Univerzitet Singidunum

Zadaci 2

www.singidunum.ac.rs

MSc Miloš Mravik

Zadatak 1

Zadatak 1: Napisati Spring projekat upotrebom Maven-a koji sadrži klasu Auto koja sadrži polja marka i tip, klasu Osoba koja sadrži polja ime, prezime i jmbg i klasu SaobracajnaDozvola koji ima polja auto tipa Auto i polje vlasnik tipa Osoba. Klase sadrže konstruktore, set i toString() metode. Kreirati po jedan objekat svake klase (zrna i koristiti injekciju kroz konstruktor kako bi prosledili vrednosti polja klasa). Izvršti automatsko ožičavanje:

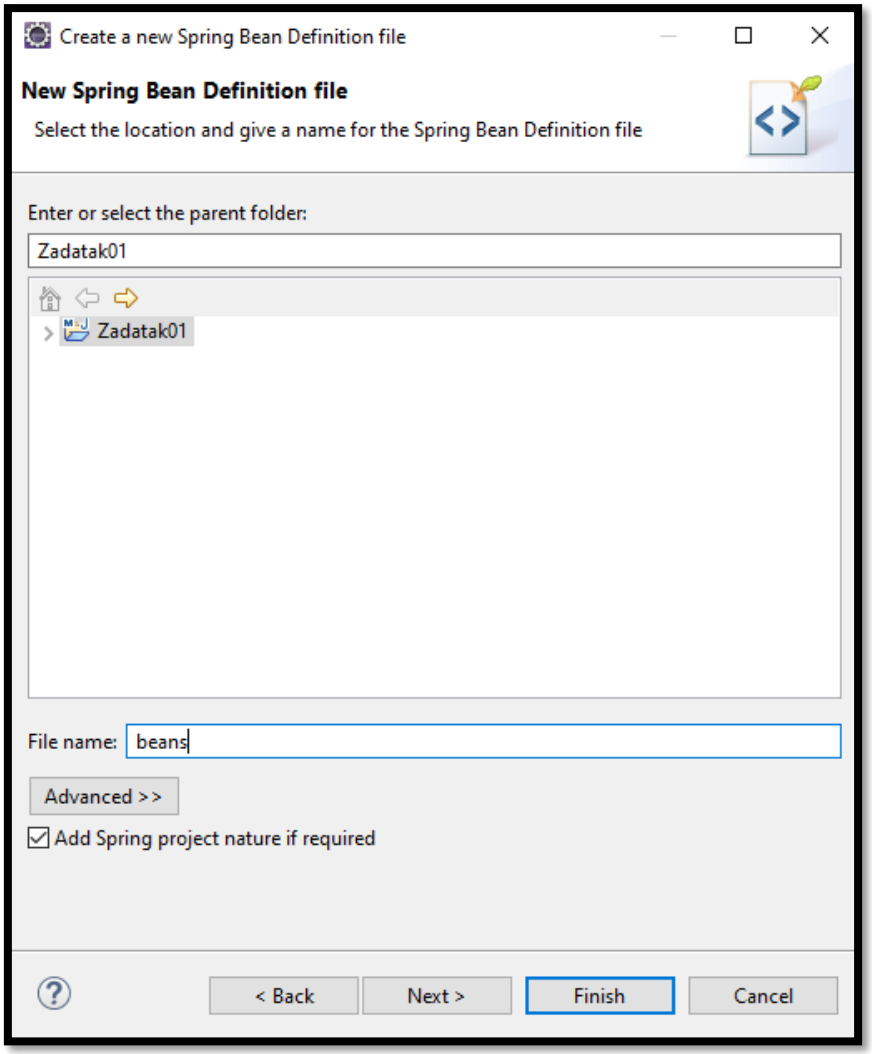
- Po imenu
- Po tipu
- Po konstruktoru

U glavnoj klasi odštampati podatke jedne saobraćajne dozvole.

Zadatak 1


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

Zadatak 1


Create a new Spring Bean Definition file
New Spring Bean Definition file
Select the location and give a name for the Spring Bean Definition file
Enter or select the parent folder:
Zadatak01
Zadatak01
File name:
beans
Advanced >>
Add Spring project nature if required
< Back
Next >
Finish
Cancel

Zadatak 1

Zadatak01/pom.xml | beans.xml | Osoba.java

```java
package rs.ac.singidunum.fir.pj;

public class Osoba {
    private String ime, prezime;
    private long jmbg;

    public Osoba(String ime, String prezime, long jmbg) {
        this.ime = ime;
        this.prezime = prezime;
        this.jmbg = jmbg;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

    public void setPrezime(String prezime) {
        this.prezime = prezime;
    }

    public void setJmbg(long jmbg) {
        this.jmbg = jmbg;
    }

    public String toString() {
        return ime + " " + prezime + " " + jmbg;
    }
}
```

Zadatak 1

Zadatak01/pom.xml | beans.xml | Osoba.java | Auto.java

```java
package rs.ac.singidunum.fir.pj;

public class Auto {
    private String marka, tip;

    public Auto(String marka, String tip) {
        this.marka = marka;
        this.tip = tip;
    }

    public void setMarka(String marka) {
        this.marka = marka;
    }

    public void setTip(String tip) {
        this.tip = tip;
    }

    public String toString() {
        return "Auto [marka=" + marka + ", tip=" + tip + "]";
    }
}
```

Zadatak 1

Zadatak01/pom.xml | beans.xml | Osoba.java | Auto.java | SaobracajnaDozvola.java

```java
package rs.ac.singidunum.fir.pj;

public class SaobracajnaDozvola {
    private Osoba vlasnik;
    private Auto auto;

    public void setVlasnik(Osoba vlasnik) {
        this.vlasnik = vlasnik;
    }

    public void setAuto(Auto auto) {
        this.auto = auto;
    }

    @Override
    public String toString() {
        return "SaobracajnaDozvola [vlasnik=" + vlasnik + ", auto=" + auto + "]";
    }
}
```

Zadatak 1

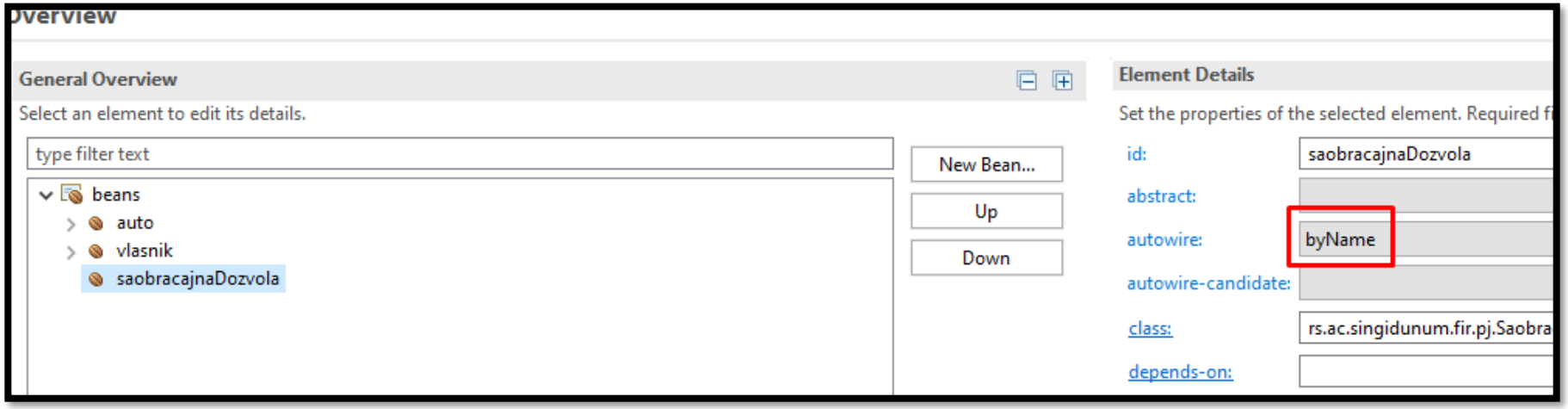

Spring Beans
Beans Overview
Select an element to edit its details.
type filter text
beans
auto
[String] "Hyundai"
[String] "Tucson"
vlasnik
[String] "Milos"
[String] "Mravik"
[Long] "1234567891239"
saobracajnaDozvola
New Bean...
Up
Down
Element Details
index:
name:
marka
ref:
type:
String
value:
Hyundai
Documentation

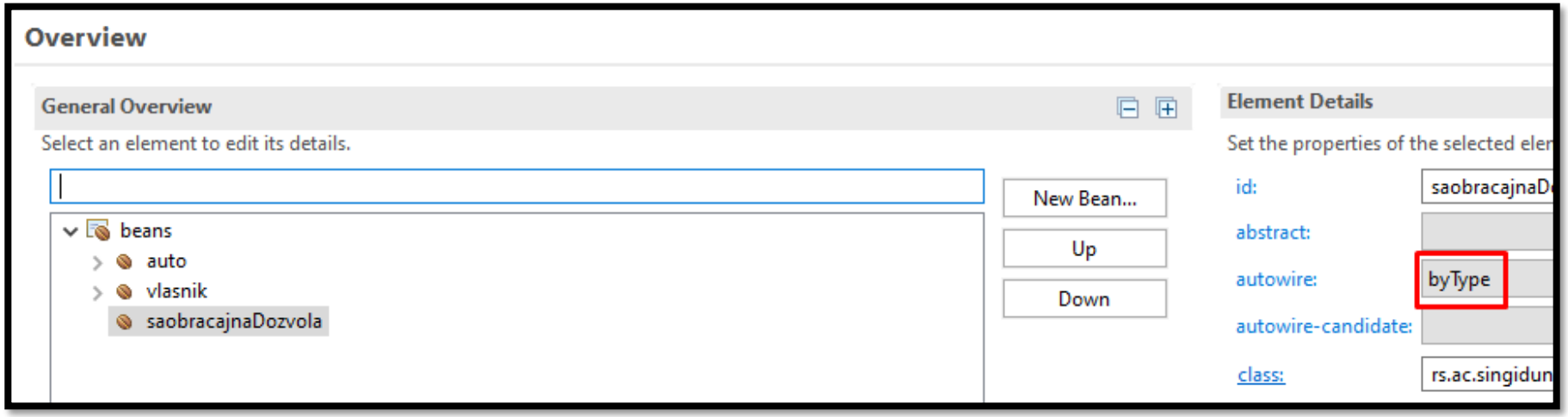
Zadatak 1

Zadatak01/pom.xml | beans.xml | Osoba.java | Auto.java | SaobracajnaDozvola.java | App.java

```java
package rs.ac.singidunum.fir.pj;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;

public class App {

    public static void main(String[] args) {
        ApplicationContext context = new FileSystemXmlApplicationContext("beans.xml");

        SaobracajnaDozvola dozvola = (SaobracajnaDozvola) context.getBean("saobracajnaDozvola");
        System.out.println(dozvola.toString());

        ((FileSystemXmlApplicationContext) context).close();
    }
}
```

Zadatak 1


Overview
General Overview
Select an element to edit its details.
type filter text
beans
auto
vlasnik
saobracajnaDozvola
New Bean...
Up
Down
Element Details
Set the properties of the selected element. Required f
id:
saobracajnaDozvola
abstract:
autowire:
byName
autowire-candidate:
class:
rs.ac.singidunum.fir.pj.Saobra
depends-on:


Overview
General Overview
Select an element to edit its details.
beans
auto
vlasnik
saobracajnaDozvola
New Bean...
Up
Down
Element Details
Set the properties of the selected eler
id:
abstract:
autowire:
byType
autowire-candidate:
class:
rs.ac.singidun

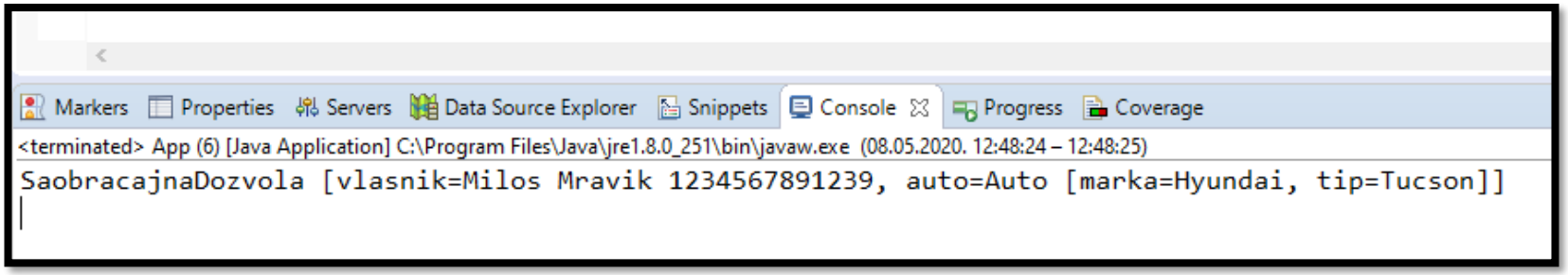
Zadatak 1

Zadatak01/pom.xml | beans.xml | Osoba.java | Auto.java | SaobracajnaDozvola.java | App.java

```java
package rs.ac.singidunum.fir.pj;

public class SaobracajnaDozvola {
    private Osoba vlasnik;
    private Auto auto;

    public void setVlasnik(Osoba vlasnik) {
        this.vlasnik = vlasnik;
    }

    public void setAuto(Auto auto) {
        this.auto = auto;
    }

    @Override
    public String toString() {
        return "SaobracajnaDozvola [vlasnik=" + vlasnik + ", auto=" + auto + "]";
    }

    public SaobracajnaDozvola(Osoba vlasnik, Auto auto) {
        this.vlasnik = vlasnik;
        this.auto = auto;
    }
}
```

Zadatak 1

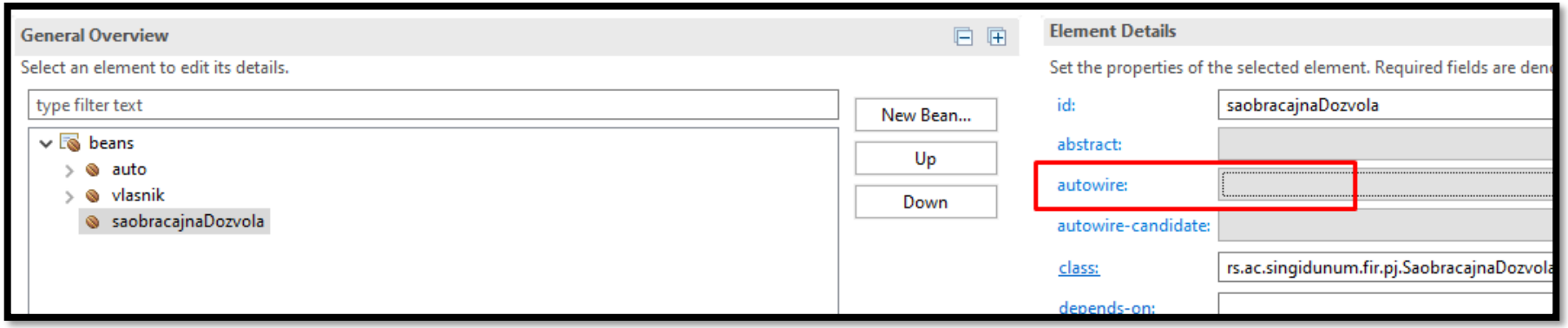

Overview
General Overview
Select an element to edit its details.
type filter text
beans
auto
vlasnik
saobracajnaDozvola
New Bean...
Up
Down
Element Details
Set the properties of the selected element. Re
id:
saobracajnaDozvola
abstract:
autowire:
constructor
autowire-candidate:

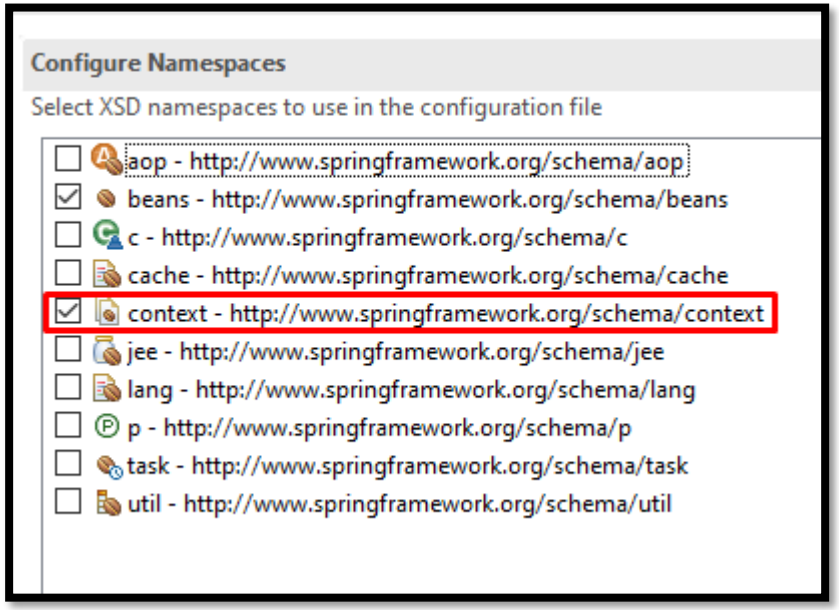

Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Progress
Coverage
<terminated> App (6) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (08.05.2020. 12:48:24 – 12:48:25)
SaobracajnaDozvola [vlasnik=Milos Mravik 1234567891239, auto=Auto [marka=Hyundai, tip=Tucson]]

Zadatak 2

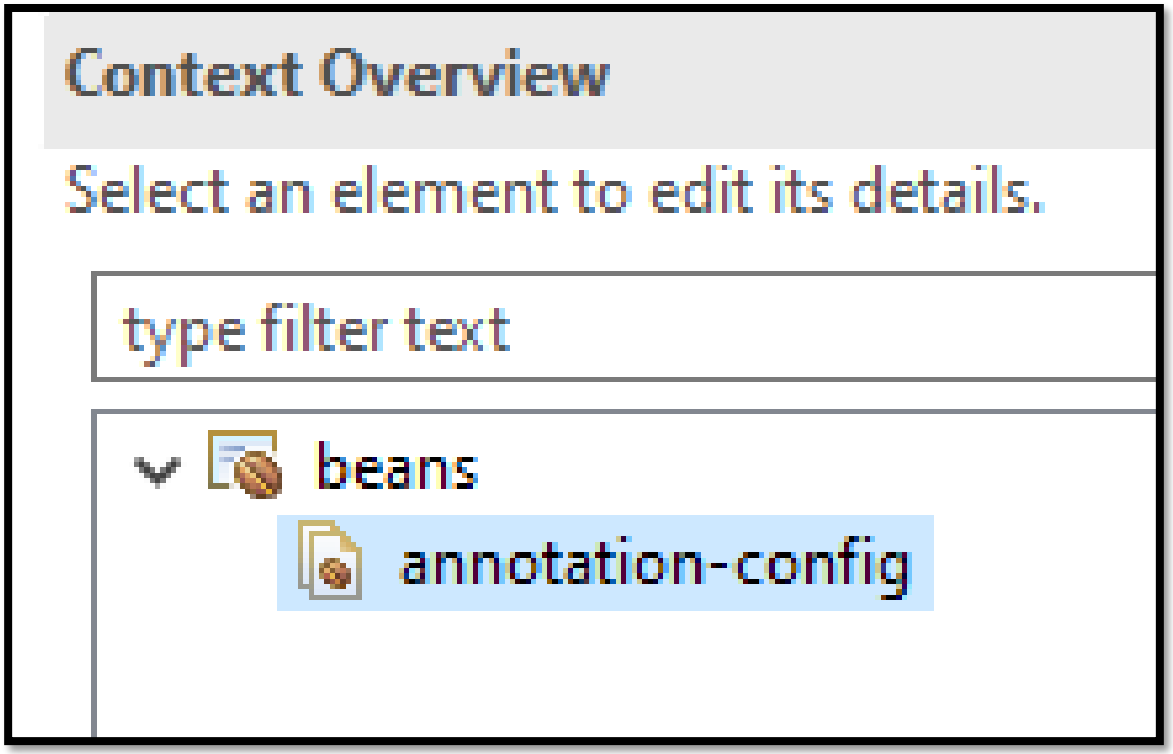
Zadatak 2: Izmeniti prehodni zadatak tako da se automatsko ožičavanje vrši pomoću anotacije.

