**Hands on 1**

**Spring Core – Implement data model for Employee using Spring XML Configuration**   
  
Implement data model for employee based on instructions provided below: 

* Create package com.cognizant.spring-learn.bean
* Create class com.cognizant.spring-learn.bean.Employee with following attributes

|  |  |
| --- | --- |
| **Type** | **Attribute Name** |
| int | id |
| String | name |
| double | salary |
| boolean | permanent |
| Date | dateOfBirth |

* Generate getters, setters and toString() methods.
* Include empty parameter constructor with debug log.
* Define values for the attributes in a new spring xml configuration file named employee.xml
* Display the employee details in a new method displayEmployee() in main method of SpringLearnApplication.java

Springlearn3Application.java:

**package** com.cognizant.springlearn.bean.springlearn3;

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

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

@SpringBootApplication

**public** **class** Springlearn3Application {

**public** **static** **void** displayEmployee()

{

ApplicationContext context = **new** ClassPathXmlApplicationContext("employee.xml");

Employee emp = context.getBean("employee", Employee.**class**);

System.***out***.println(emp.toString());

}

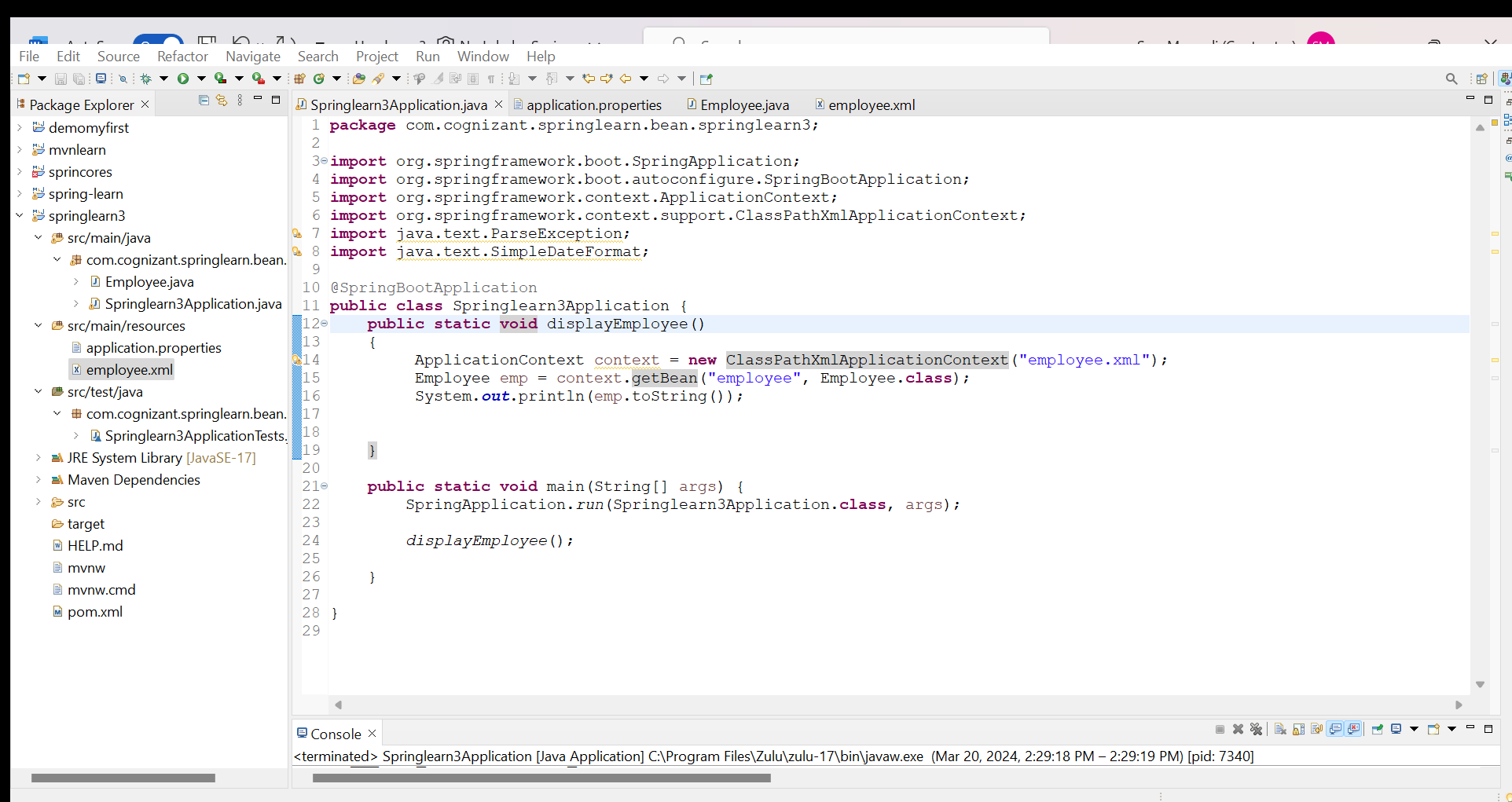
**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Springlearn3Application.**class**, args);

