**Week-3 Hands-on**

**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

import org.springframework.context.annotation.\*;

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);

HelloService helloService = context.getBean(HelloService.class);

helloService.sayHello();

}

}

class HelloService {

public void sayHello() {

System.*out*.println("Hello from Spring!");

}

}

*@Configuration*

class AppConfig {

*@Bean*

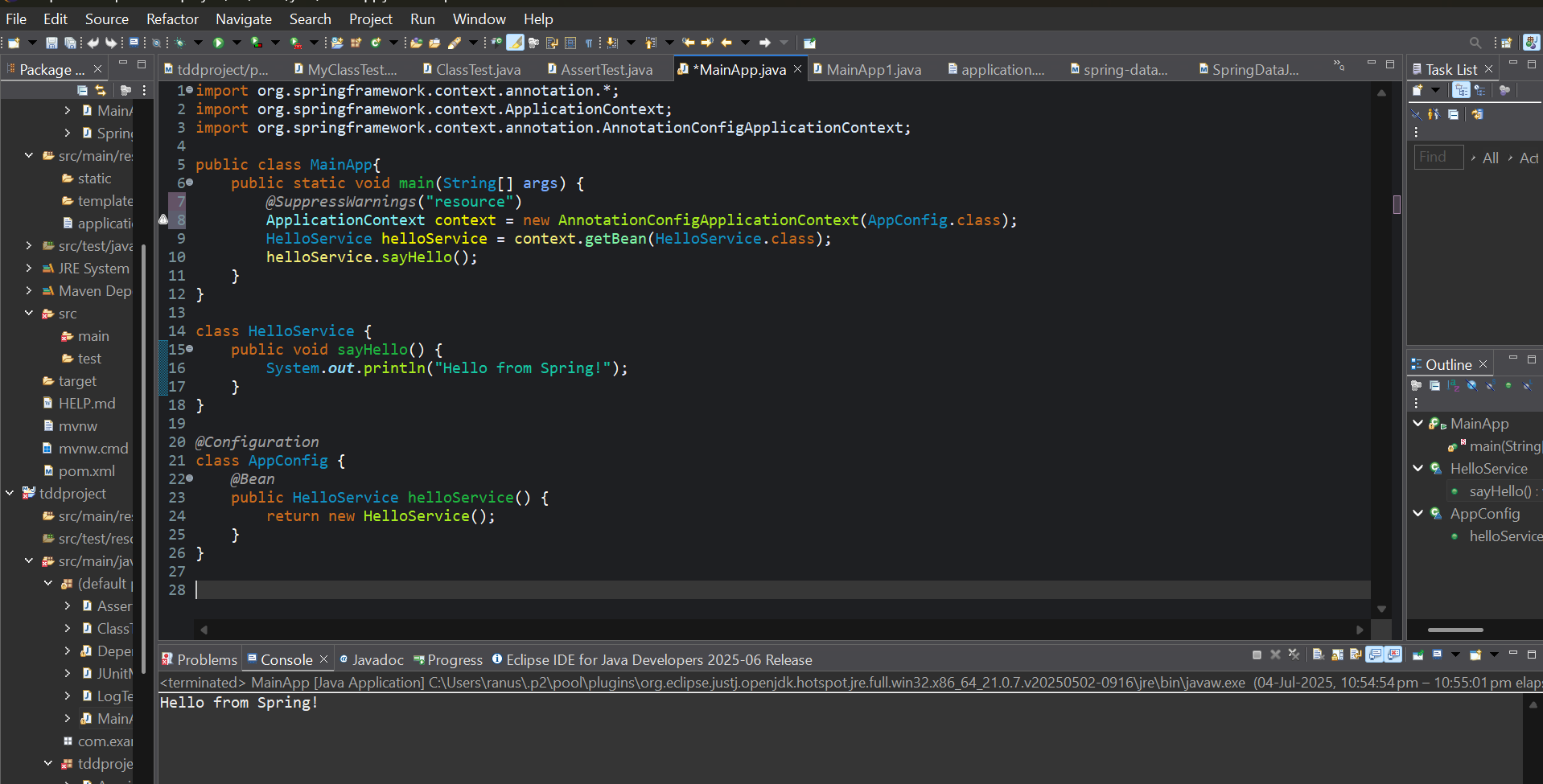
public HelloService helloService() {

return new HelloService();

