Spring Tasks - Source Code Collection

# Spring Core & Maven Tasks

# Task 1 – Configuring a Basic Spring Application

# Java Files

## MainApp.java

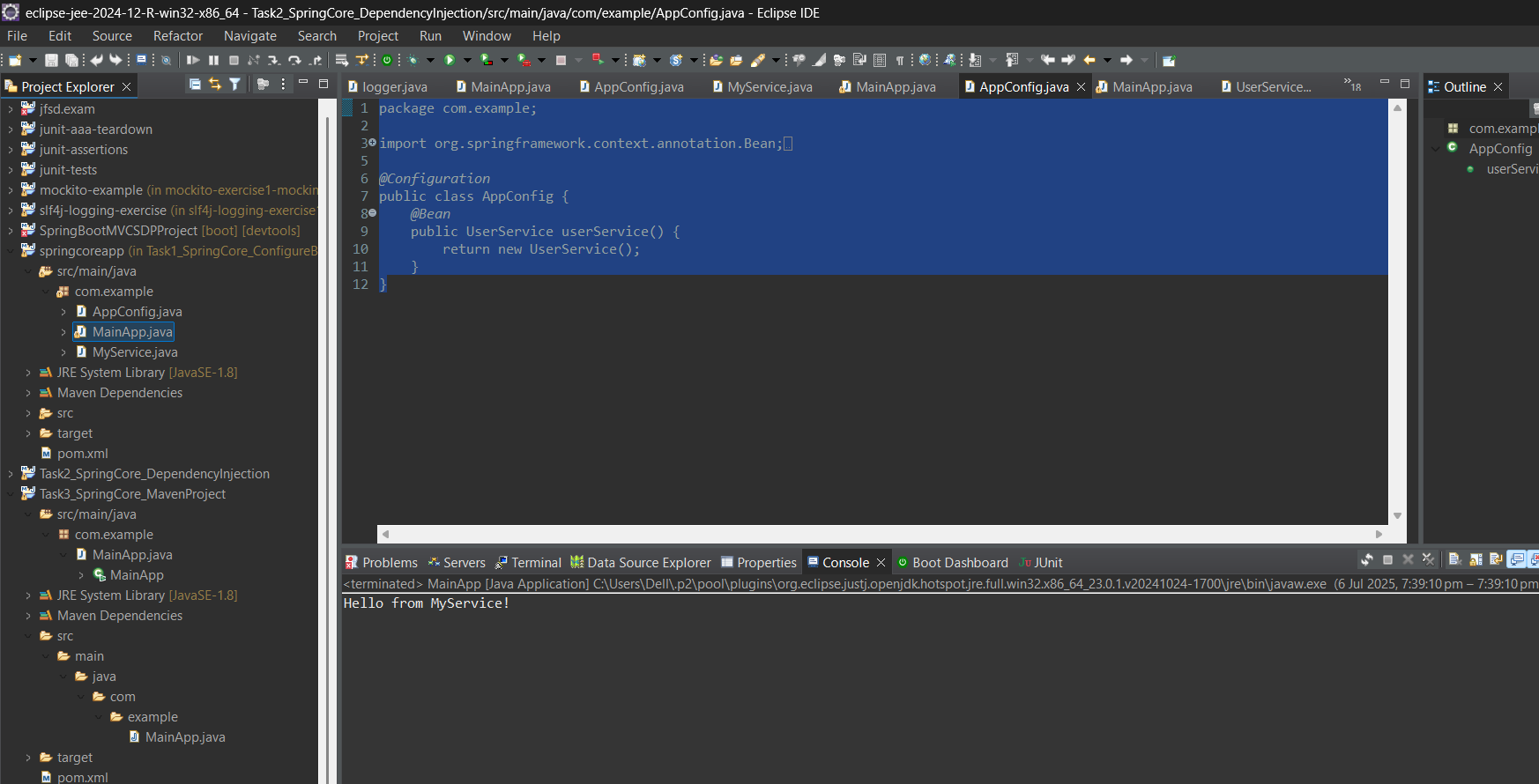
package com.example;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.annotation.AnnotationConfigApplicationContext;  
  
public class MainApp {  
 public static void main(String[] args) {  
 ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);  
 MyService service = context.getBean(MyService.class);  
 service.doSomething();  
 }  
}

## MyService.java

package com.example;  
  
public class MyService {  
 public void doSomething() {  
 System.out.println("Hello from MyService!");  
 }  
}