*displayEmployee*();

}

}



Employee.java:

**package** com.cognizant.springlearn.bean.springlearn3;

**import** java.util.Date;

**public** **class** Employee {

**private** **int** id;

**private** String name;

**private** **double** salary;

**private** **boolean** permanent;

**private** Date dateOfBirth;

**public** Employee() {

**super**();

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** **boolean** isPermanent() {

**return** permanent;

}

**public** **void** setPermanent(**boolean** permanent) {

**this**.permanent = permanent;

}

**public** Date getDateOfBirth() {

**return** dateOfBirth;

}

**public** **void** setDateOfBirth(Date dateOfBirth) {

**this**.dateOfBirth = dateOfBirth;

}

@Override

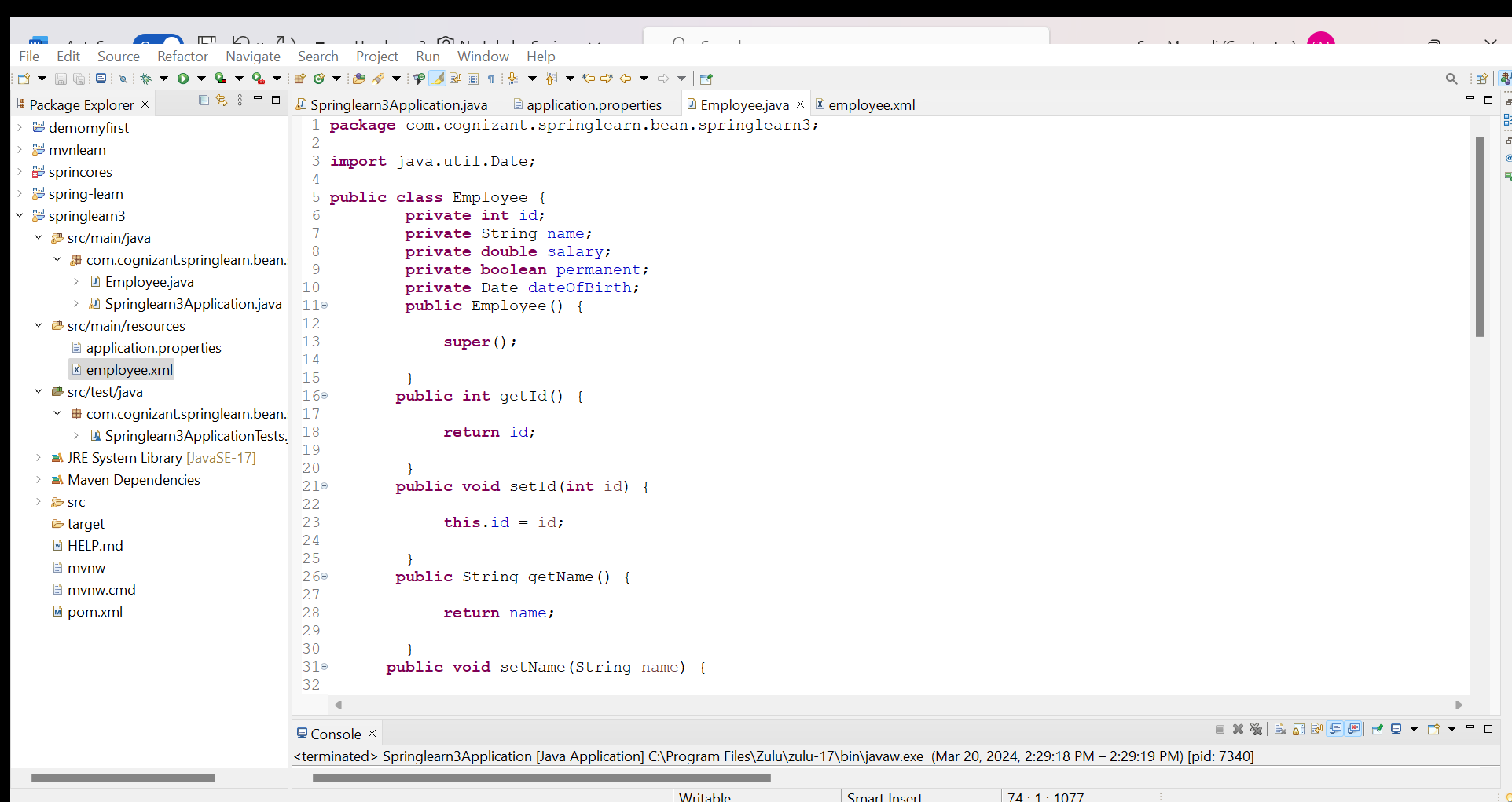
**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent

+ ", dateOfBirth=" + dateOfBirth + "]";

}

}



employee.xml:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*https://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"dateFormat"* class=*"java.text.SimpleDateFormat"*>

<constructor-arg value=*"yyyy-MM-dd"* />

</bean>

<bean id=*"employee"* class=*"com.cognizant.springlearn.bean.springlearn3.Employee"*>

<property name=*"id"* value=*"101"*/>

<property name=*"name"* value=*"John Doe"*/>

<property name=*"salary"* value=*"75000.0"*/>

<property name=*"permanent"* value=*"true"*/>

<property name=*"dateOfBirth"*>

<bean factory-bean=*"dateFormat"* factory-method=*"parse"*>

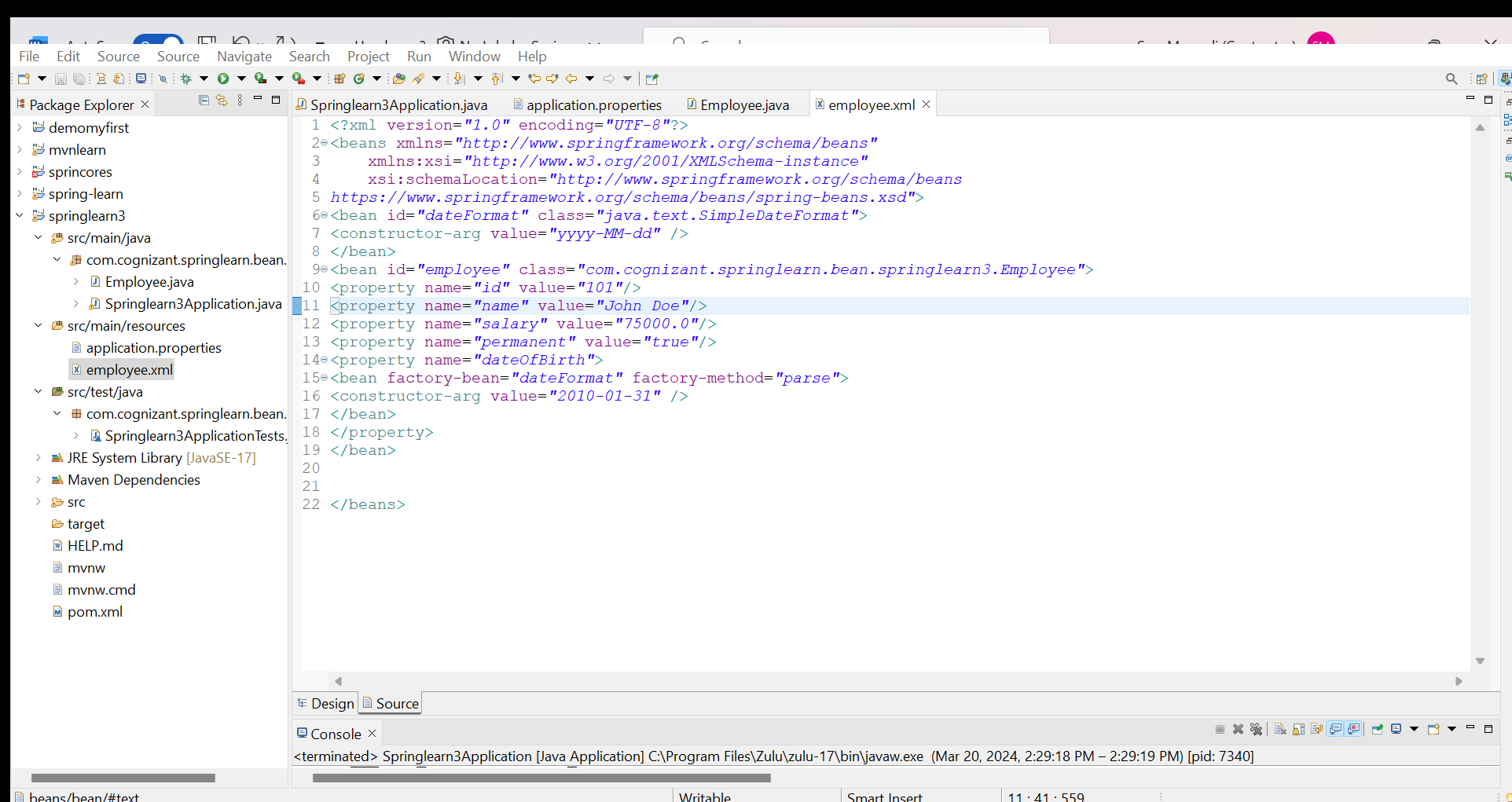
<constructor-arg value=*"2010-01-31"* />

