







# Zadaci 1




# Zadatak 1

Zadatak 1: Napisati Spring projekat upotrebom Maven-a koji sadrži klasu Instrument koji predstavlja jedan muzički instrument. Polja klase su naziv tipa String i polje tipa enum TipInstrumenta koji ima vrednosti: UDARALJKE, DUVACKI i ZICANI. Klasa sadrži konstruktor koji postavlja vrednosti polja na prosleđene i metod toString(). Kreirati jedan objekat (zrno i koristiti injekciju kroz konstruktor kako bi prosledili vrednosti polja klase) i ispisati na standardni izlaz.

# Zadatak 1

## Dependencies

Dependencies    

 spring-beans : 5.2.5.RELEASE	Add...
 spring-context : 5.2.5.RELEASE	Remove
 spring-core : 5.2.5.RELEASE	Properties...
	Manage...

# Zadatak 1





```
Zadatak_1/pom.xml  Instrument.java  beans.xml  App.java
1 package rs.ac.singidunum.fir.pj;
2
3 public class Instrument {
4     private String naziv;
5
6     enum TipInstrumenta {
7         UDARALJKE, DUVACKI, ZICANI
8     };
9
10    private TipInstrumenta tip;
11
12    public Instrument(String naziv, TipInstrumenta tip) {
13        this.naziv = naziv;
14        this.tip = tip;
15    }
16
17    public String toString() {
18        return "Instrument [naziv=" + naziv + ", tip=" + tip + "];"
19    }
20 }
```

# Zadatak 1

## Spring Beans

### Beans Overview

Select an element to edit its details.

- ▼  beans
  - ▼  instrument [rs.ac.singidunum.fir.pj.Instrument]
    -  [String] "violina"
    -  [TipInstrumenta] "ZICANI"

# Zadatak 1

```
Zadatak_1/pom.xml  Instrument.java  beans.xml  App.java ✕
1 package rs.ac.singidunum.fir.pj;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.FileSystemXmlApplicationContext;
5
6 public class App {
7
8     public static void main(String[] args) {
9         ApplicationContext context = new FileSystemXmlApplicationContext("C:\\Users\\
10        \\mmravik\\eclipse-workspace\\Zadatak_1\\beans.xml");
11         Instrument instrument = (Instrument) context.getBean("instrument");
12         System.out.println(instrument.toString());
13
14         ((FileSystemXmlApplicationContext)context).close();
15     }
16 }
```

# Zadatak 2

Zadatak 2: Proširiti prethodni zadatak klasom Muzičar koja ima polje instrument koji svira tipa Instrument i polje ime tipa String, konstruktor i metod toString() koji će ispisati i podatke instrumenta. Konfigurisati 3 objekta (kao zrna sa odgovarajućom konfiguracijom) klase Instrument i 4 objekta (ponovo zrna sa odgovarajućom konfiguracijom, referiše se adresa za polje instrument) klase Muzičar.

# Zadatak 2

```
beans.xml  App.java  Instrument.java  Muzicar.java ✕  
1 package rs.ac.singidunum.fir.pj;  
2  
3 public class Muzicar {  
4     private Instrument instrument;  
5     private String ime;  
6  
7     public Muzicar(String ime) {  
8         this.ime = ime;  
9     }  
10  
11     public void setInstrument(Instrument instrument) {  
12         this.instrument = instrument;  
13     }  
14  
15     public String toString() {  
16         return "Muzicar " + ime + " svira " + instrument.toString();  
17     }  
18  
19 }
```