}

}

**Output:**

****

**Exercise 2: Implementing Dependency Injection**

interface GreetingService {

void greet();

}

class EnglishGreetingService implements GreetingService {

public void greet() {

System.*out*.println("Good Morning!");

}

}

class Greeter {

private GreetingService greetingService;

public void setGreetingService(GreetingService service) {

this.greetingService = service;

}

public void greet() {

greetingService.greet();

}

}

class Config {

*@Bean*

public GreetingService greetingService() {

return new EnglishGreetingService();

}

*@Bean*

public Greeter greeter() {

Greeter greeter = new Greeter();

greeter.setGreetingService(greetingService());

return greeter;

}

}

public class DependencyInjection {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(Config.class);

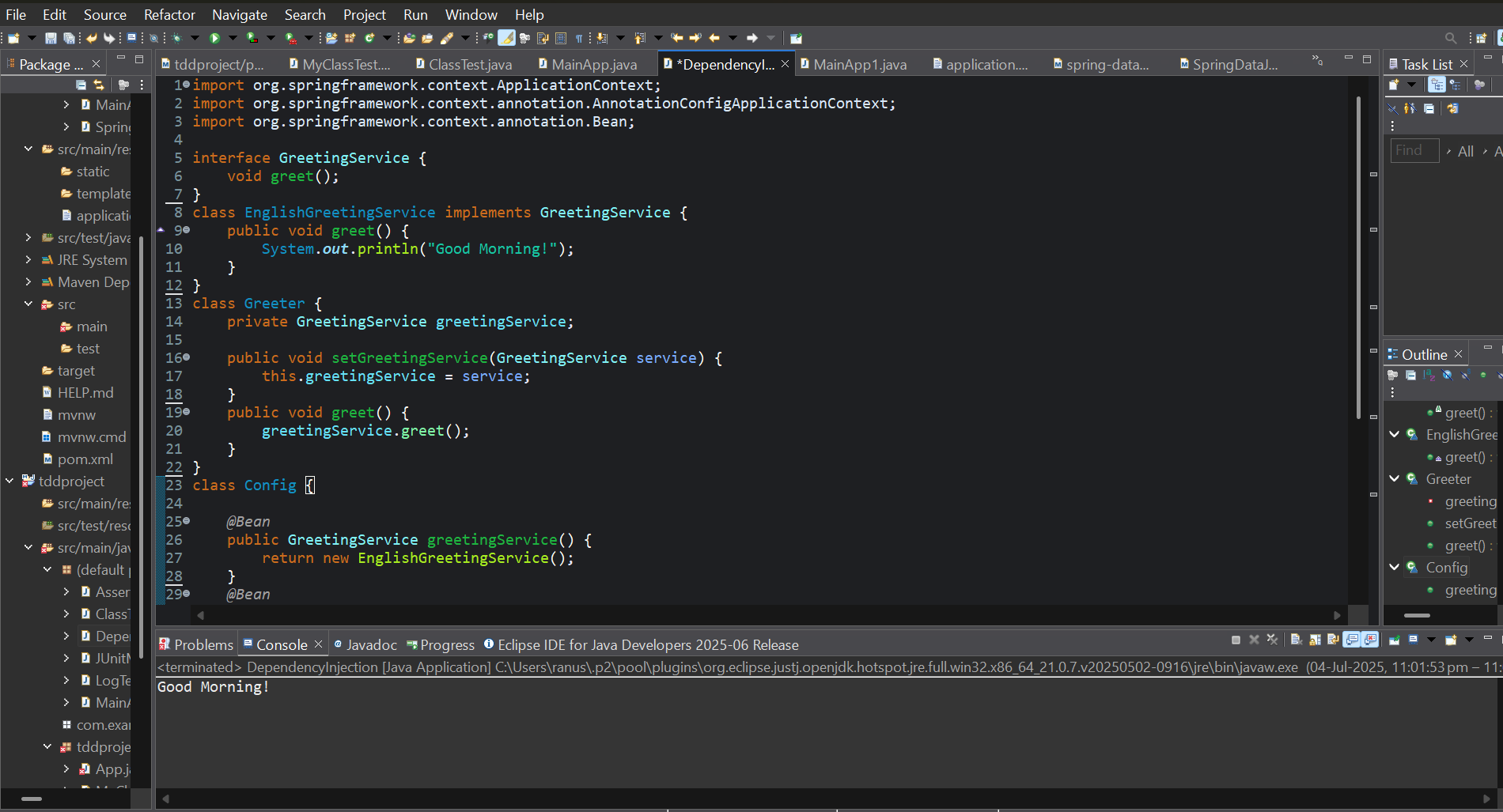
Greeter greeter = context.getBean(Greeter.class);