</bean>

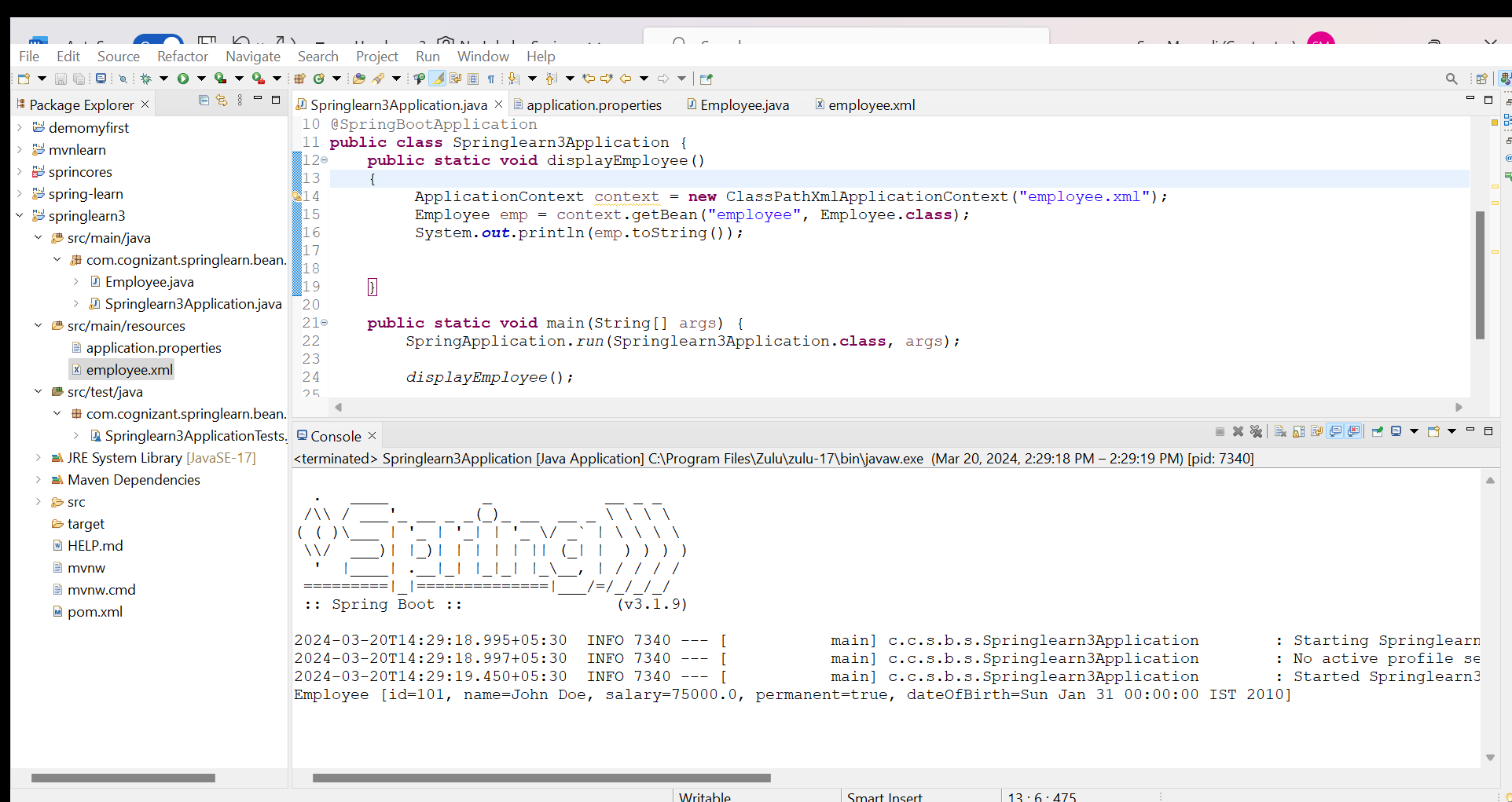
</property>

</bean>

