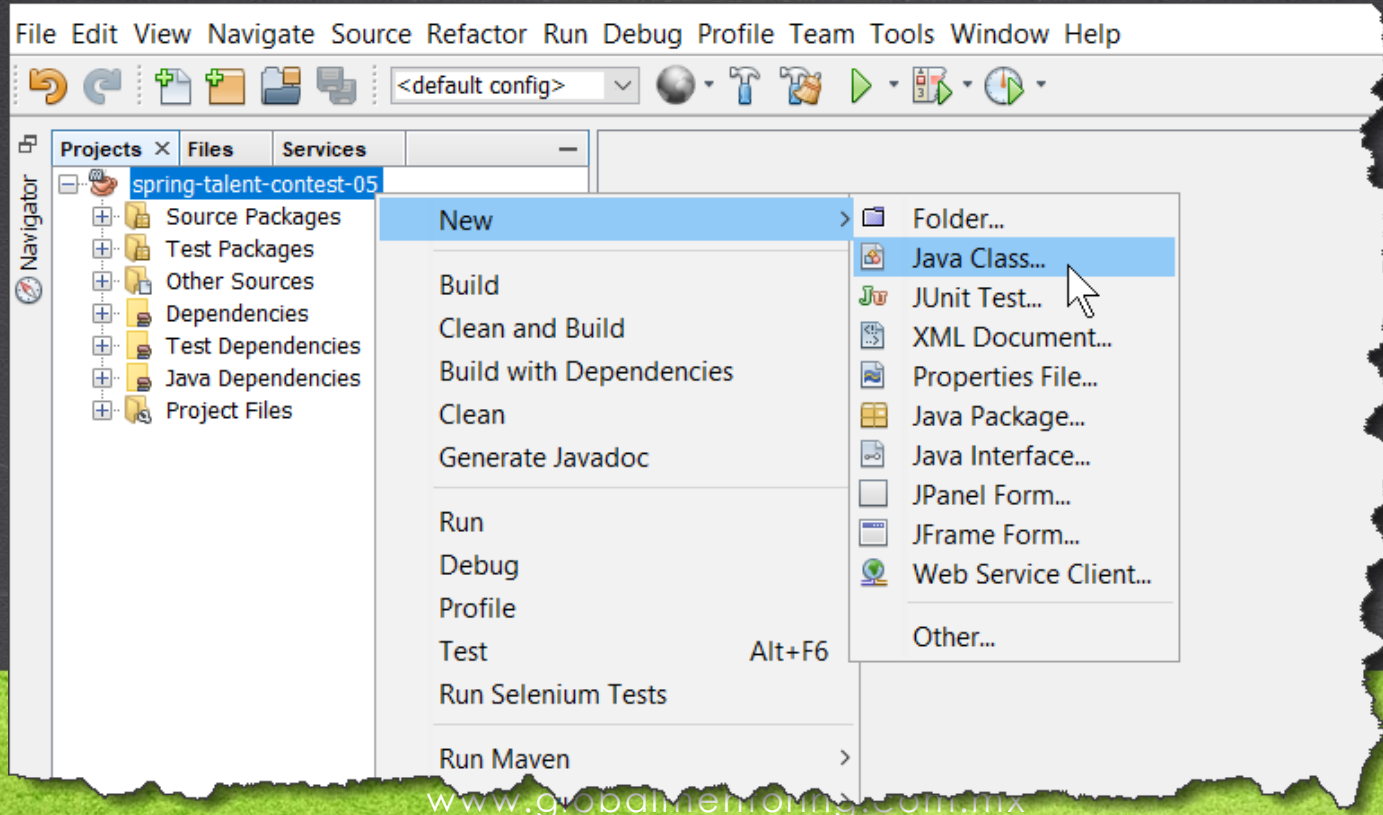


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

9. CREATE A NEW CLASS

Add the Audience.java class:



9. CREATE A NEW CLASS

Add the Audience.java class:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

10. MODIFY THE FILE

Audience.java:

[Click to download](#)

```
package competitors;

import org.springframework.stereotype.Component;

@Component
public class Audience {

    public void sit() {
        System.out.println("The show is about to start, please take a seat...");
    }

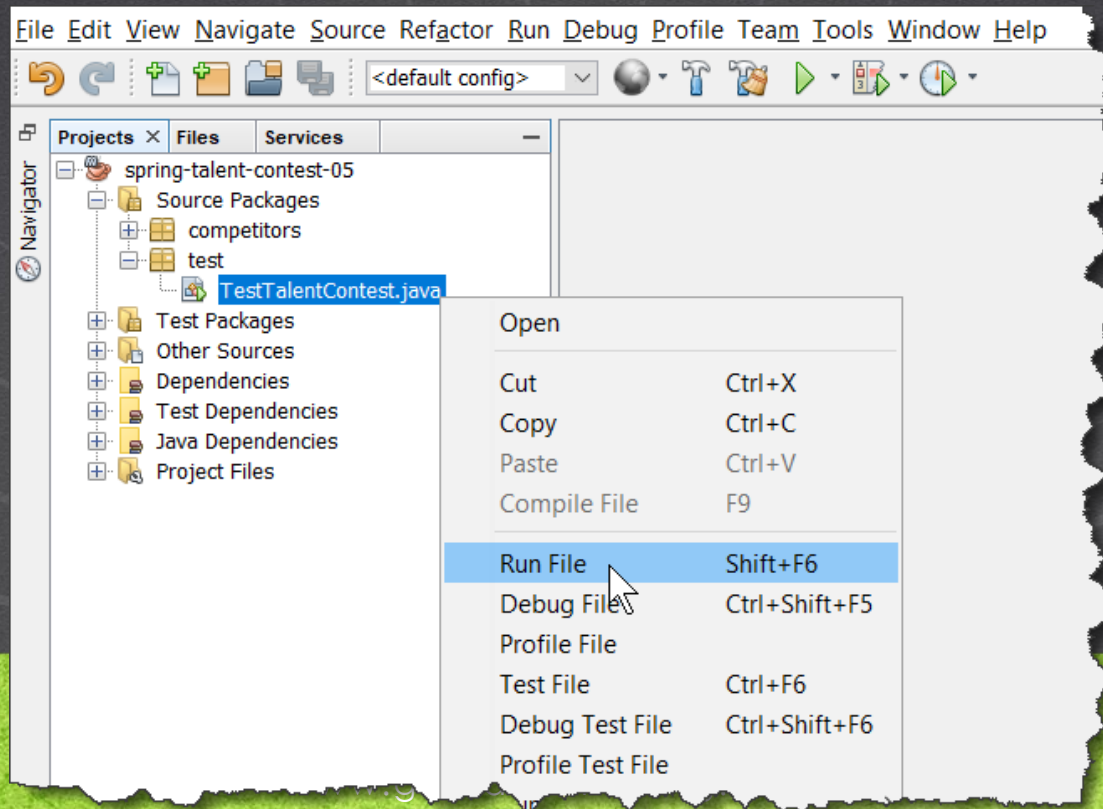
    public void turnOffCells() {
        System.out.println("Please turn off cell phones...");
    }

    public void clap() {
        System.out.println("The show is over, clap clap clap");
    }

    public void refund() {
        System.out.println("The show was terrible, tickets will be returned");
    }
}
```

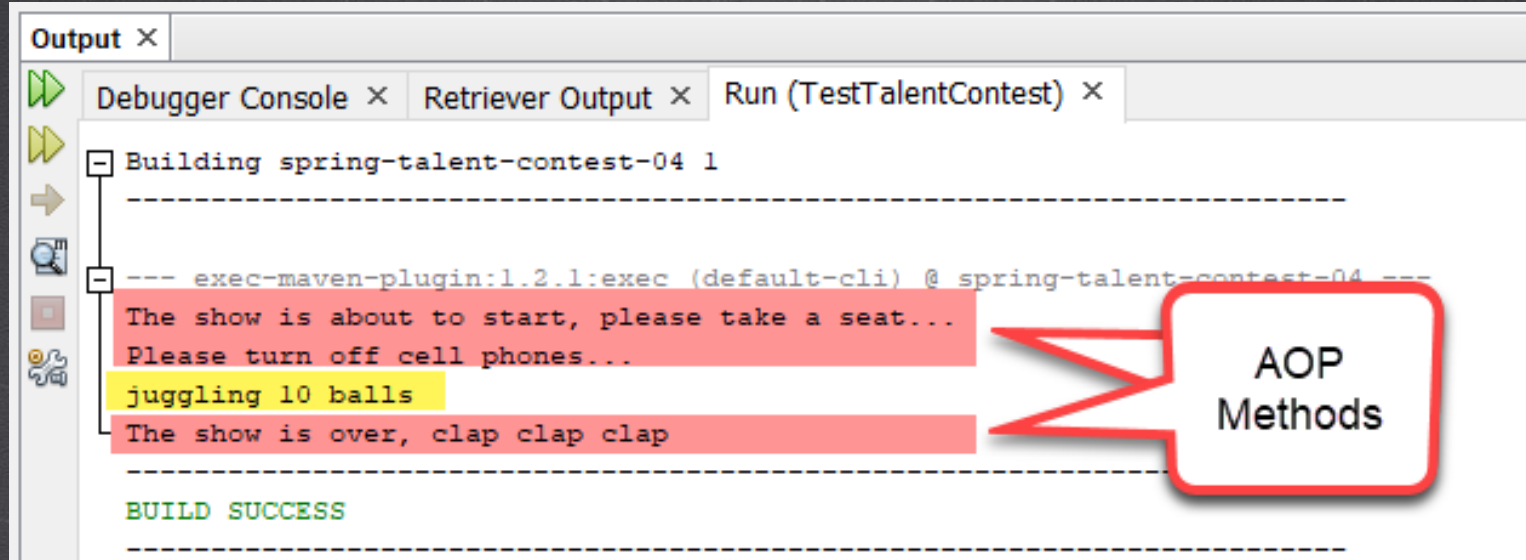
11. EXECUTE THE PROJECT

Execute the TestTalentContest.java class:



11. EXECUTE THE PROJECT

Execute the TestTalentContest.java class:



```
Output x
Debugger Console x  Retriever Output x  Run (TestTalentContest) x
Building spring-talent-contest-04 1
-----
--- exec-maven-plugin:1.2.1:exec (default-cli) @ spring-talent-contest-04 ---
The show is about to start, please take a seat...
Please turn off cell phones...
juggling 10 balls
The show is over, clap clap clap
-----
BUILD SUCCESS
-----
```

AOP Methods

EXERCISE CONCLUSION

With this exercise we have put into practice the basics of AOP concepts, in particular we modified the applicationContext.xml file which already recognizes the AOP concept, we also added the AOP and AspectJ libraries to our project, and finally we created the class that Add the AOP functionality to our project.

In the following exercises we will continue adding more AOP features to our project.



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

ONLINE COURSE

SPRING FRAMEWORK

By: Eng. Ubaldo Acosta



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx