

# SPRING FRAMEWORK COURSE

## EXERCISE

### TALENT CONTEST V7

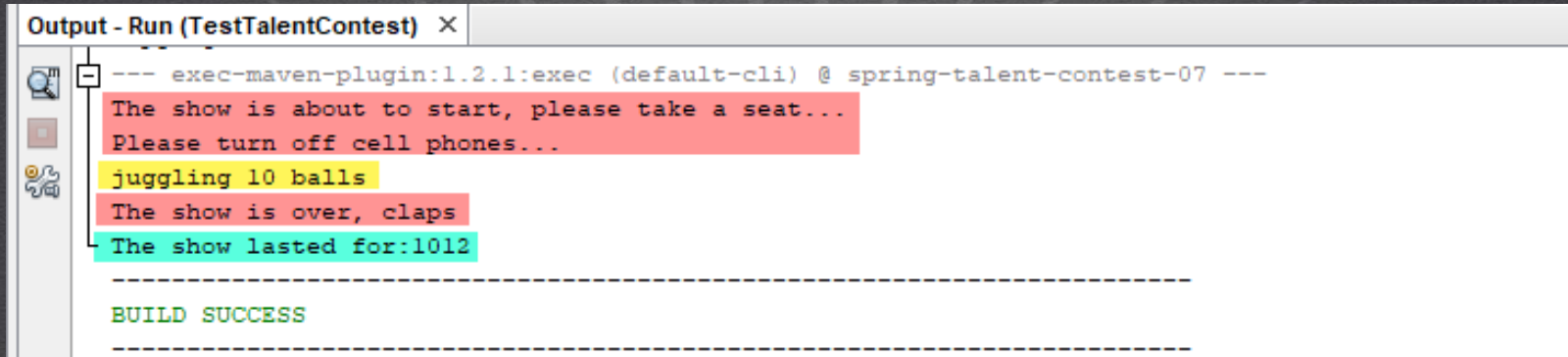


SPRING FRAMEWORK COURSE

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# EXERCISE OBJECTIVE

- The objective of the exercise is to add AOP functionality with the use of Annotations.
- At the end we must see the following output, the result of applying the described AOP concepts:

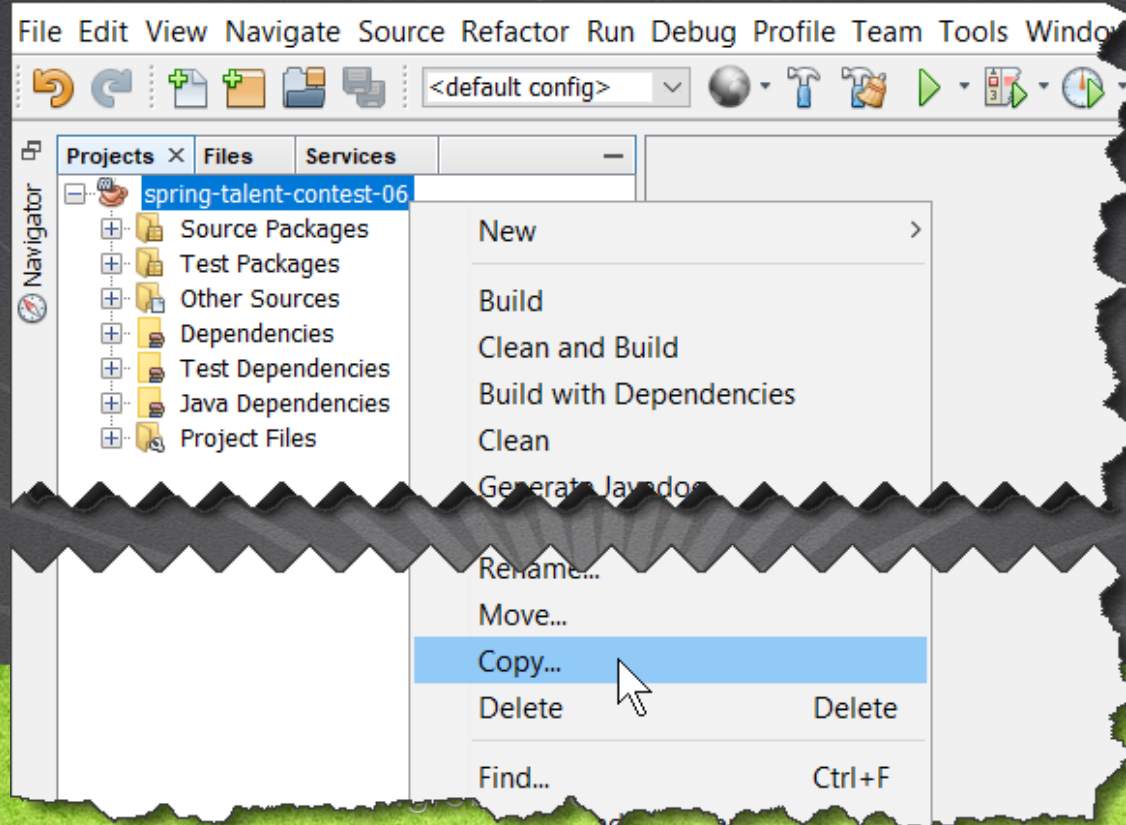


```
Output - Run (TestTalentContest) X
--- exec-maven-plugin:1.2.1:exec (default-cli) @ spring-talent-contest-07 ---
The show is about to start, please take a seat...
Please turn off cell phones...
juggling 10 balls
The show is over, claps
The show lasted for:1012
-----
BUILD SUCCESS
-----
```



