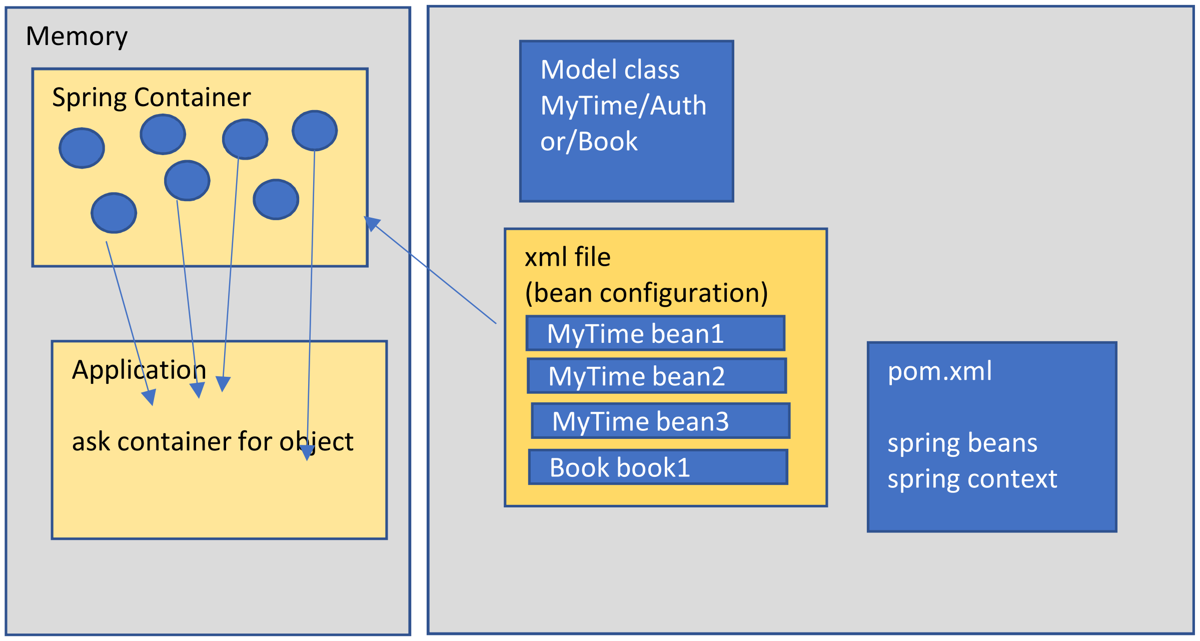
Bean
It is an object created in spring container
Ready to use object
By injecting all dependencies

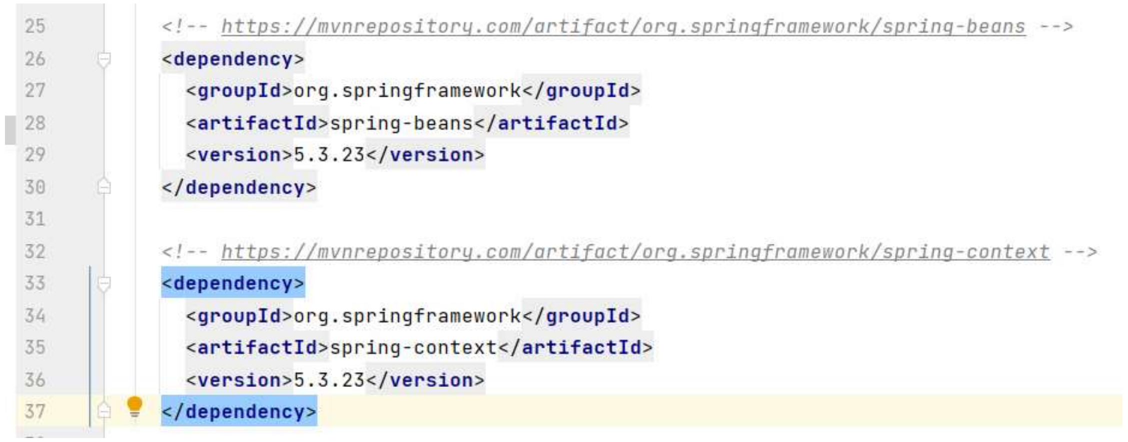
Container
Area in RAM to load all beans defined in configuration
Beans which are loaded into container can be used by application directly
Beans load to container from configuration before using



Steps to create spring application, define beans by injecting values thru setters

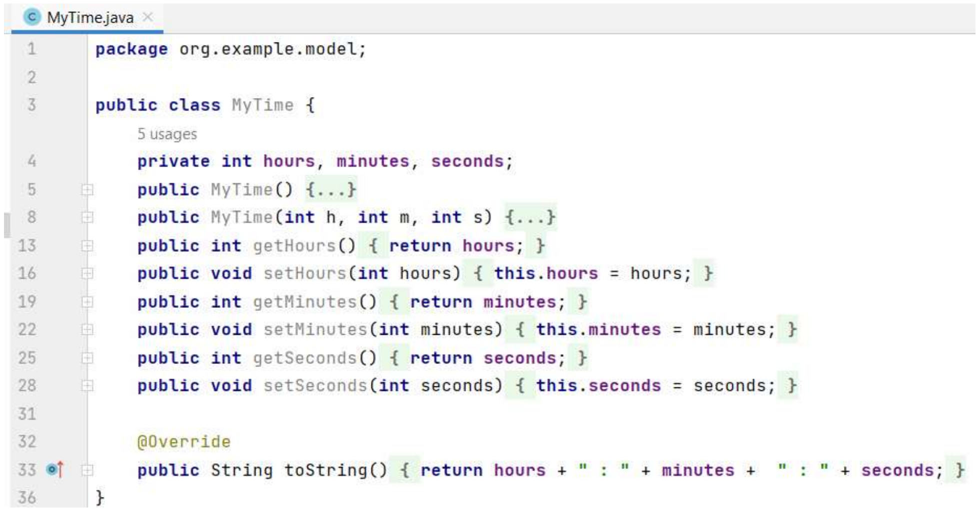
1 Create maven project

2 add required dependencies



```xml
<!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>5.3.23</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>5.3.23</version>
</dependency>
```

3 Define model class (MyTime)

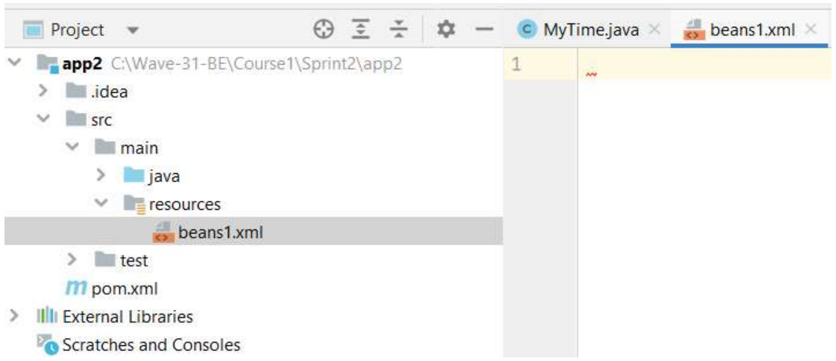


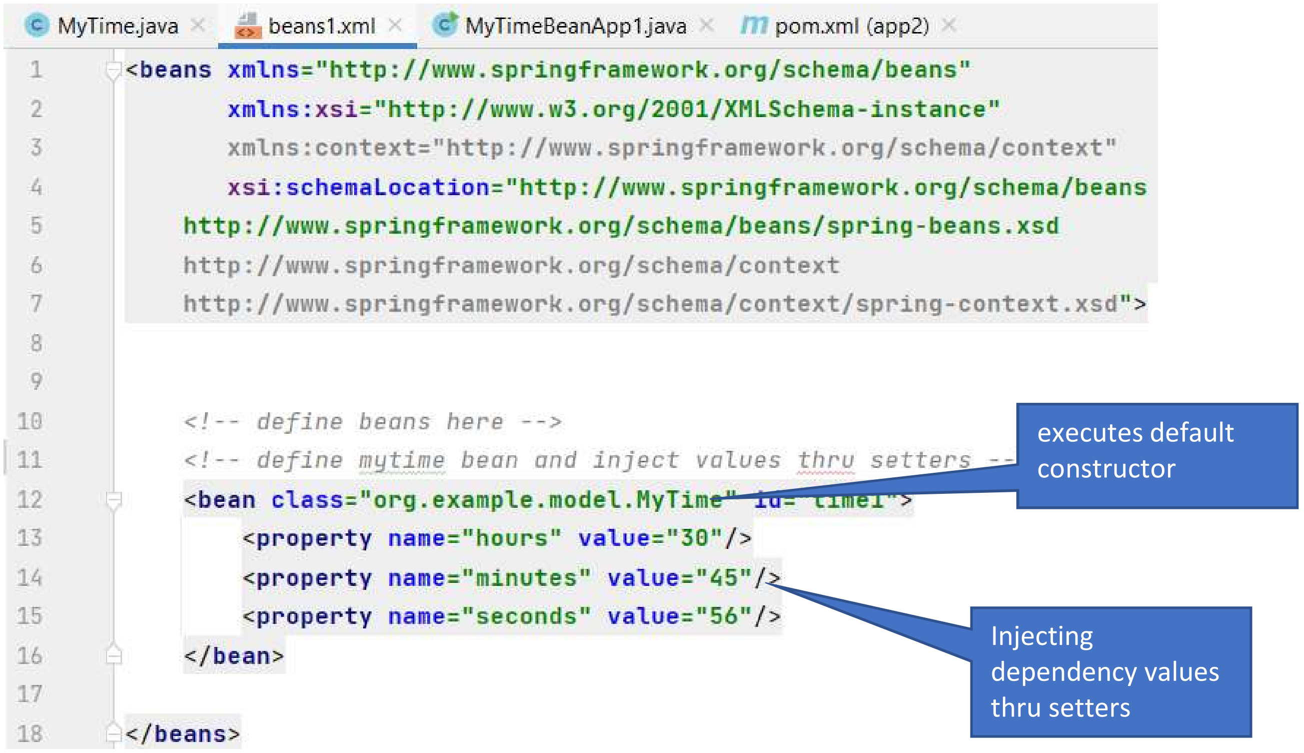
```java
package org.example.model;

public class MyTime {
    5 usages
    private int hours, minutes, seconds;
    public MyTime() {...}
    public MyTime(int h, int m, int s) {...}
    public int getHours() { return hours; }
    public void setHours(int hours) { this.hours = hours; }
    public int getMinutes() { return minutes; }
    public void setMinutes(int minutes) { this.minutes = minutes; }
    public int getSeconds() { return seconds; }
    public void setSeconds(int seconds) { this.seconds = seconds; }

    @Override
    public String toString() { return hours + " : " + minutes +  " : " + seconds; }
}
```

4 Define beans in xml configuration file

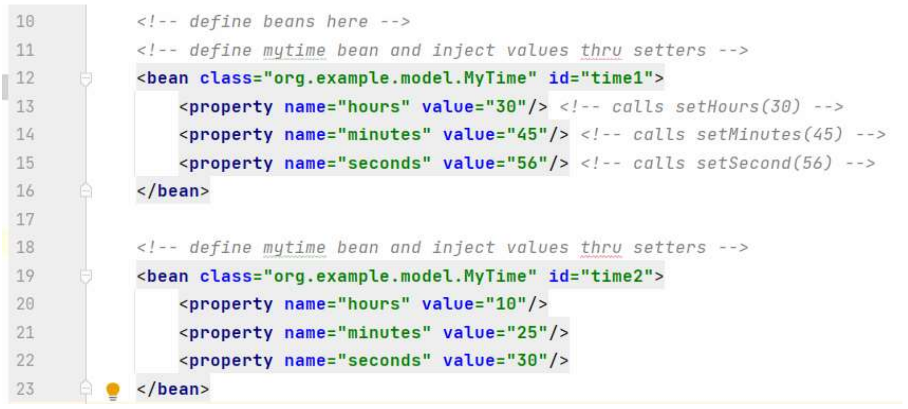
create resources folder under main

create beans xml file under resources folder



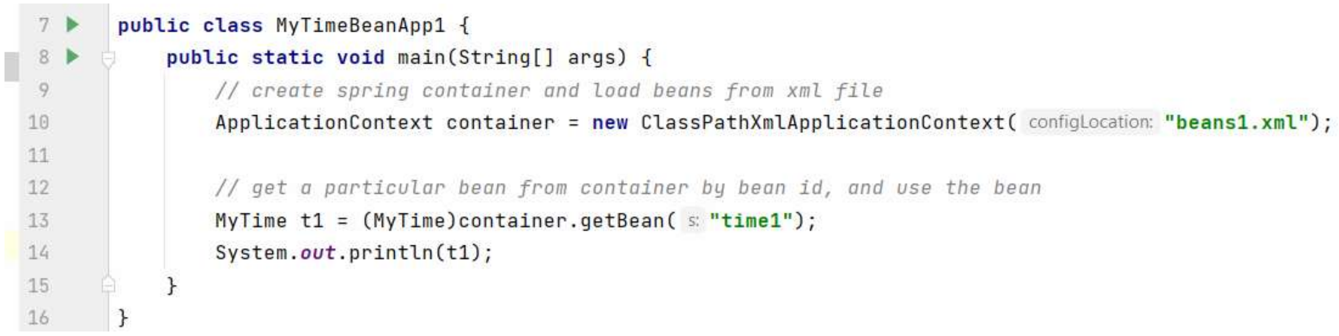

MyTime.java
beans1.xml
MyTimeBeanApp1.java
pom.xml (app2)
executes default constructor
Injecting dependency values thru setters

```xml
<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"
       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">


    <!-- define beans here -->
    <!-- define mytime bean and inject values thru setters --
    <bean class="org.example.model.MyTime" id="time1">
        <property name="hours" value="30"/>
        <property name="minutes" value="45"/>
        <property name="seconds" value="56"/>
    </bean>

</beans>
```

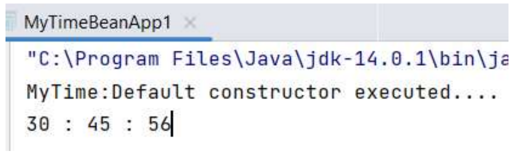


```xml
<!-- define beans here -->
<!-- define mytime bean and inject values thru setters -->
<bean class="org.example.model.MyTime" id="time1">
    <property name="hours" value="30"/> <!-- calls setHours(30) -->
    <property name="minutes" value="45"/> <!-- calls setMinutes(45) -->
    <property name="seconds" value="56"/> <!-- calls setSecond(56) -->
</bean>

<!-- define mytime bean and inject values thru setters -->
<bean class="org.example.model.MyTime" id="time2">
    <property name="hours" value="10"/>
    <property name="minutes" value="25"/>
    <property name="seconds" value="30"/>
</bean>
```

5 in app, create spring container and load beans from configuration file



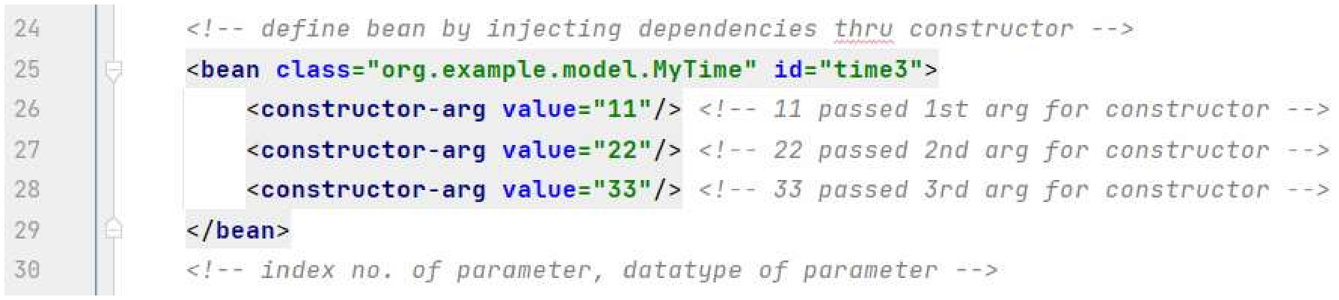
```java
public class MyTimeBeanApp1 {
    public static void main(String[] args) {
        // create spring container and load beans from xml file
        ApplicationContext container = new ClassPathXmlApplicationContext( configLocation: "beans1.xml");

        // get a particular bean from container by bean id, and use the bean
        MyTime t1 = (MyTime)container.getBean( s: "time1");
        System.out.println(t1);
    }
}
```



MyTimeBeanApp1

```
"C:\Program Files\Java\jdk-14.0.1\bin\ja
MyTime:Default constructor executed....
30 : 45 : 56
```

Injecting dependencies thru constructor


```xml
<!-- define bean by injecting dependencies thru constructor -->
<bean class="org.example.model.MyTime" id="time3">
    <constructor-arg value="11"/> <!-- 11 passed 1st arg for constructor -->
    <constructor-arg value="22"/> <!-- 22 passed 2nd arg for constructor -->
    <constructor-arg value="33"/> <!-- 33 passed 3rd arg for constructor -->
</bean>
<!-- index no. of parameter, datatype of parameter -->
```

Example 2 with ref type dependencies

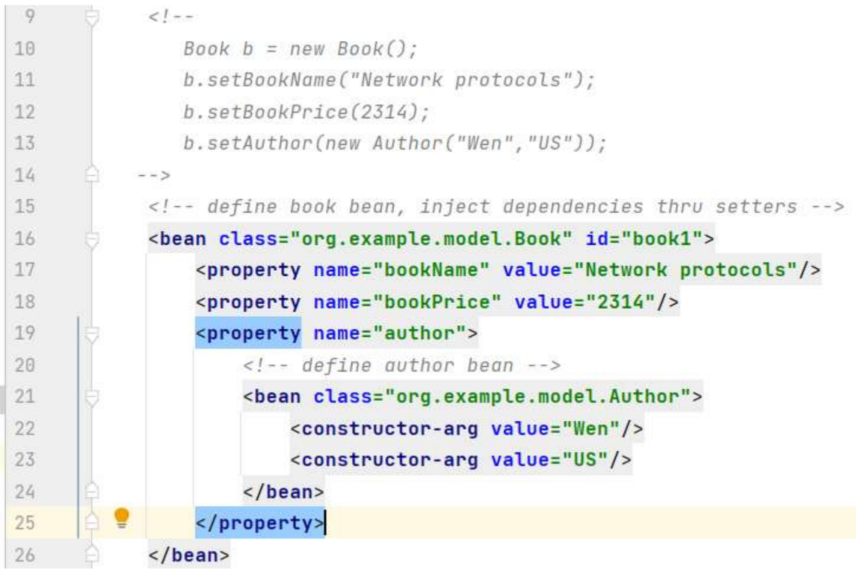
Author class to be defined first

Book class to be defined next

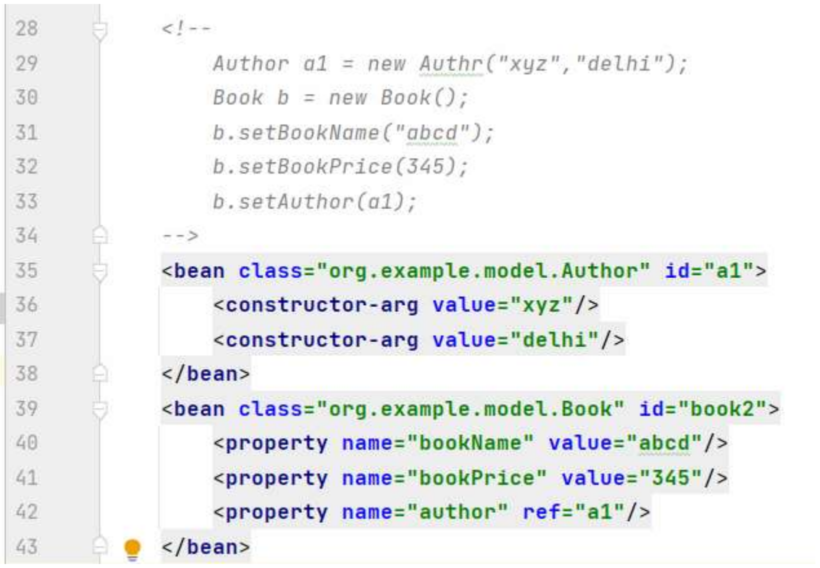
Book HAS_A Author

```
class Book { // dependent
private string bookname;
private int price
private Author author;
public Book() {  } // not assigned value
public Book( String bn, int price, Author auth) {  }
Setters
public String toString() {  ... }
```

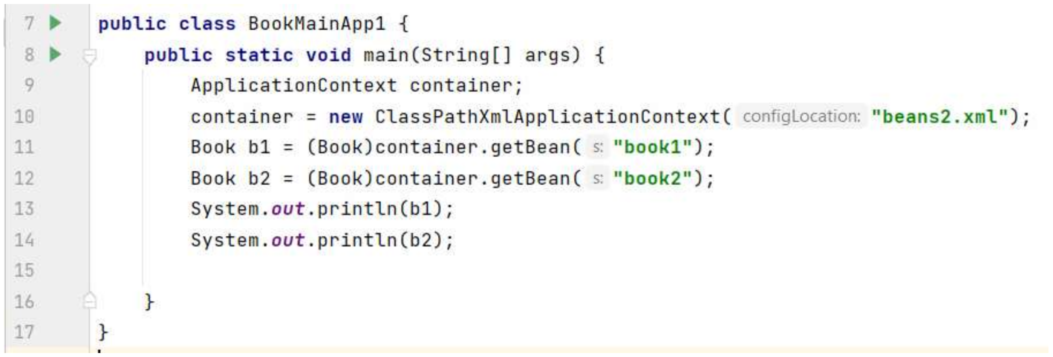
```
class Author{ // dependency
private string name,address;
public Author() {  } // not assigned value
public Author(String nm, String add) {  }
Setters
public String toString() {  ... }
```



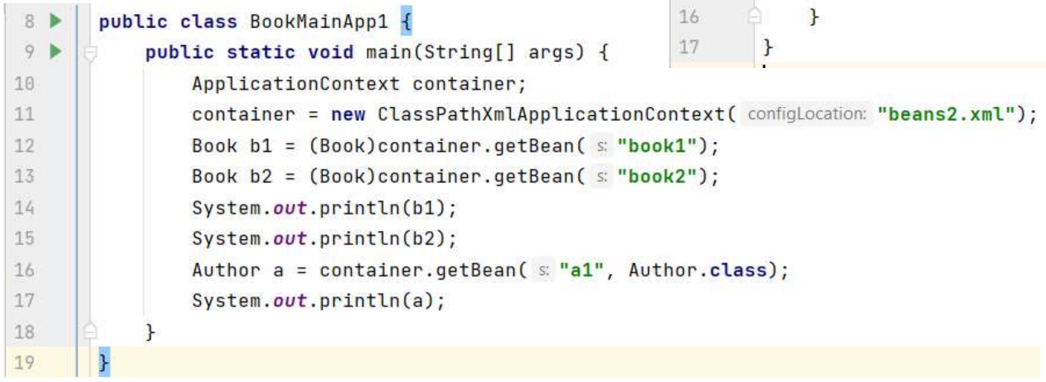
```xml
<!--
    Book b = new Book();
    b.setBookName("Network protocols");
    b.setBookPrice(2314);
    b.setAuthor(new Author("Wen","US"));
-->
<!-- define book bean, inject dependencies thru setters -->
<bean class="org.example.model.Book" id="book1">
    <property name="bookName" value="Network protocols"/>
    <property name="bookPrice" value="2314"/>
    <property name="author">
        <!-- define author bean -->
        <bean class="org.example.model.Author">
            <constructor-arg value="Wen"/>
            <constructor-arg value="US"/>
        </bean>
    </property>
</bean>
```



```xml
<!--
    Author a1 = new Authr("xyz","delhi");
    Book b = new Book();
    b.setBookName("abcd");
    b.setBookPrice(345);
    b.setAuthor(a1);
-->
<bean class="org.example.model.Author" id="a1">
    <constructor-arg value="xyz"/>
    <constructor-arg value="delhi"/>
</bean>
<bean class="org.example.model.Book" id="book2">
    <property name="bookName" value="abcd"/>
    <property name="bookPrice" value="345"/>
    <property name="author" ref="a1"/>
</bean>
```



```java
public class BookMainApp1 {
    public static void main(String[] args) {
        ApplicationContext container;
        container = new ClassPathXmlApplicationContext( configLocation: "beans2.xml");
        Book b1 = (Book)container.getBean( s: "book1");
        Book b2 = (Book)container.getBean( s: "book2");
        System.out.println(b1);
        System.out.println(b2);

    }
}
```



```java
public class BookMainApp1 {
    public static void main(String[] args) {
        ApplicationContext container;
        container = new ClassPathXmlApplicationContext( configLocation: "beans2.xml");
        Book b1 = (Book)container.getBean( s: "book1");
        Book b2 = (Book)container.getBean( s: "book2");
        System.out.println(b1);
        System.out.println(b2);
        Author a = container.getBean( s: "a1", Author.class);
        System.out.println(a);
    }
}
```