**Important note: Before going further you are requested to go through theory of Spring Dependency Injection with the help of xml based autowiring mentioned in JavaLive Notes on Spring Core thoroughly so that you can understand the concept very well. (\*\*This activity i.e. going through theory part of respective subject is applicable for each of our project.)**

**What this project does??**

This project explains DI with the help of auto wiring in spring framework of following types

i) byName

ii)byType

iii) ConstructorBased

**Note that this is xml based spring project having configuration file viz. ' spring-config.xml' in resources folder.**

**Steps to create project:-**

1. Create quick start maven project. Refer file 'CreateQuickStartMavenProjectInEclipse.docx' in this project for more details.

2. Add require dependencies for spring as shown in pom.xml file.

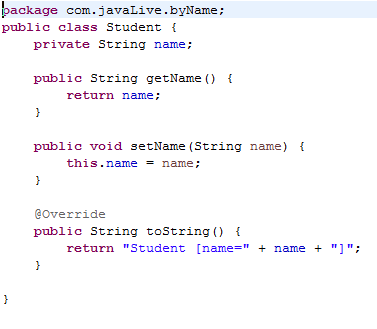
3. Create required packages and add the files.

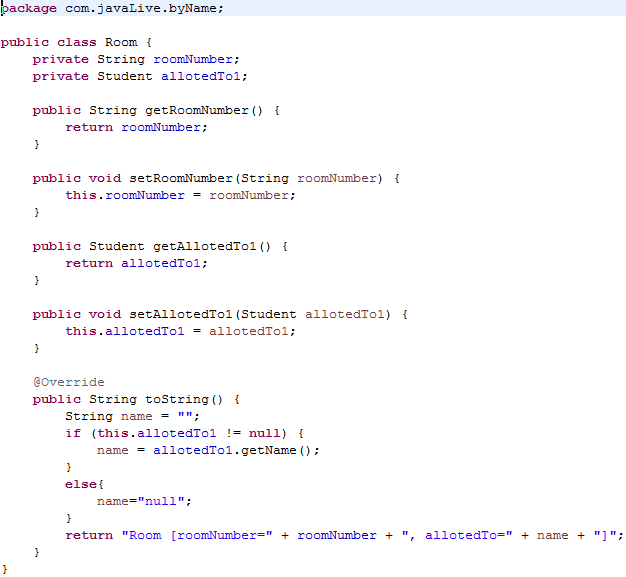
**Functioning of the project**

As mentioned earlier, this project deals with explaining various dependency injection types in spring. In this project please carefully go through packages com.javaLive.byName, com.javaLive.byType, com.javaLive.constructorBased.

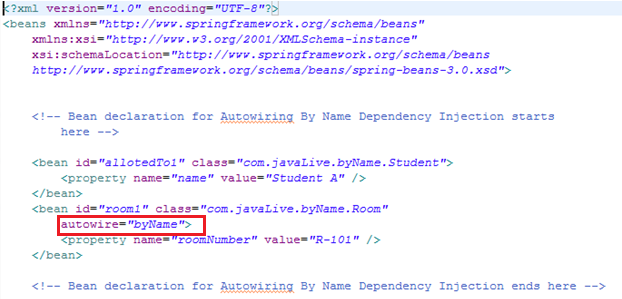
Each of these package contains bean files along with main class to explain each type.

We will study byName package for understanding. Rest of the two have same functionality.

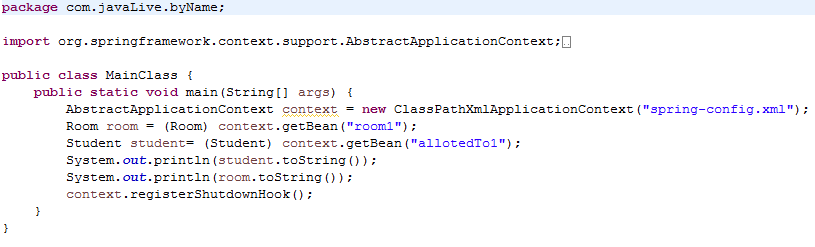




To understand the relation and functioning carefully go through spring configuration file viz 'spring-config.xml' in resources folder. **Here we do not used 'ref' unlike in project 'SpringCoreDependencyInjectionDemo' instead we have used autowire attribute in bean tag which implicitly refer the mentioned bean behind the scene.**



Now class com.javaLive.byName.MainClass with its output is as follow: Taking above explanation in mind its quite straightforward.





Now please go through packages com.javaLive.byType and com.javaLive.constructorBased on the basis of above explanation with respective changes.