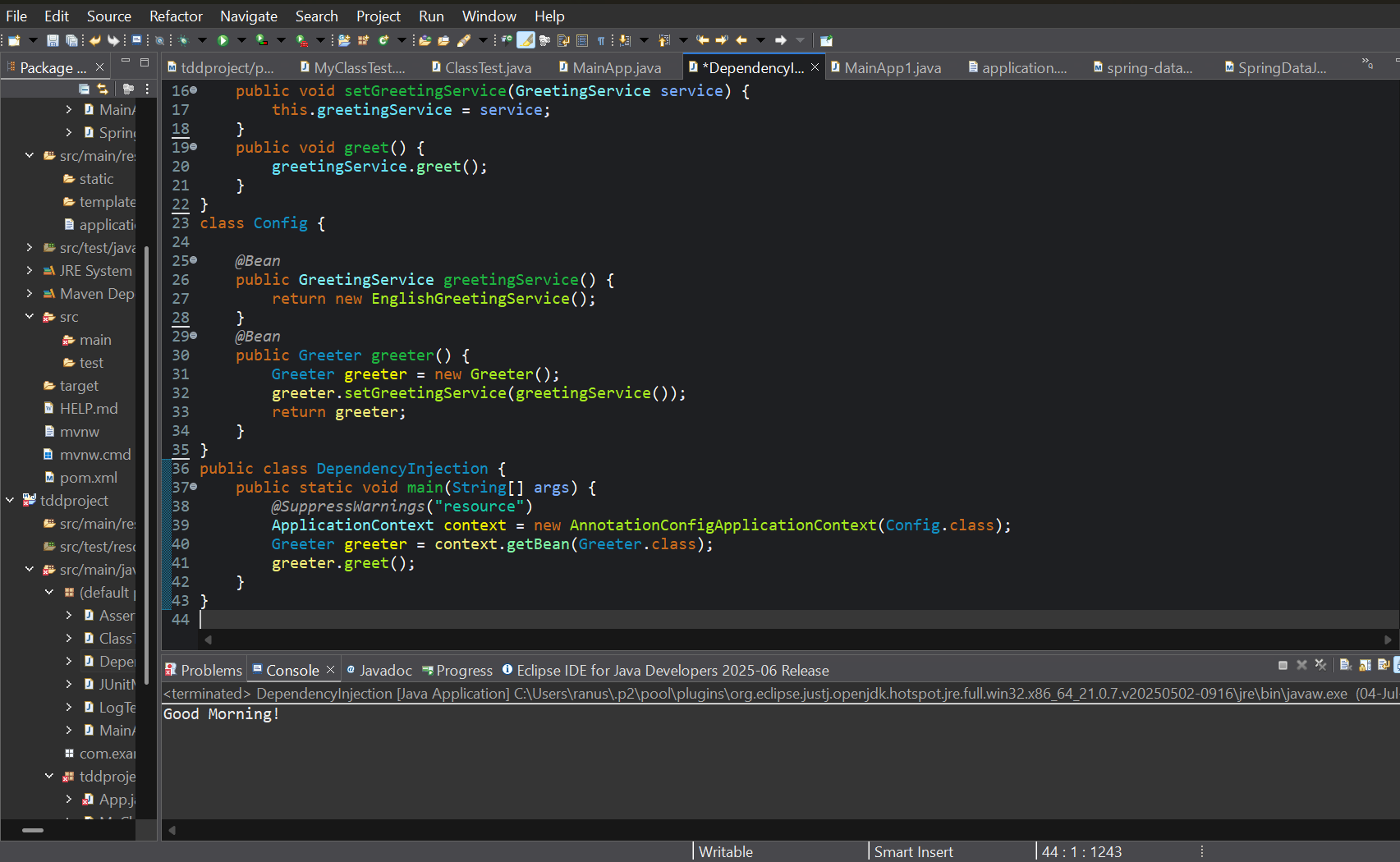
greeter.greet();

}

}

**Output:**

****

****

**Exercise 3: Creating and Configuring a Maven Project**

import org.springframework.context.annotation.\*;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class MainApp{

public static void main(String[] args) {

*@SuppressWarnings*("resource")

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

HelloService helloService = context.getBean(HelloService.class);

helloService.sayHello();

}

}

class HelloService {

public void sayHello() {

System.out.println("Hello from a Maven Project");

}

}

*@Configuration*

class AppConfig {

*@Bean*

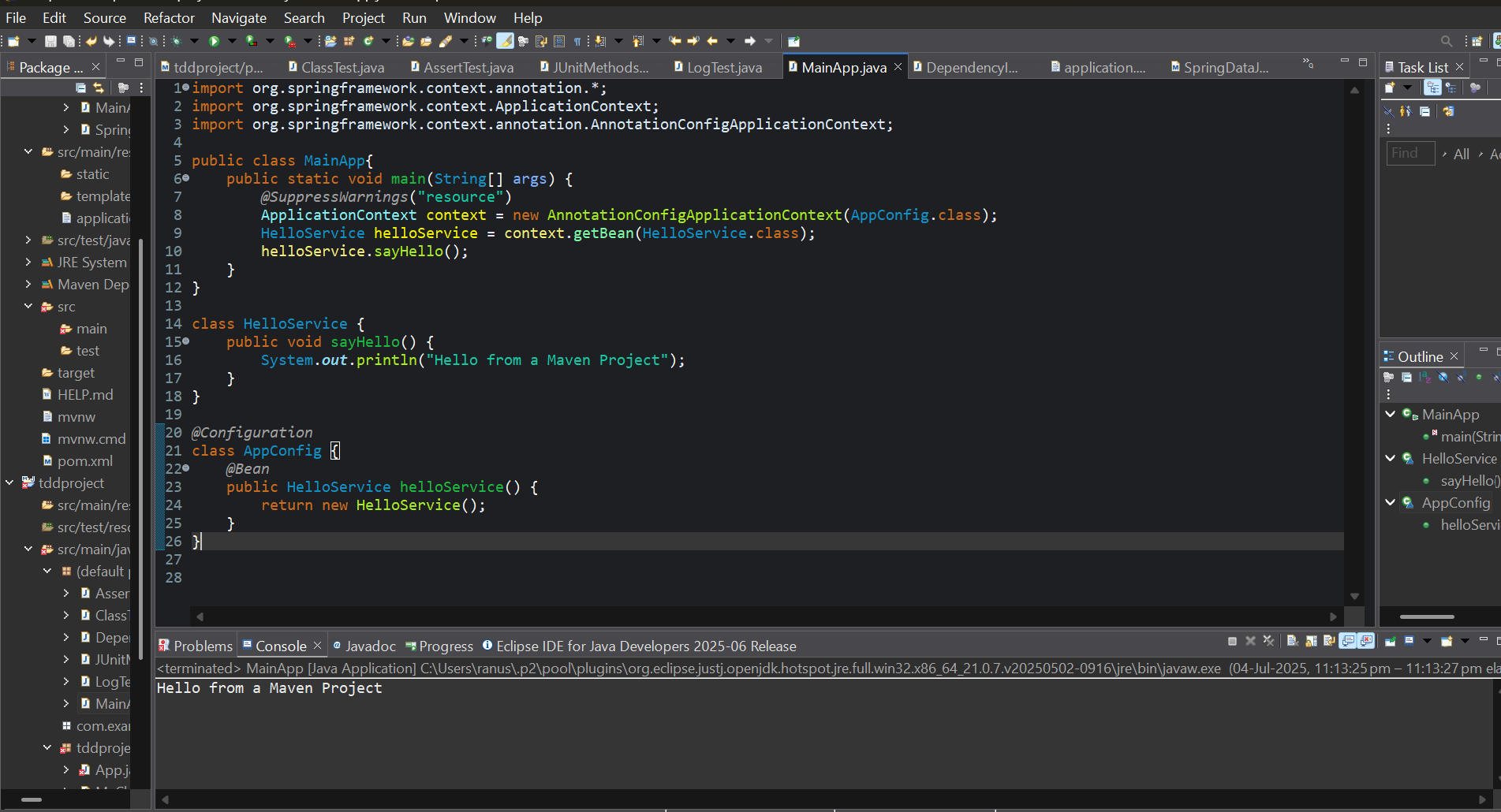
public HelloService helloService() {

return new HelloService();

}

}

**Output:**

****

**Spring Data JPA with Spring Boot, Hibernate**

**Exercise 1: Spring Data JPA - Quick Example**

import jakarta.persistence.\*;

import org.springframework.data.jpa.repository.JpaRepository;

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

import org.springframework.web.bind.annotation.\*;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.List;

@Entity

@Table(name = "users")

class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

interface UserRepository extends JpaRepository<User, Long> {

}

@RestController

class UserController {

@Autowired

private UserRepository userRepository;

@PostMapping("/users")

public User createUser(@RequestBody User user) {

return userRepository.save(user);

}

@GetMapping("/users")

public List<User> getAllUsers() {

return userRepository.findAll();

}

}

@SpringBootApplication

public class MainApp1{

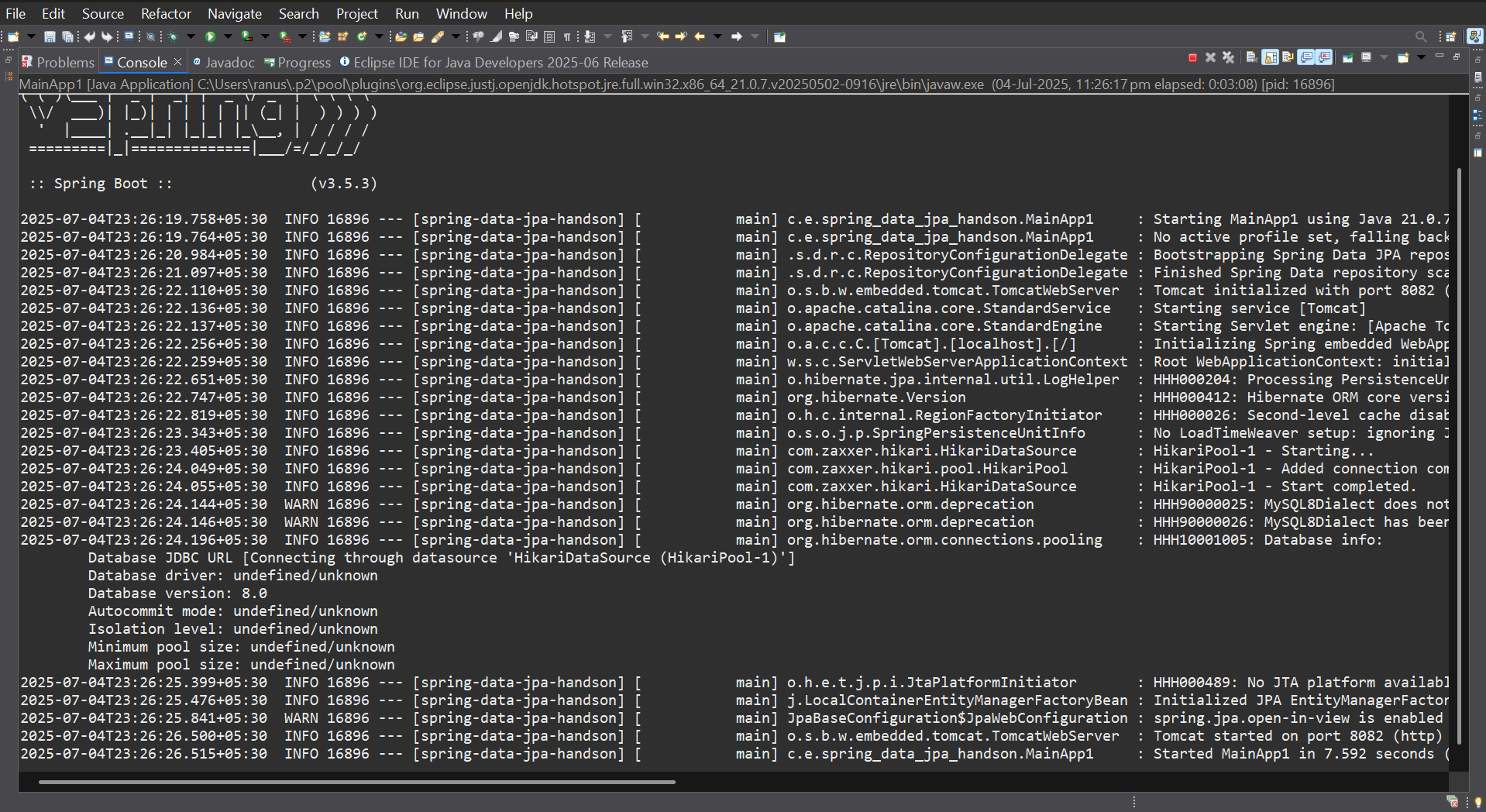
public static void main(String[] args) {

SpringApplication.run(MainApp1.class, args);

}

}

**Output:**

****

**Exercise 2: Difference between JPA, Hibernate, and Spring Data JPA**

**import jakarta.persistence.\*;**

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

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.data.jpa.repository.JpaRepository;**