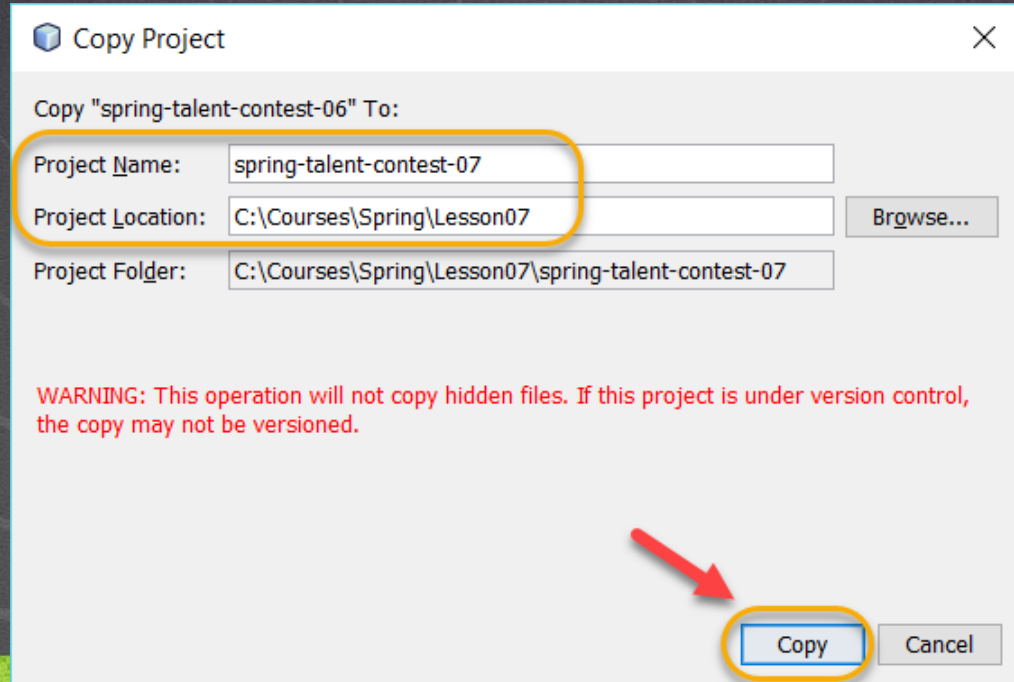
# 1. COPY THE PROJECT

Copy the spring-talent-contest-06 project:



# 1. COPY THE PROJECT

Change the Project name to spring-talent-contest-07:

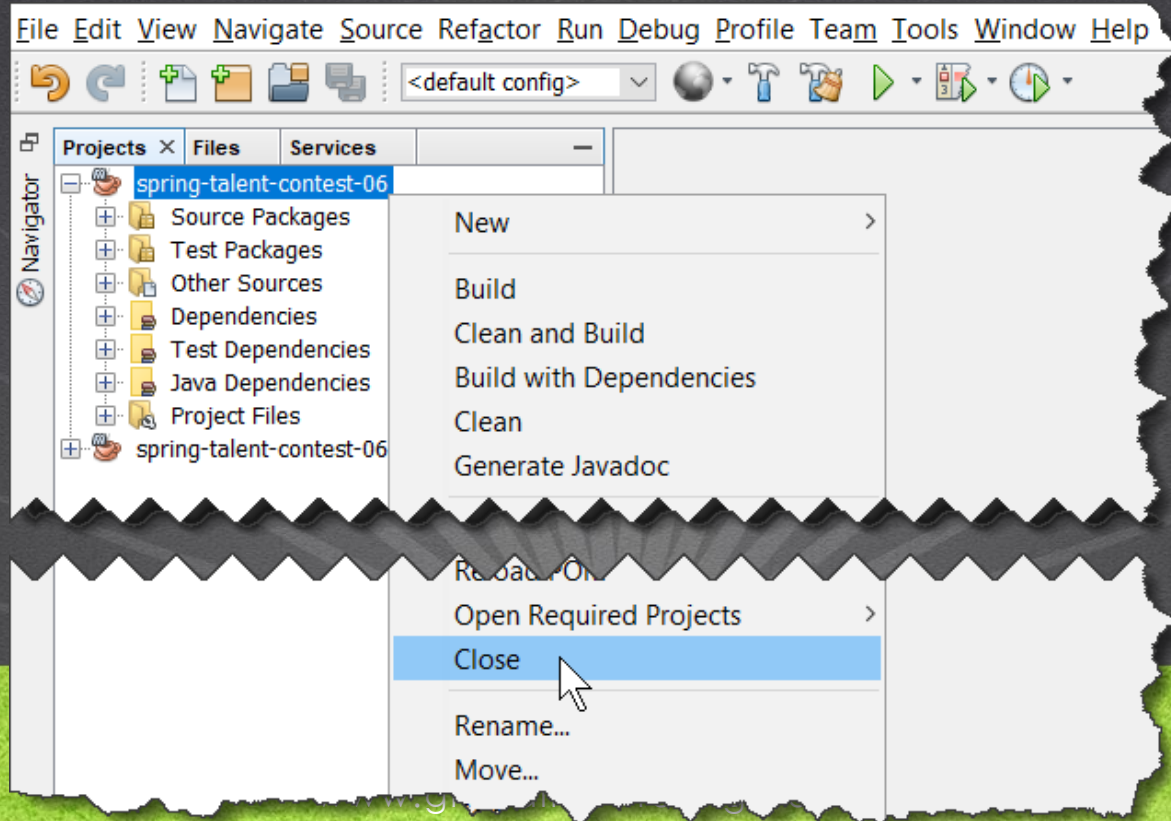


**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 2. CLOSE THE PREVIOUS PROJECT

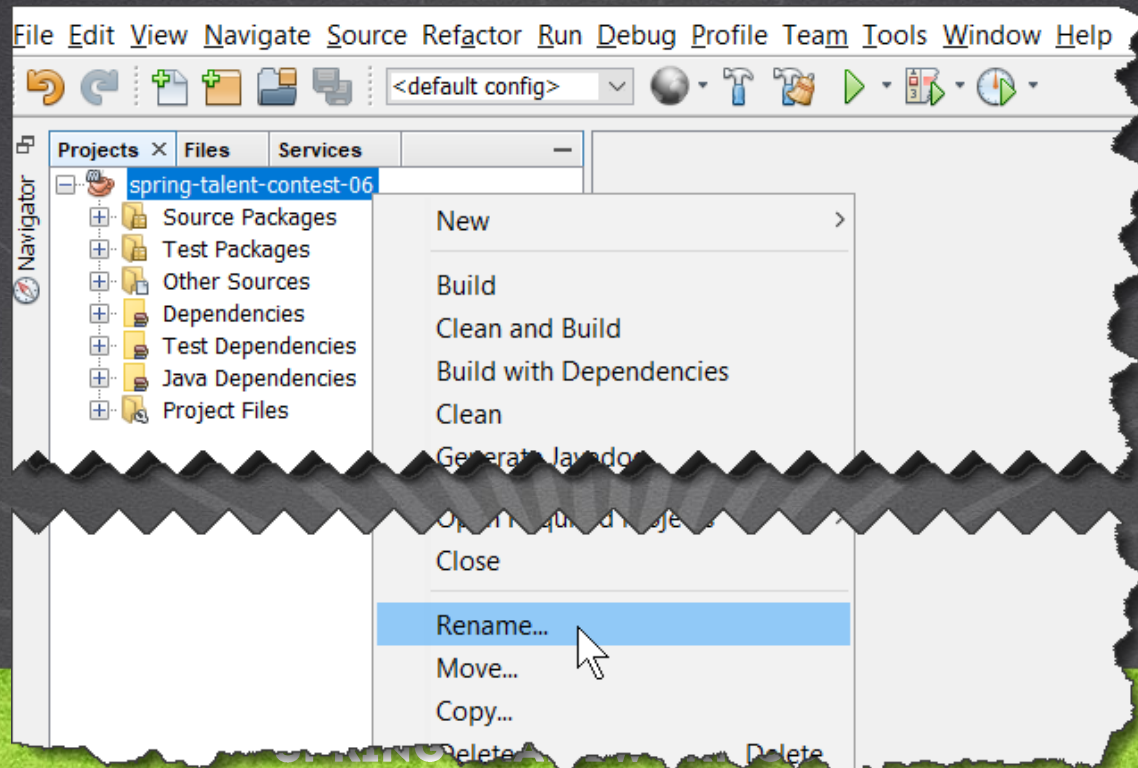
We closed the previous project and we are left with the new :





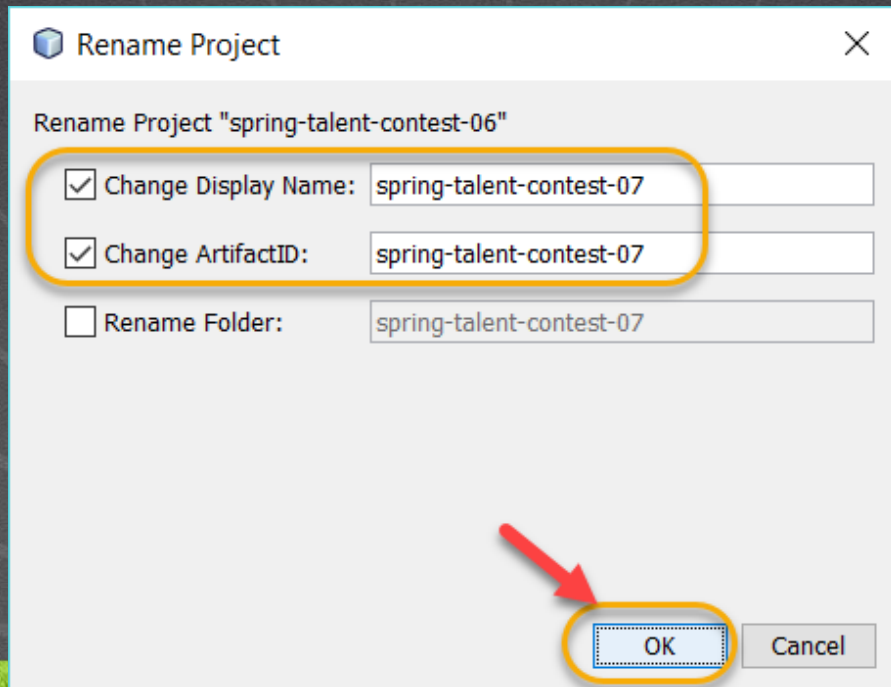
### 3. RENAME THE PROJECT

Rename the Project to spring-talent-contest-07:



### 3. RENAME THE PROJECT

Rename the Project to spring-talent-contest-07:



Rename Project

Rename Project "spring-talent-contest-06"

☒ Change Display Name: spring-talent-contest-07

☒ Change ArtifactID: spring-talent-contest-07

☐ Rename Folder: spring-talent-contest-07

OK Cancel

**SPRING FRAMEWORK COURSE**

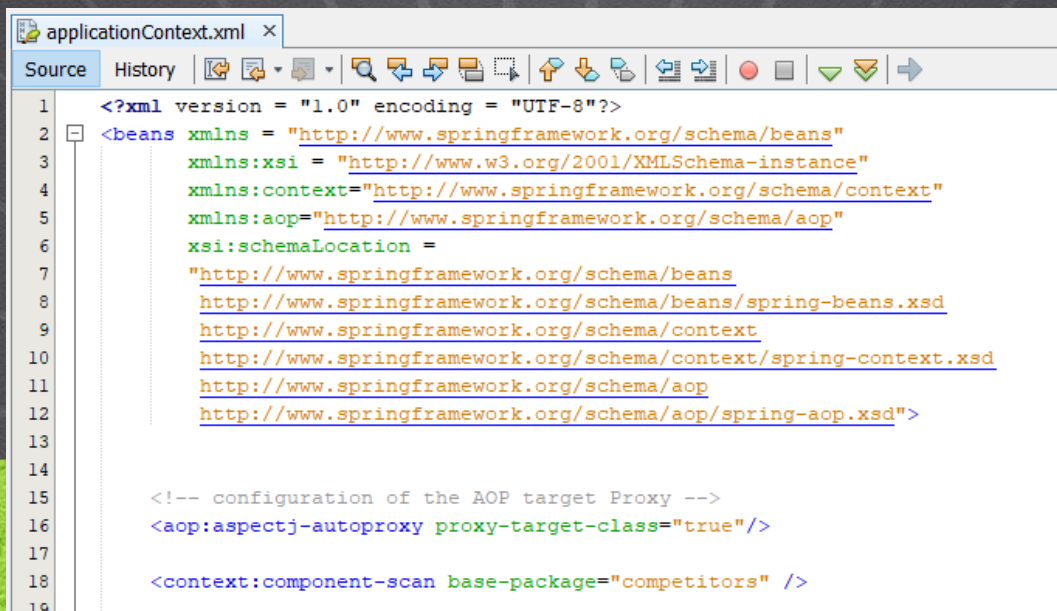
[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 4. MODIFICATION FILE APPLICATIONCONTEXT.XML