# Zadatak 2

```
beans.xml App.java Instrument.java Muzicar.java
7
8 <bean id="instrument1" class="rs.ac.singidunum.fir.pj.Instrument">
9   <constructor-arg name="naziv" type="String"
10     value="violina"></constructor-arg>
11   <constructor-arg name="tip" type="TipInstrumenta"
12     value="ZICANI">
13   </constructor-arg>
14 </bean>
15 <bean id="instrument2" class="rs.ac.singidunum.fir.pj.Instrument">
16   <constructor-arg name="naziv" type="String"
17     value="tuba"></constructor-arg>
18   <constructor-arg name="tip" type="TipInstrumenta"
19     value="DUVACKI">
20   </constructor-arg>
21 </bean>
22 <bean id="instrument3" class="rs.ac.singidunum.fir.pj.Instrument">
23   <constructor-arg name="naziv" type="String"
24     value="bubanj"></constructor-arg>
25   <constructor-arg name="tip" type="TipInstrumenta"
26     value="UDARALJKE">
27   </constructor-arg>
28 </bean>
```

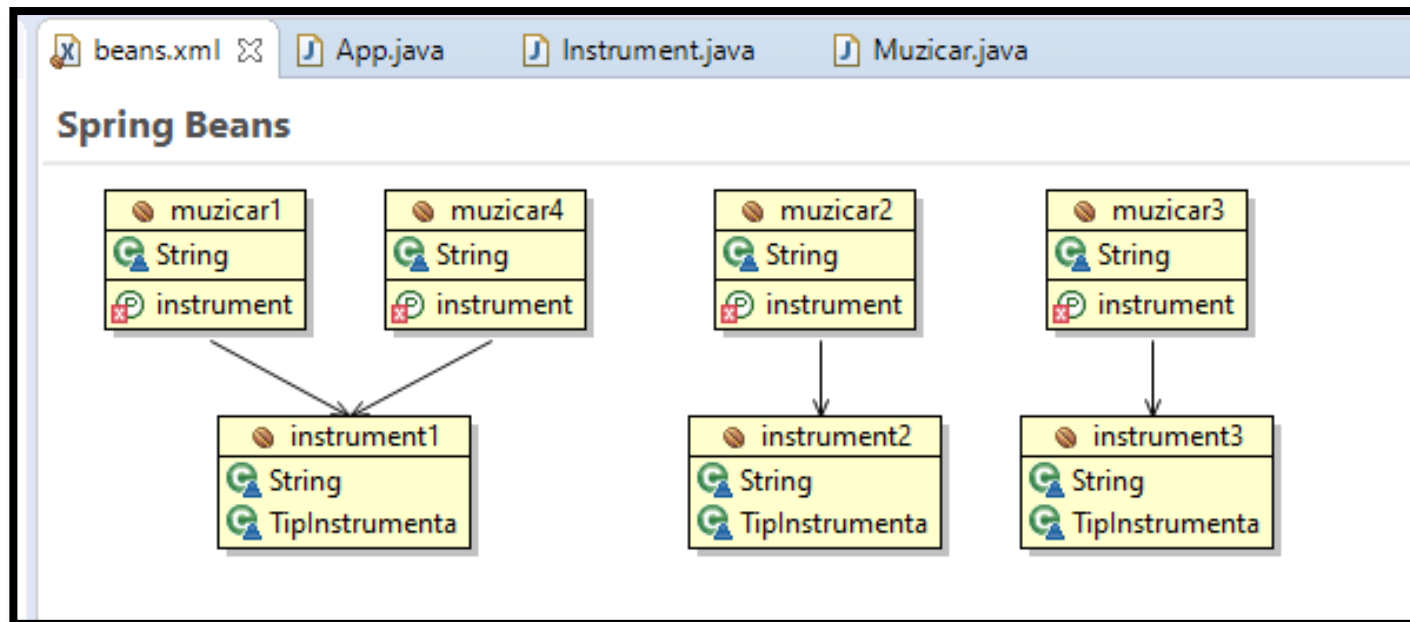
# Zadatak 2

```
<bean id="muzicar1" class="rs.ac.singidunum.fir.pj.Muzicar">
  <constructor-arg name="ime" type="String"
    value="Milos Mravik">
  </constructor-arg>
  <property name="instrument" ref="instrument1"></property>
</bean>
<bean id="muzicar2" class="rs.ac.singidunum.fir.pj.Muzicar">
  <constructor-arg name="ime" type="String"
    value="Pera Peric">
  </constructor-arg>
  <property name="instrument" ref="instrument2"></property>
</bean>
<bean id="muzicar3" class="rs.ac.singidunum.fir.pj.Muzicar">
  <constructor-arg name="ime" type="String"
    value="Marko Markovic">
  </constructor-arg>
  <property name="instrument" ref="instrument3"></property>
</bean>
<bean id="muzicar4" class="rs.ac.singidunum.fir.pj.Muzicar">
  <constructor-arg name="ime" type="String"
    value="Slavko Slavkovic">
  </constructor-arg>
  <property name="instrument" ref="instrument1"></property>
</bean>
</beans>
```