**import org.springframework.web.bind.annotation.\*;**

**import java.util.List;**

***@Entity***

***@Table*(name = "users") // optional but good practice**

**class User {**

***@Id***

***@GeneratedValue*(strategy = *GenerationType*.*IDENTITY*)**

**private Long id;**

**private String name;**

**public Long getId() { return id; }**

**public void setId(Long id) { this.id = id; }**

**public String getName() { return name; }**

**public void setName(String name) { this.name = name; }**

**}**

**interface UserRepository extends JpaRepository<User, Long> {**

**}**

***@RestController***

***@RequestMapping*("/users")**

**class UserController {**

***@Autowired***

**private UserRepository userRepository;**

***@PostMapping***

**public User createUser(*@RequestBody* User user) {**

**return userRepository.save(user);**

**}**

***@GetMapping***

**public List<User> getAllUsers() {**

**return userRepository.findAll();**

**}**

**}**

***@SpringBootApplication***

**public class DemoApplication {**

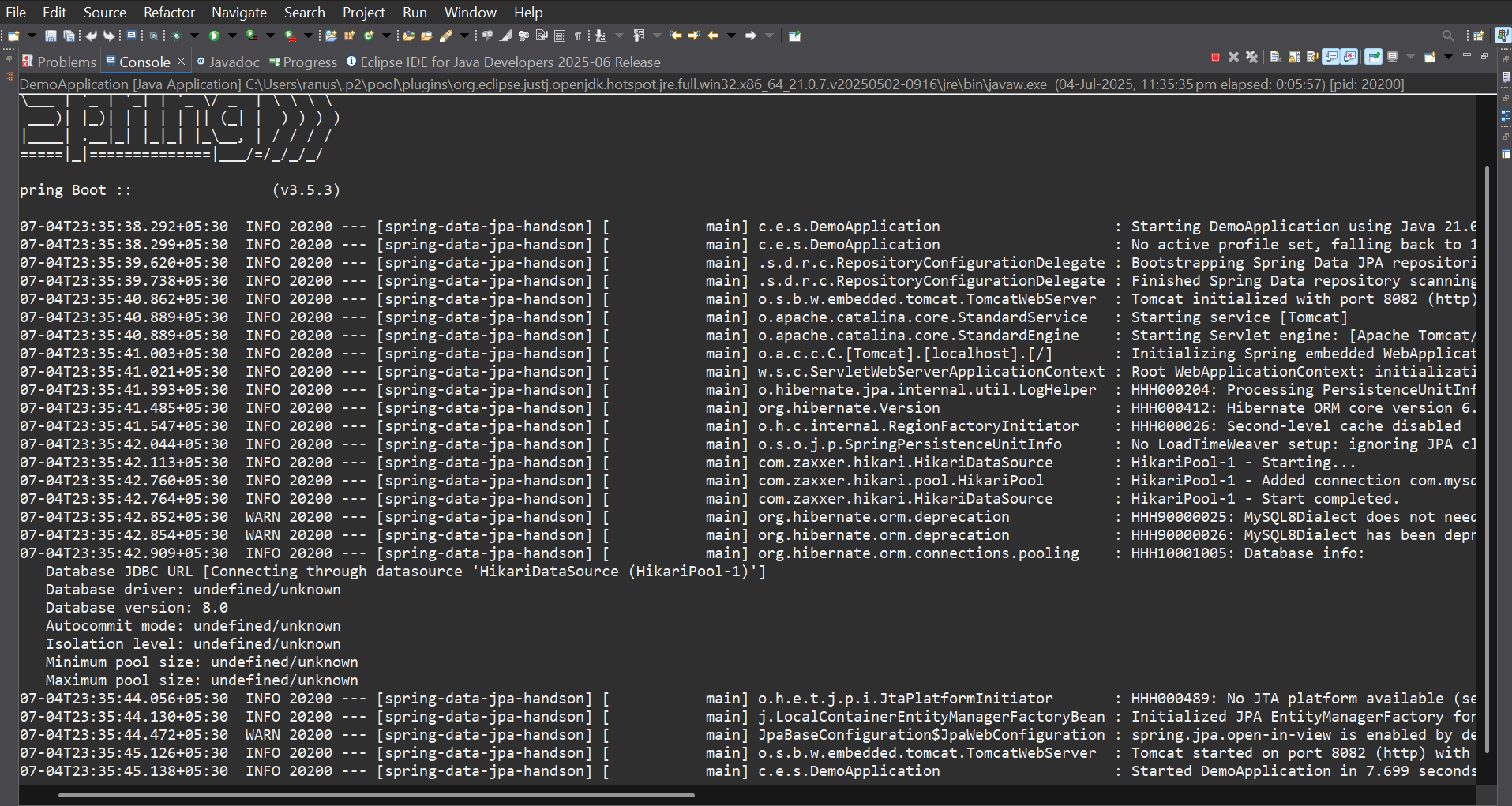
**public static void main(String[] args) {**

**SpringApplication.*run*(DemoApplication.class, args);**

**}**

**}**

**Output:**

****