We must modify the definition of the Aspect in the applicationContext.xml file, we eliminate the configuration of the aspect and leave only the following label, which will also serve to recognize the use of annotations.

```
<aop:aspectj-autoproxy proxy-target-class="true"/>
```

The applicationContext.xml file is left as follows:



```
1 <?xml version = "1.0" encoding = "UTF-8"?>
2 <beans xmlns = "http://www.springframework.org/schema/beans"
3       xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:aop="http://www.springframework.org/schema/aop"
6       xsi:schemaLocation =
7       "http://www.springframework.org/schema/beans
8       http://www.springframework.org/schema/beans/spring-beans.xsd
9       http://www.springframework.org/schema/context
10      http://www.springframework.org/schema/context/spring-context.xsd
11      http://www.springframework.org/schema/aop
12      http://www.springframework.org/schema/aop/spring-aop.xsd">
13
14
15     <!-- configuration of the AOP target Proxy -->
16     <aop:aspectj-autoproxy proxy-target-class="true"/>
17
18     <context:component-scan base-package="competitors" />
19
```



# 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">

<!-- configuration of the AOP target Proxy -->
<aop:aspectj-autoproxy proxy-target-class="true"/>

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

<!-- 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>
```

## 6. MODIFY THE CLASS

Then we modify the Audience.java class. This class will now have the annotations that will allow us to avoid the configuration in the applicationContext.xml file. We will use annotations such as @Aspect, @Pointcut and @Around.

Let's see how our class is:



**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 6. MODIFY THE CODE

[Audience.java:](#)

[Click to download](#)

```
package competitors;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.*;
import org.springframework.stereotype.Component;

@Component
@Aspect
public class Audience {

    @Pointcut("execution(* competitors.Competitor.execute(..))")
    public void executeShow() {
        System.out.println("Executing show");
    }
}
```

**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



## 6. MODIFY THE CODE

[Audience.java:](#)

[Click to download](#)

```
@Around("executeShow()")
public void monitorShow(ProceedingJoinPoint joinpoint) {
    try {
        System.out.println("The show is about to start, please take a seat...");
        System.out.println("Please turn off cell phones...");
        //We record the start time
        long startTime = System.currentTimeMillis();

        //The business method is called (target method)
        joinpoint.proceed();

        Thread.sleep(1000); //1 second
        //This delay in milliseconds is optional and can be put into business methods
        //to simulate the duration of the method