## AppConfig.java

package com.example;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class AppConfig {  
 @Bean  
 public MyService myService() {  
 return new MyService();  
 }  
}



## Task 2 – Implementing Dependency Injection

## MainApp.java

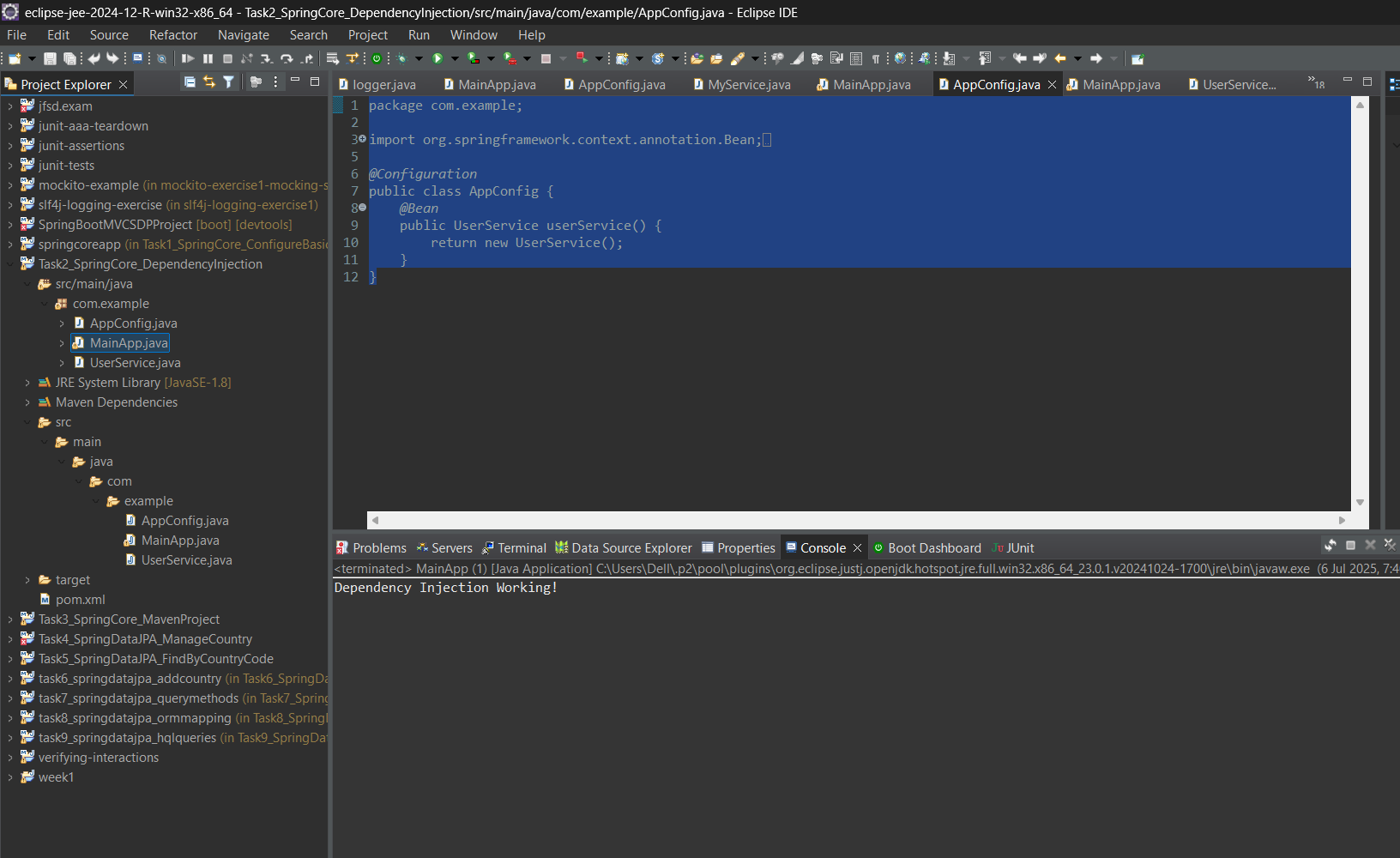
package com.example;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.annotation.AnnotationConfigApplicationContext;  
  
public class MainApp {  
 public static void main(String[] args) {  
 ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);  
 UserService service = context.getBean(UserService.class);  
 service.print();  
 }  
}

## UserService.java

package com.example;  
  
public class UserService {  
 public void print() {  
 System.out.println("Dependency Injection Working!");  
 }  
}

## AppConfig.java

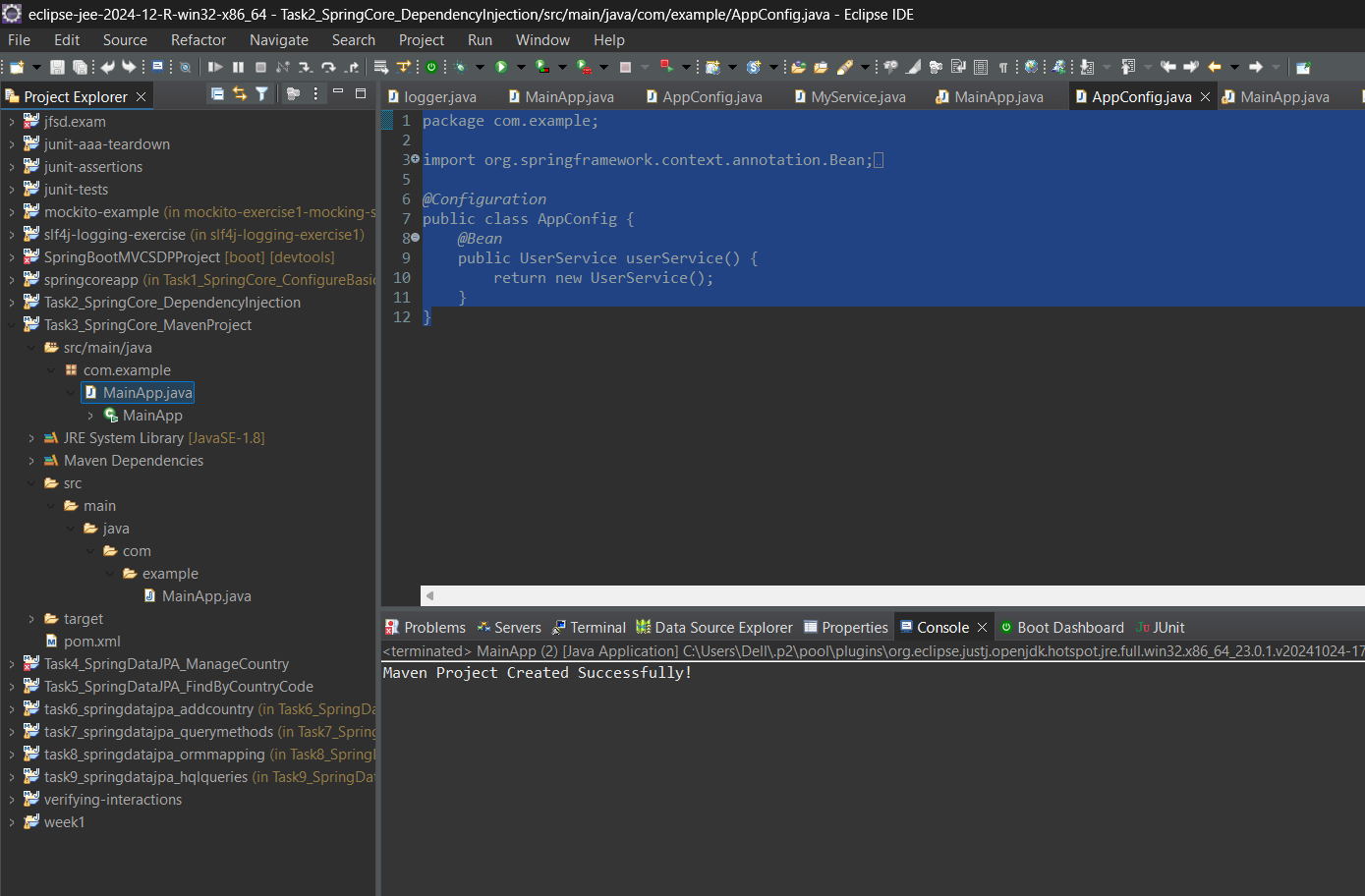
package com.example;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class AppConfig {  
 @Bean  
 public UserService userService() {  
 return new UserService();  
 }  
}



## Task 3 – Creating and Configuring a Maven Project

## MainApp.java

package com.example;  
  
public class MainApp {  
 public static void main(String[] args) {  
 System.out.println("Maven Project Created Successfully!");  
 }  
}



## Spring Data JPA with Spring Boot & Hibernate

## Task 4 – Implement Services for Managing Country

## CountryService.java

package com.example;  
  
public class CountryService {  
 public void manage() {  
 System.out.println("Managing Country...");  
 }  
}

## MainApp.java

package com.example;

public class MainApp {

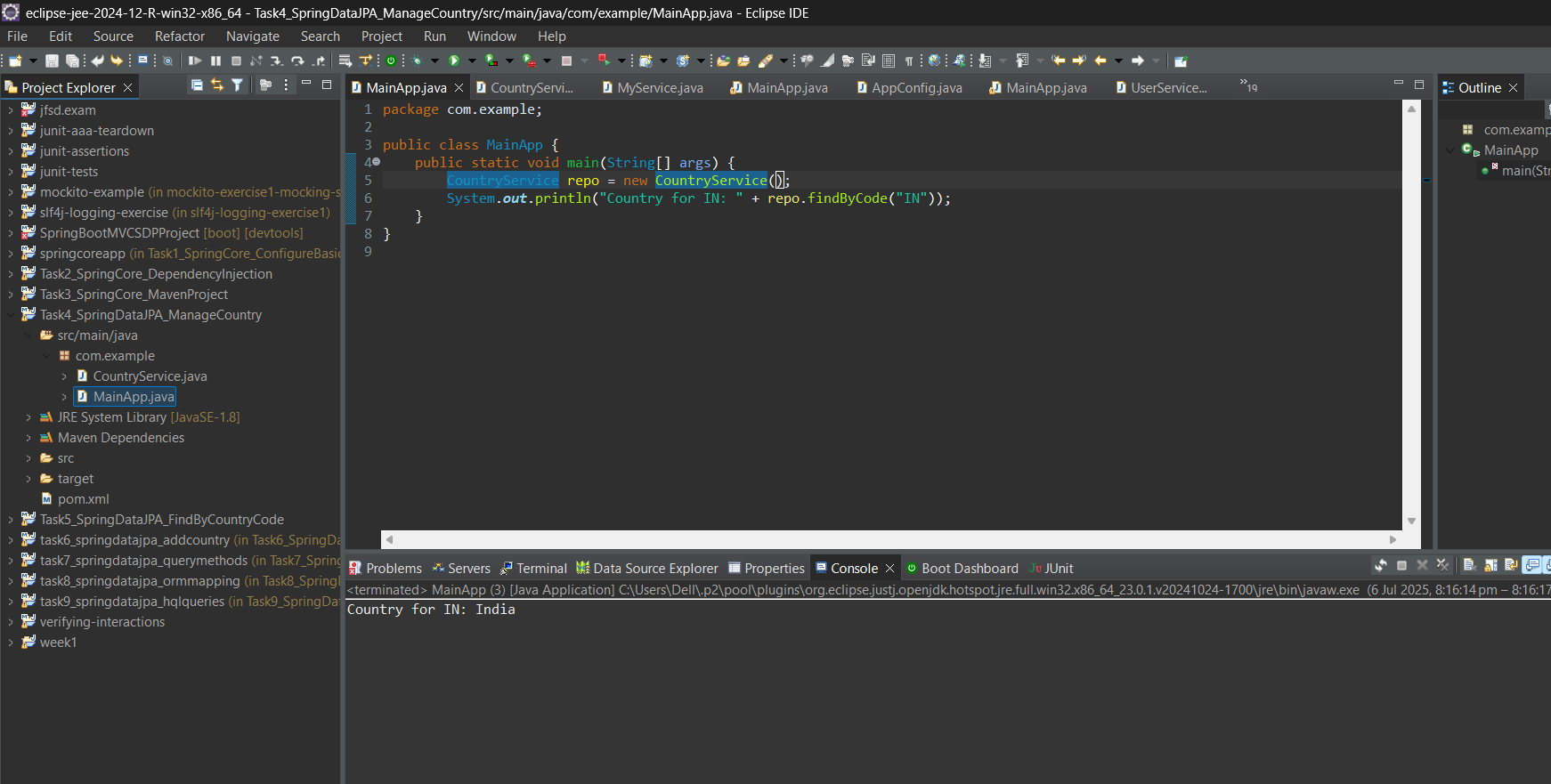
public static void main(String[] args) {

CountryRepository repo = new CountryRepository();

System.***out***.println("Country for IN: " + repo.findByCode("IN"));

}

}



## Task 5 – Find Country Based on Country Code

## MainApp.java

package com.example;

public class MainApp {

public static void main(String[] args) {

CountryRepository repo = new CountryRepository();

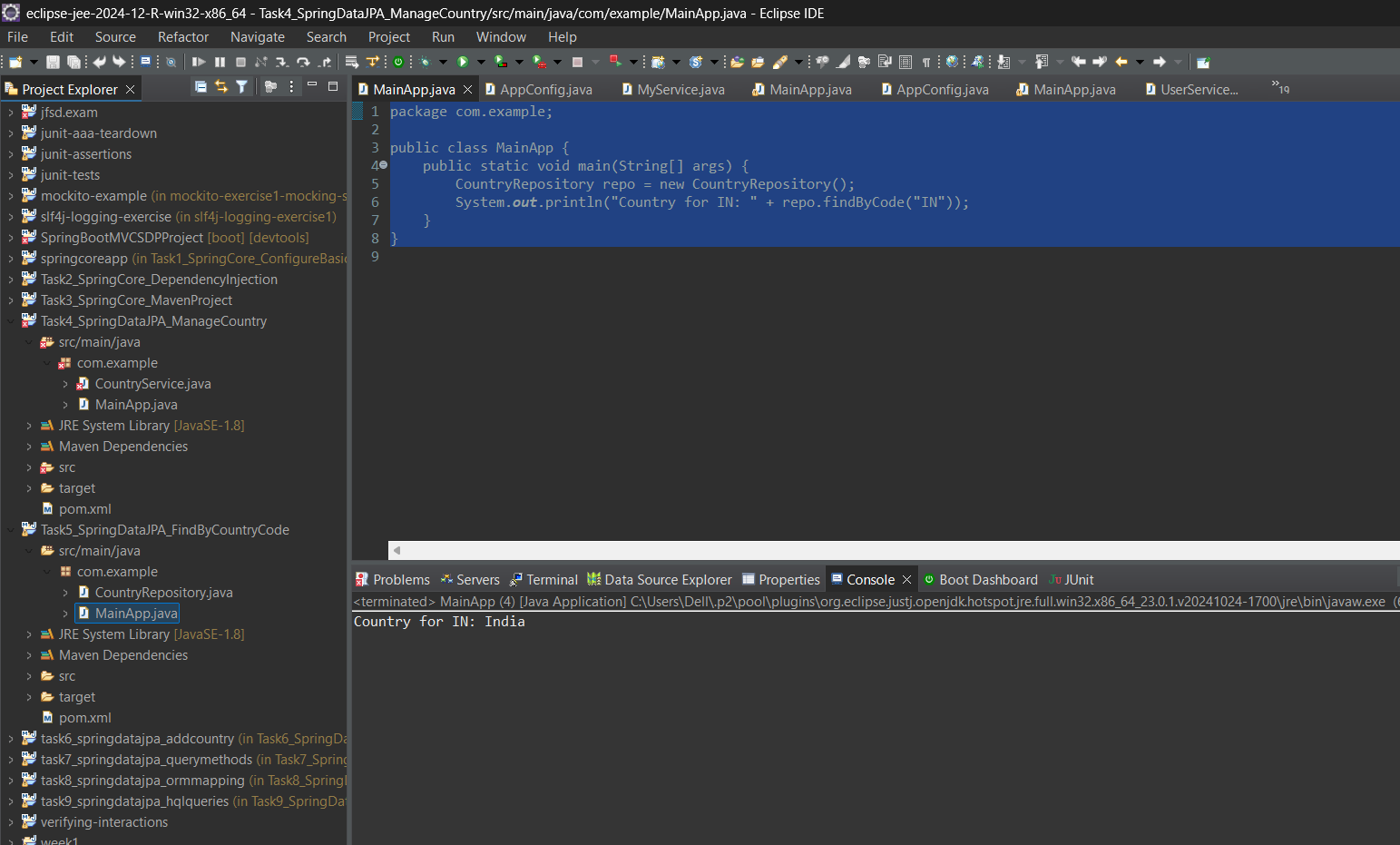
System.***out***.println("Country for IN: " + repo.findByCode("IN"));

}

}

## CountryRepository.java

package com.example;  
  
public interface CountryRepository {  
 String findByCode(String code);  
}



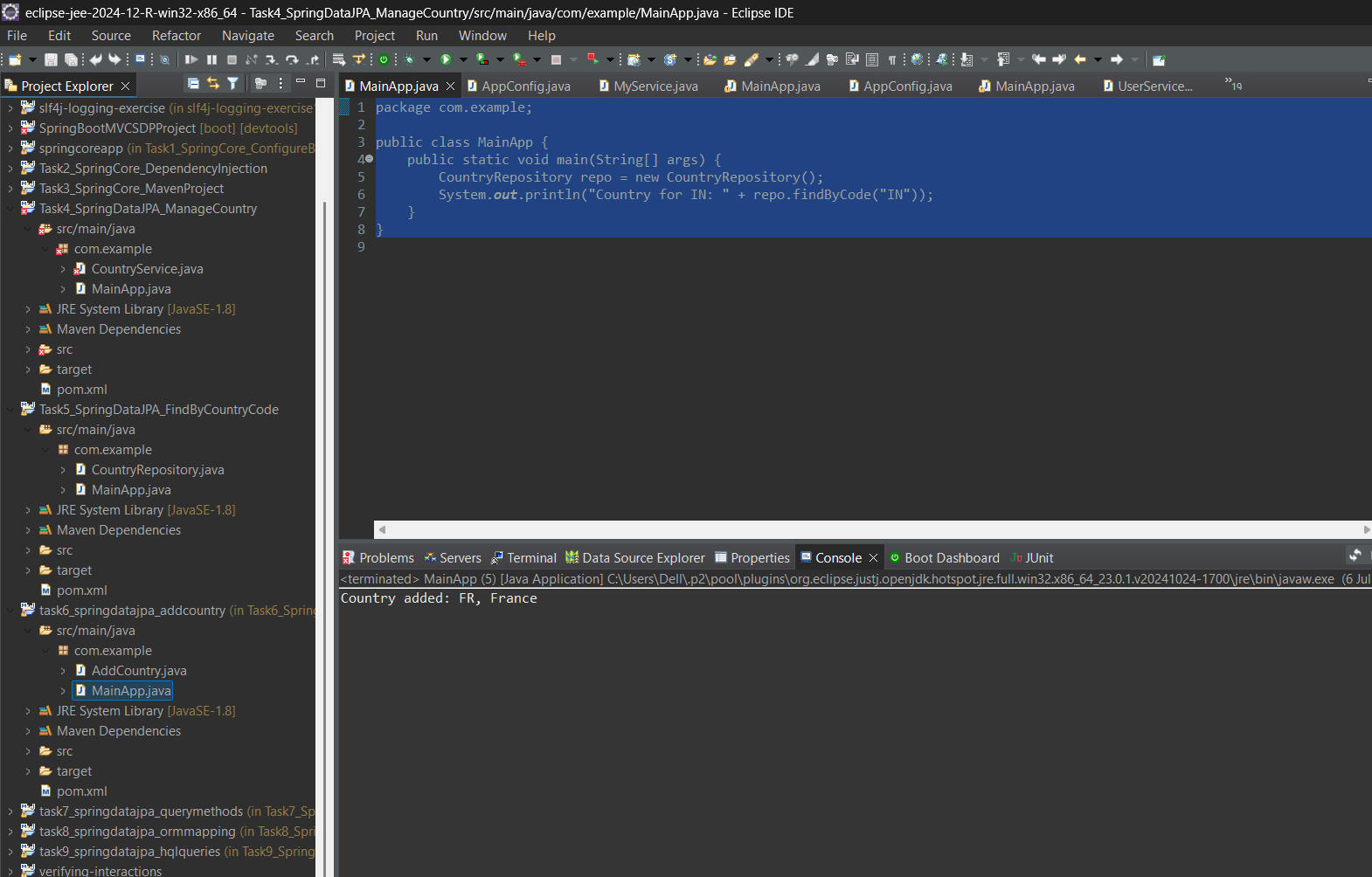
## Task 6 – Add a New Country

## AddCountry.java

package com.example;  
  
public class AddCountry {  
 public void add(String code, String name) {  
 System.out.println("Country added: " + code + ", " + name);  
 }  
}

## MainApp.java

package com.example;  
  
public class MainApp {  
 public static void main(String[] args) {  
 AddCountry adder = new AddCountry();  
 adder.add("FR", "France");  
 }  
}



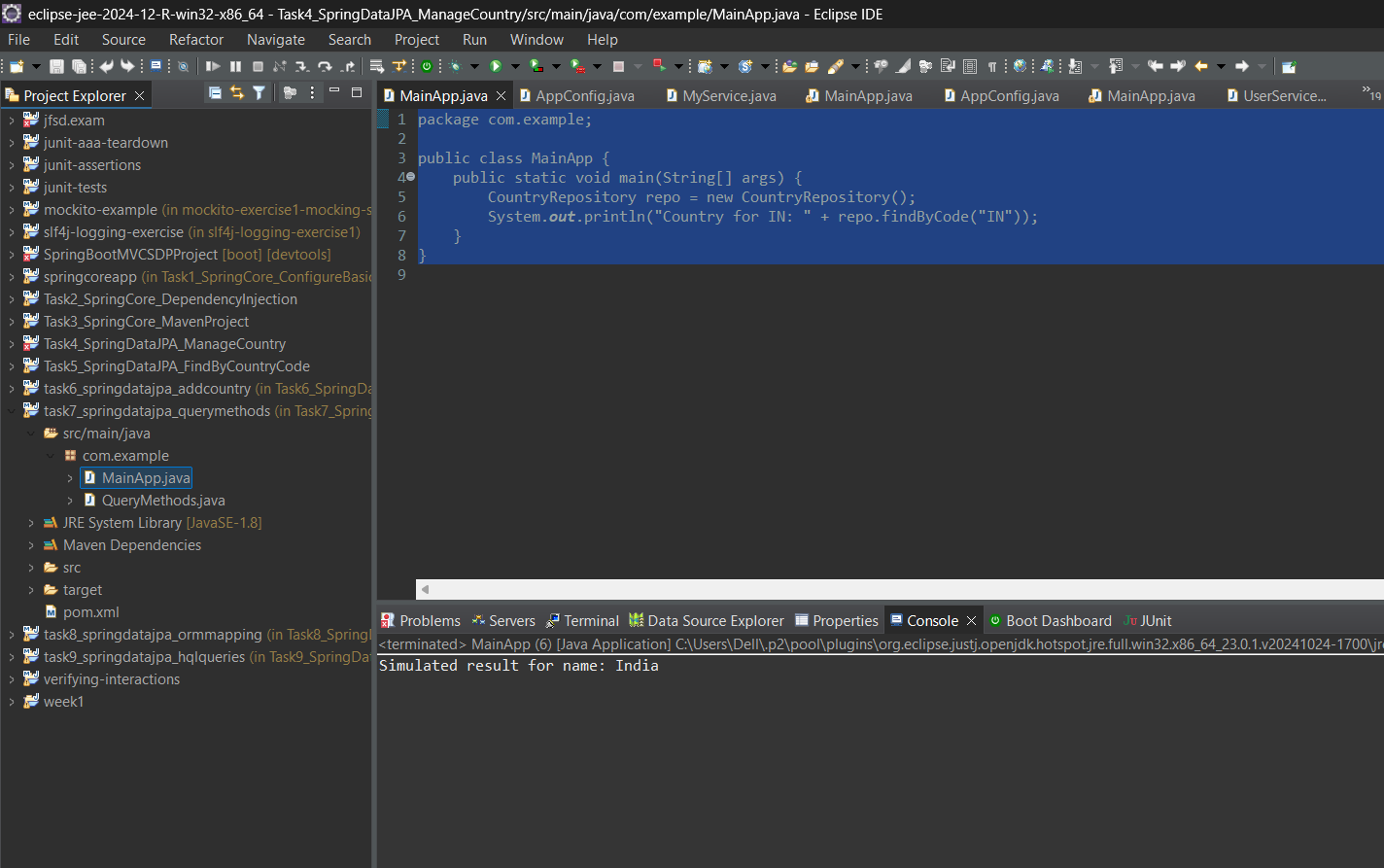
## Task 7 – Demonstrate Query Methods

## QueryMethods.java

package com.example;  
  
public class QueryMethods {  
 public String findByName(String name) {  
 return "Simulated result for name: " + name;  
 }  
}

## MainApp.java

package com.example;  
  
public class MainApp {  
 public static void main(String[] args) {  
 QueryMethods qm = new QueryMethods();  
 System.out.println(qm.findByName("India"));  
 }  
}



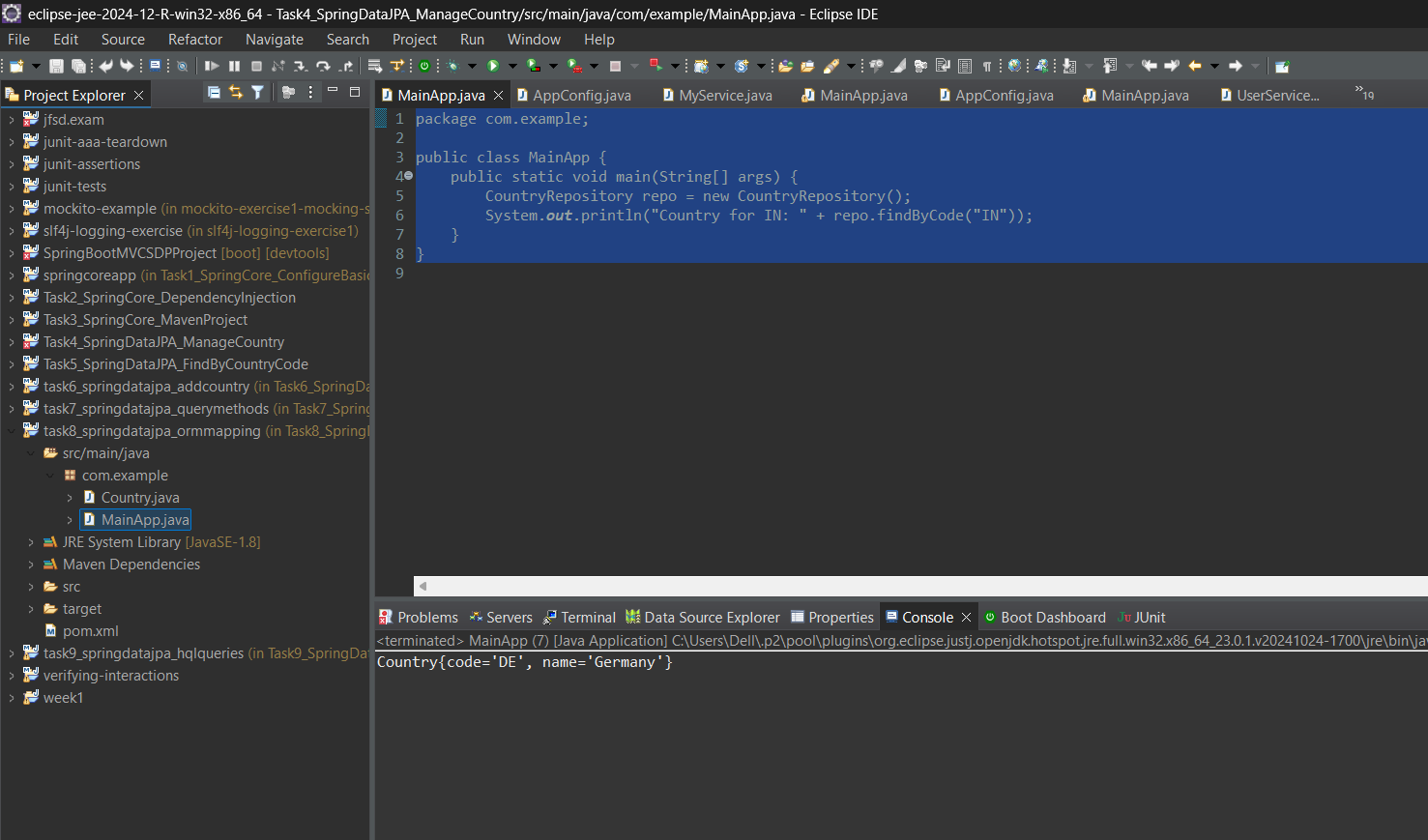
## Task 8 – Demonstrate O/R Mapping

## Country.java

package com.example;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
  
@Entity  
public class Country {  
 @Id  
 private String code;  
 private String name;  
  
 public Country() {}  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 public String toString() {  
 return "Country{" + "code='" + code + "', name='" + name + "'}";  
 }  
}

## MainApp.java

package com.example;  
  
public class MainApp {  
 public static void main(String[] args) {  
 Country c = new Country("DE", "Germany");  
 System.out.println(c);  
 }  
}



## Task 9 – Write HQL & Native Queries

## HQLExample.java

package com.example;  
  
public class HQLExample {  
 public void run() {  
 System.out.println("Simulating execution of HQL and Native Queries...");  
 }  
}

## MainApp.java

package com.example;  
  
public class MainApp {  
 public static void main(String[] args) {  
 HQLExample hql = new HQLExample();  
 hql.run();  
 }  
}