</beans>



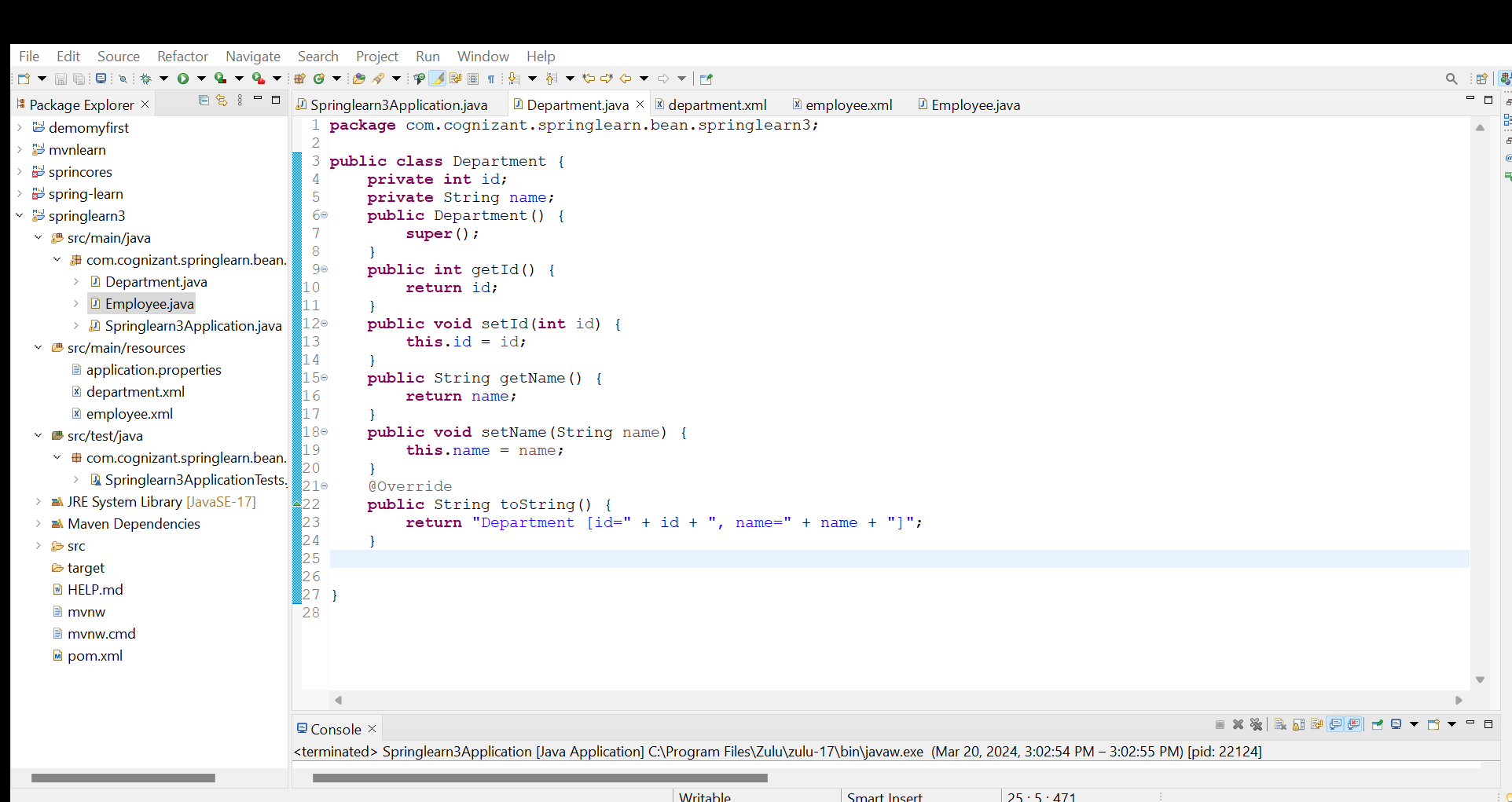
Output:



**Hands on 2**

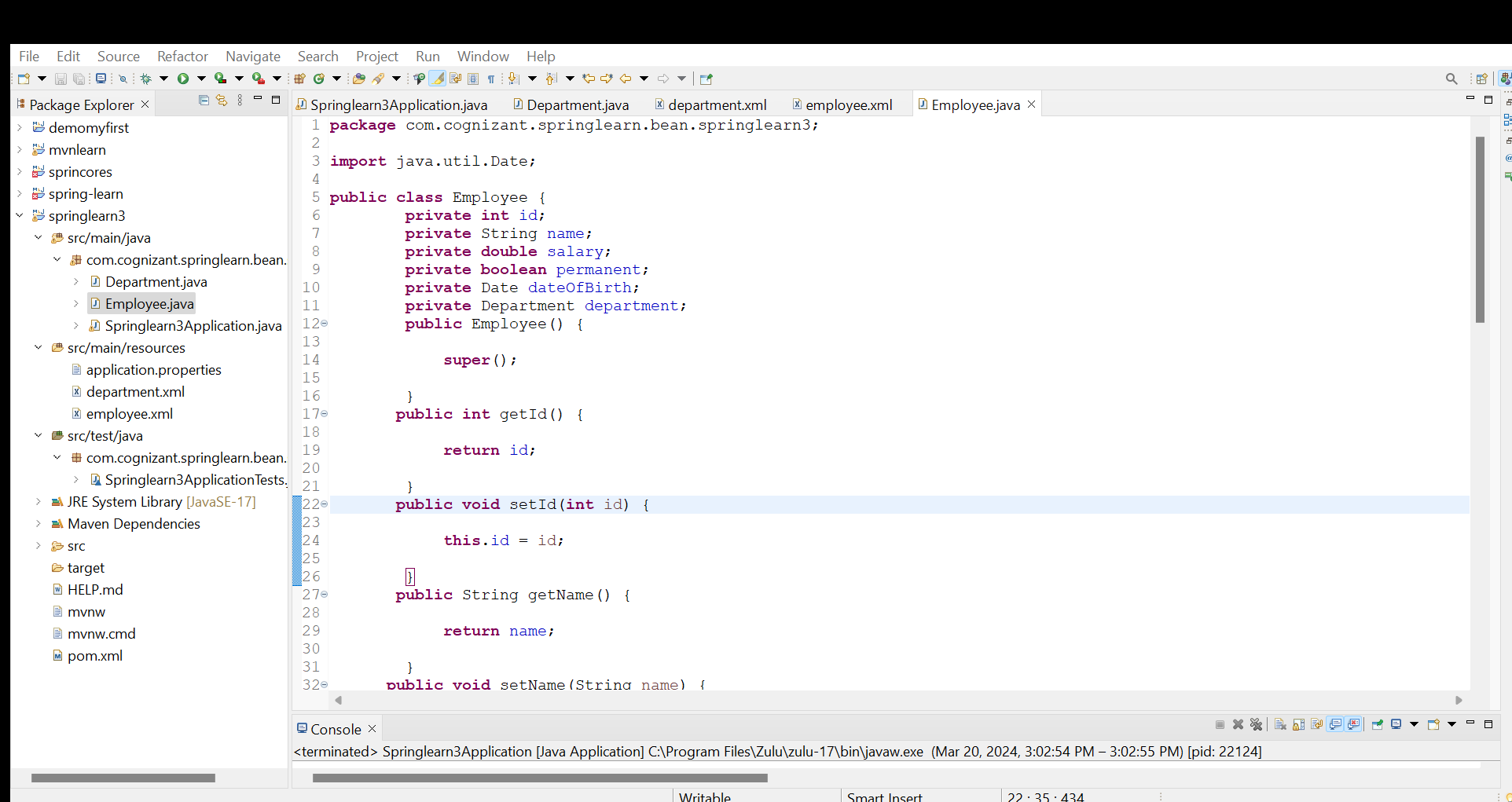
**Spring Core - Implement data model for Department in Employee**   
  
In continuation to the previous problem, let us add department details to Employee. Refer instructions below. 

