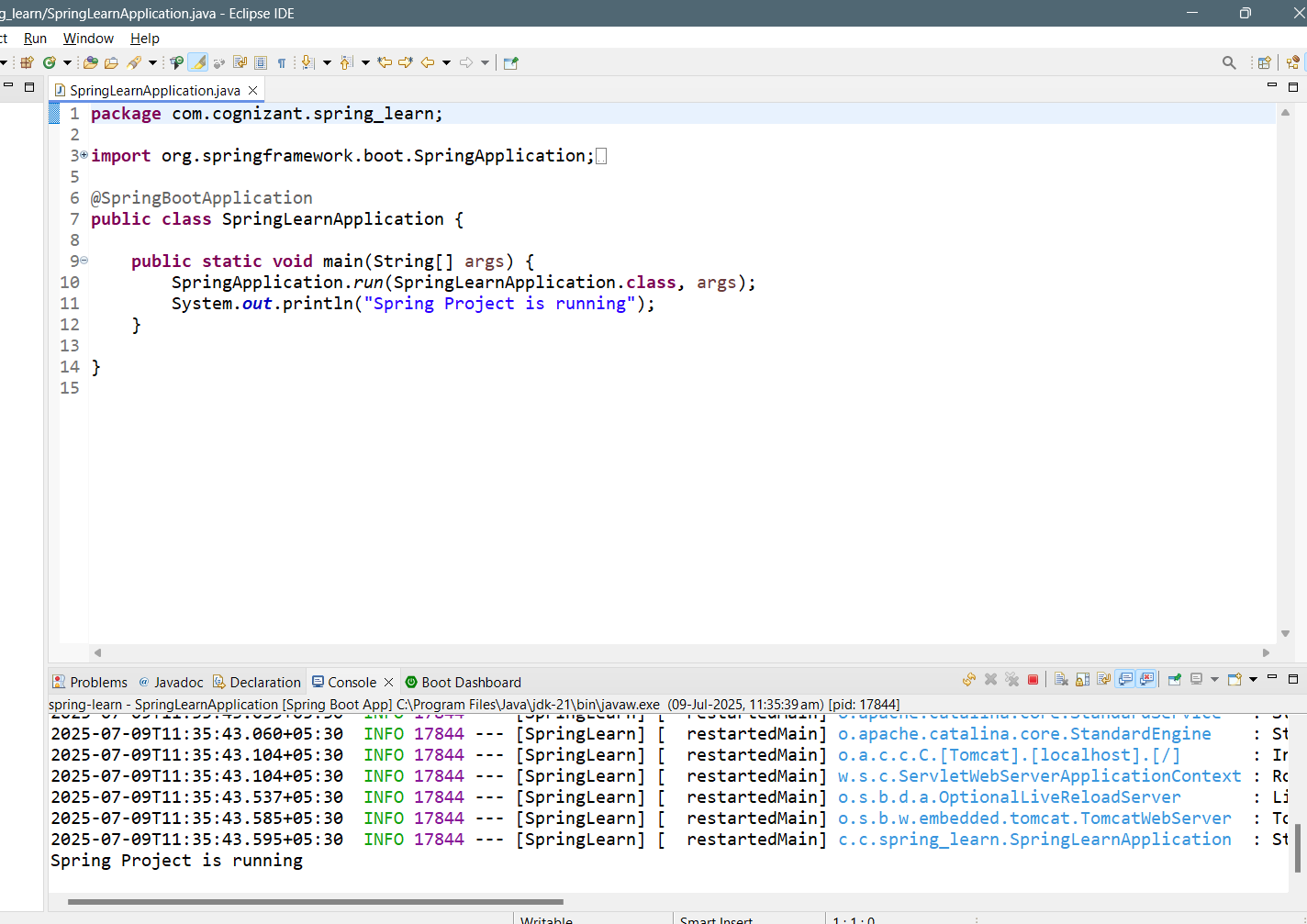
**Hands on 1**

**Create a Spring Web Project using Maven** 

**A screenshot of a computer

Description automatically generated**

****

Project Structure:

**A screenshot of a computer

Description automatically generated**

**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**

country.xml

**A computer screen shot of a program

Description automatically generated**

country.java:

**package** com.cognizant.springlearn.model;

**public** **class** Country

{

**private** String code;

**private** String name;

**public** String getCode() {

**return** code;

}

**public** **void** setCode(String code) {

**this**.code = code;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**public** String toString() {

**return** "Country [code=" + code + ", name=" + name + "]";

}

}

App.java:

**A screenshot of a computer program

Description automatically generated**

**Project Structure:**

**A screenshot of a computer program

Description automatically generated**

SME to provide more detailing about the following aspects:

* bean tag - Defines a **Spring bean**, which is an object managed by the Spring container.
* id attribute - A **unique name** for the bean within the Spring container.
* class attribute - Fully-qualified name of the Java class that Spring should instantiate.
* property tag - Used to **set values** on the bean’s fields via setter injection.
* name attribute - Name of the property in the Java class
* value attribute - value that is passed to the setter method of the property.

ApplicationContext - This is the **central interface** to access Spring's IoC container.

It is used to:

Load beans

Manage their lifecycle

Inject dependencies

* ClassPathXmlApplicationContext - One of the implementations of ApplicationContext.
* What exactly happens when context.getBean() is invoked

**Looks up** the bean definition with ID "country" in the container.

**Checks the class** type (Country.class) for type safety.

**Creates an object** of that class

**Performs dependency injection** (e.g., calls setters like setCode("IN")).

**Returns the bean instance** to your Java code.