# Zadatak 2

```
beans.xml  *App.java  Instrument.java  Muzicar.java
1 package rs.ac.singidunum.fir.pj;
2
3 import org.springframework.context.ApplicationContext;
4
5
6 public class App {
7
8     public static void main(String[] args) {
9         ApplicationContext context = new FileSystemXmlApplicationContext(
10             "C:\\Users\\mmravik\\eclipse-workspace\\Zadatak_1\\beans.xml");
11         //Instrument instrument = (Instrument) context.getBean("instrument");
12         //System.out.println(instrument.toString());
13
14         Muzicar muzicar1 = (Muzicar) context.getBean("muzicar1");
15         Muzicar muzicar2 = (Muzicar) context.getBean("muzicar2");
16         Muzicar muzicar3 = (Muzicar) context.getBean("muzicar3");
17         Muzicar muzicar4 = (Muzicar) context.getBean("muzicar4");
18
19         System.out.println(muzicar1.toString());
20         System.out.println(muzicar2.toString());
21         System.out.println(muzicar3.toString());
22         System.out.println(muzicar4.toString());
23
24         ((FileSystemXmlApplicationContext) context).close();
25     }
26 }
27
```

# Zadatak 2



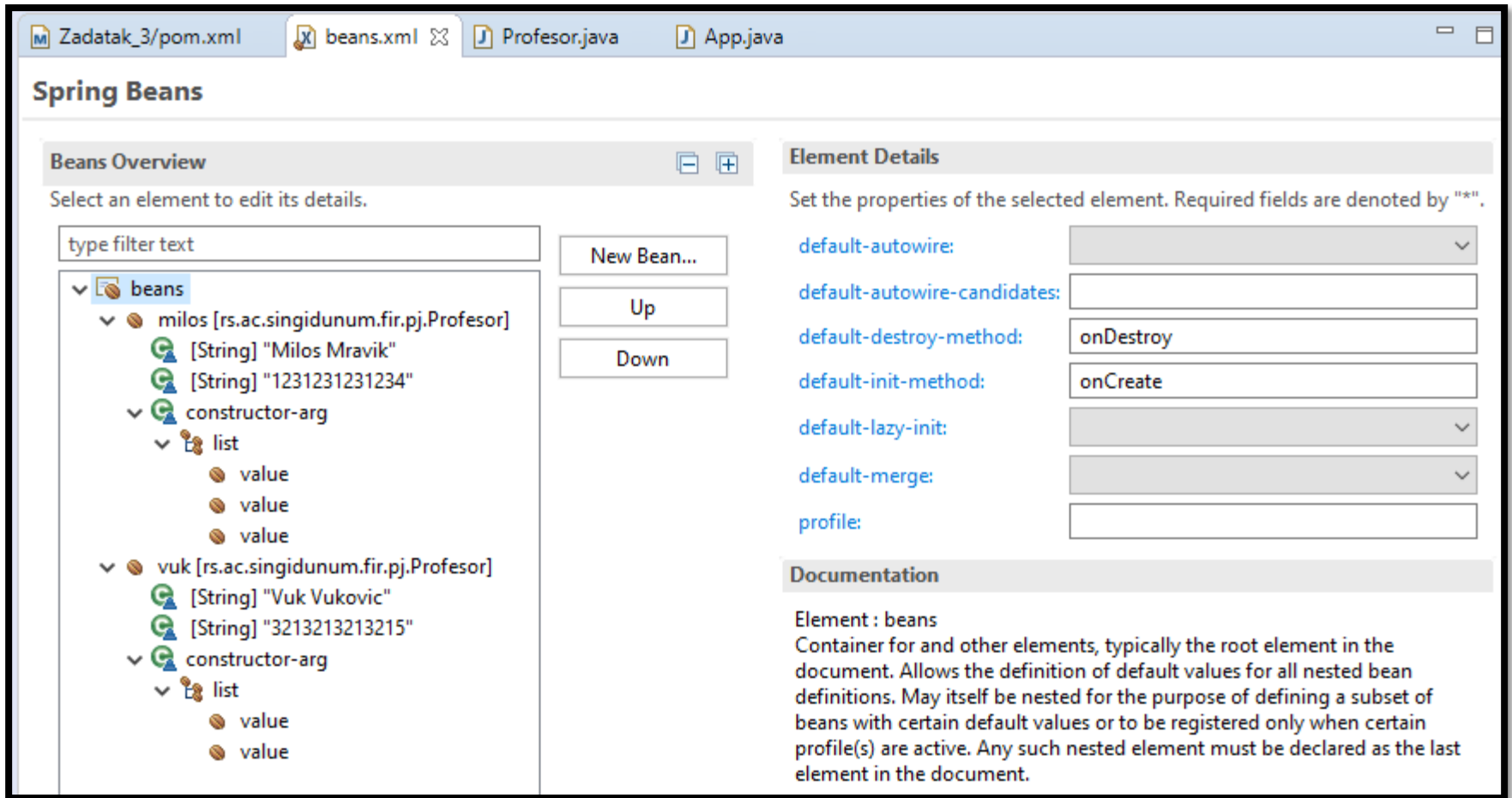
# Zadatak 3 (Zadatak 6 u knjizi)

Zadatak 6: Napisati Spring projekat upotrebom Maven-a koji sadrži klasu Profesor koja ima polja ime, jmbg i listu predmeta koje predaje (lista stringova). Potrebno je ispisati poruku „Profesor *imeProfesora* je počeo da predaje“ svaki put kad se kreira objekat (koristiti metodu onCreate()) i poruku „Profesor *imeProfesora* vise ne predaje“ neposredno pre uništavanja zrna (onDestroy() metod).

# Zadatak 3 (Zadatak 6 u knjizi)

```
Zadatak_3/pom.xml  beans.xml  Profesor.java  App.java
1 package rs.ac.singidunum.fir.pj;
2
3 import java.util.List;
4
5 public class Profesor {
6     private String ime, jmbg;
7     private List<String> predmeti;
8
9     public Profesor(String ime, String jmbg, List<String> predmeti) {
10         this.ime = ime;
11         this.jmbg = jmbg;
12         this.predmeti = predmeti;
13     }
14
15     public String toString() {
16         StringBuilder sb = new StringBuilder();
17         sb.append("Profesor " + ime + "(jmbg: " + jmbg + " predaje sledece predmete:\n");
18         for (String predmet : predmeti) {
19             sb.append(predmet);
20             sb.append("\n");
21         }
22         return sb.toString();
23     }
24
25     public void onCreate() {
26         System.out.println("Profesor " + this.ime + " je pocao da predaje!");
27     }
28
29     public void onDestroy() {
30         System.out.println("Profesor " + this.ime + " vise ne predaje.");
31     }
32
33 }
```

# Zadatak 3 (Zadatak 6 u knjizi)



The screenshot shows the Spring Beans IDE interface with the following components:

- Windows:** Zadatak\_3/pom.xml, beans.xml, Profesor.java, App.java
- Spring Beans Title Bar:** Spring Beans
- Beans Overview:**
  - type filter text
  - New Bean...
  - Up
  - Down
  - tree view:
    - beans
      - milos [rs.ac.singidunum.fir.pj.Profesor]
        - [String] "Milos Mravik"
        - [String] "1231231231234"
        - constructor-arg
          - list
            - value
            - value
            - value
        - vuk [rs.ac.singidunum.fir.pj.Profesor]
          - [String] "Vuk Vukovic"
          - [String] "3213213213215"
          - constructor-arg
            - list
              - value
              - value

- Element Details:**
- Set the properties of the selected element. Required fields are denoted by "\*".
- default-autowire: [dropdown]
- default-autowire-candidates: [text field]
- default-destroy-method: onDestroy
- default-init-method: onCreate
- default-lazy-init: [dropdown]
- default-merge: [dropdown]
- profile: [text field]
- Documentation:**
- Element : beans
- Container for and other elements, typically the root element in the document. Allows the definition of default values for all nested bean definitions. May itself be nested for the purpose of defining a subset of beans with certain default values or to be registered only when certain profile(s) are active. Any such nested element must be declared as the last element in the document.

# Zadatak 3 (Zadatak 6 u knjizi)

```

Zadatak_3/pom.xml  beans.xml  Profesor.java  App.java
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     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.
5     default-destroy-method="onDestroy" default-init-method="onCreate">
6
7
8  <bean id="milos" class="rs.ac.singidunum.fir.pj.Profesor">
9     <constructor-arg name="ime" type="String"
10        value="Milos Mravik">
11    </constructor-arg>
12    <constructor-arg name="jmbg" type="String"
13        value="1231231231234">
14    </constructor-arg>
15
16    <constructor-arg>
17        <list>
18            <value>Programski jezici</value>
19            <value>OOP</value>
20            <value>Osnove programiranja</value>
21        </list>
22    </constructor-arg>
23 </bean>
24 <bean id="vuk" class="rs.ac.singidunum.fir.pj.Profesor">
25    <constructor-arg name="ime" type="String" value="Vuk Vukovic"></constructor-arg>
26    <constructor-arg name="jmbg" type="String"
27        value="3213213213215">
28    </constructor-arg>
29    <constructor-arg>
30        <list>
31            <value>Engleski jezik</value>
32            <value>Ruski jezik</value>
33        </list>
34    </constructor-arg>
35 </bean>
36 </beans>
  
```



# Zadatak 3 (Zadatak 6 u knjizi)

```
Zadatak_3/pom.xml  beans.xml  Profesor.java  App.java ✕
1 package rs.ac.singidunum.fir.pj;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.FileSystemXmlApplicationContext;
5
6 public class App {
7
8     public static void main(String[] args) {
9         ApplicationContext context = new FileSystemXmlApplicationContext("beans.xml");
10        Profesor prof = (Profesor) context.getBean("milos");
11        System.out.println(prof);
12
13        Profesor prof2 = (Profesor) context.getBean("vuk");
14        System.out.println(prof2);
15        ((FileSystemXmlApplicationContext)context).close();
16    }
17 }
```

# Zadatak 3 (Zadatak 6 u knjizi)

```
Profesor Milos Mravik je pocao da predaje!  
Profesor Vuk Vukovic je pocao da predaje!  
Profesor Milos Mravik(jmbg: 1231231231234 predaje sledece predmete:  
Programski jezici  
OOP  
Osnove programiranja  
  
Profesor Vuk Vukovic(jmbg: 3213213213215 predaje sledece predmete:  
Engleski jezik  
Ruski jezik  
  
Profesor Vuk Vukovic vise ne predaje.  
Profesor Milos Mravik vise ne predaje.
```

**Hvala na pažnji!**  
**Pitanja?**