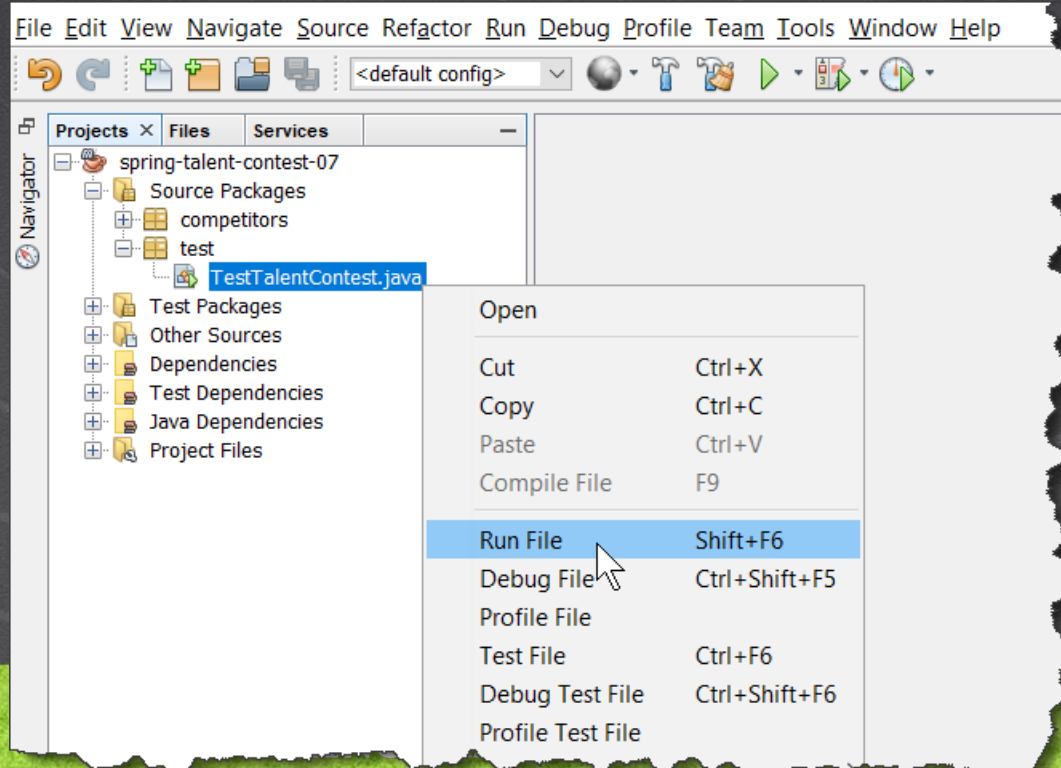
        long endTime = System.currentTimeMillis();
        System.out.println("The show is over, claps");
        System.out.println("The show lasted for:" + (endTime - startTime));
    } catch (Throwable t) {
        System.out.println("The show was terrible, the tickets will be refunded");
    }
}
```

**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

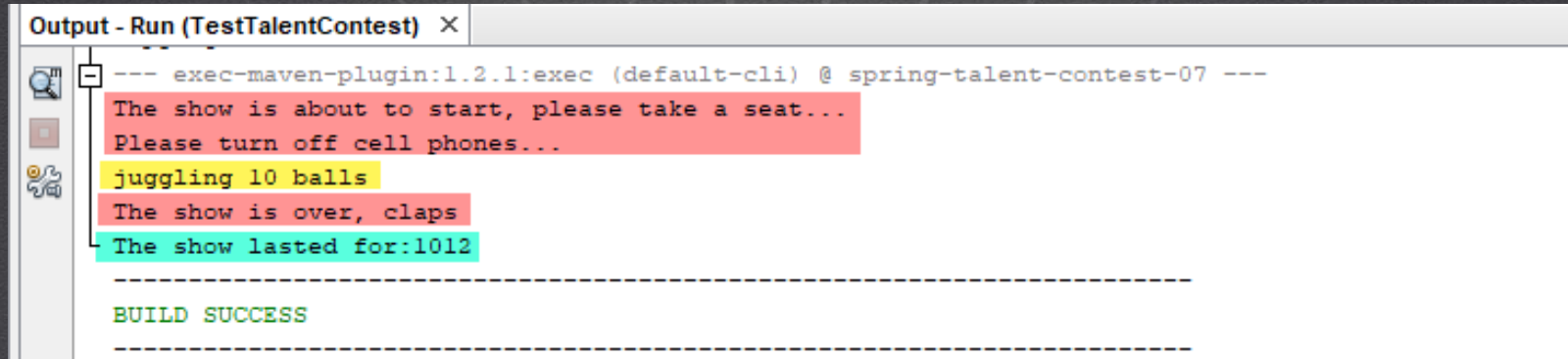
# 7. EXECUTE THE PROJECT

Execute the class TestTalentContest.java:



# 7. EXECUTE THE PROJECT

Execute the class TestTalentContest.java:



```
Output - Run (TestTalentContest) X
--- exec-maven-plugin:1.2.1:exec (default-cli) @ spring-talent-contest-07 ---
The show is about to start, please take a seat...
Please turn off cell phones...
juggling 10 balls
The show is over, claps
The show lasted for:1012
-----
BUILD SUCCESS
-----
```



# EXERCISE CONCLUSION

With this exercise we have configured our AOP exercise, now with the use of annotations such as @Aspect, @Pointcut and @Around.

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



Experiencia y Conocimiento para tu vida

**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

**ONLINE COURSE**

# **SPRING FRAMEWORK**

By: Eng. Ubaldo Acosta



**SPRING FRAMEWORK COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)