Zadatak 2


General Overview
Select an element to edit its details.
type filter text
beans
auto
vlasnik
saobracajnaDozvola
New Bean...
Up
Down
Element Details
Set the properties of the selected element. Required fields are den
id:
saobracajnaDozvola
abstract:
autowire:
autowire-candidate:
class:
rs.ac.singidunum.fir.pj.SaobracajnaDozvola
depends-on:


Configure Namespaces
Select XSD namespaces to use in the configuration file
aop - http://www.springframework.org/schema/aop
beans - http://www.springframework.org/schema/beans
c - http://www.springframework.org/schema/c
cache - http://www.springframework.org/schema/cache
context - http://www.springframework.org/schema/context
jee - http://www.springframework.org/schema/jee
lang - http://www.springframework.org/schema/lang
p - http://www.springframework.org/schema/p
task - http://www.springframework.org/schema/task
util - http://www.springframework.org/schema/util

Zadatak 2


Context Overview
Select an element to edit its details.
type filter text
beans
annotation-config

Zadatak 2

Zadatak01/pom.xml | beans.xml | Osoba.java | Auto.java | SaobracajnaDozvola.java | App.java

```java
package rs.ac.singidunum.fir.pj;

import org.springframework.beans.factory.annotation.Autowired;

public class SaobracajnaDozvola {
    //@Autowired
    private Osoba vlasnik;
    //@Autowired
    private Auto auto;

    @Autowired
    public void setVlasnik(Osoba vlasnik) {
        this.vlasnik = vlasnik;
    }

    @Autowired
    public void setAuto(Auto auto) {
        this.auto = auto;
    }

    @Override
    public String toString() {
        return "SaobracajnaDozvola [vlasnik=" + vlasnik + ", auto=" + auto + "]";
    }
    /* @Autowired
    public SaobracajnaDozvola(Osoba vlasnik, Auto auto) {
        this.vlasnik = vlasnik;
        this.auto = auto;
    }
    */
}

//SAOBRACAJNA DOZVOLA NE SME DA IMA KONSTRUKTOR AKO RADIMO OZICAVANJE PO TIPU I IMENU,
//KADA RADIMO PO KONSTRUKTORU ONDA MORA DA IMA KONSTRUKTOR
```

Zadatak 3

Zadatak 3: Napisati Spring projekat upotrebom Maven-a koji sadrži funkcionalni interfejs Posao koji ima metod uradiPosao(). Klase Menadzer i Radnik implementiraju interefejs Posao tako što menadzer ispisuje poruku „Plan rada je napravljen“ dok radnik ispisuje poruku „Dodeljen posao je izvrsen“. Klasa Firma ima dva polje tipa Posao, menadzer i radnik, i metode naparaviPlan() i sprovediPlan(). Prva metoda poziva metodu uradiPosao polja menadzer, a druga metoda poziv uradiPosao polja radnik. Kreirati po jedan objekat svake klase (zrna) i izvršiti automatsko ožičavanje.

Hvala na pažnji!

Pitanja?