* Create com.cognizant.spring-learn.bean.Department
* Add attributes id and name
* Include empty parameter constructor with log
* Generate getters, setters and toString()

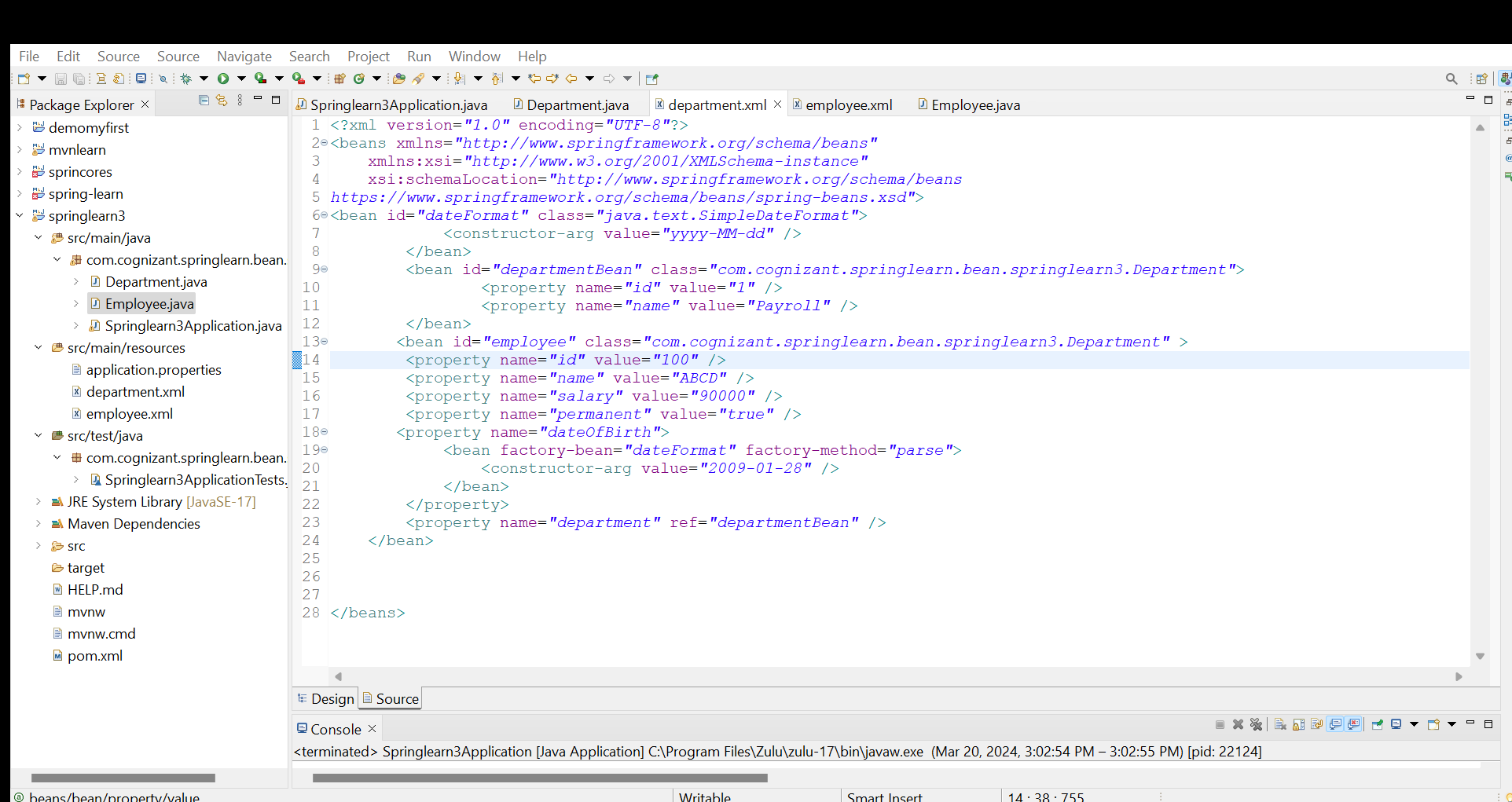


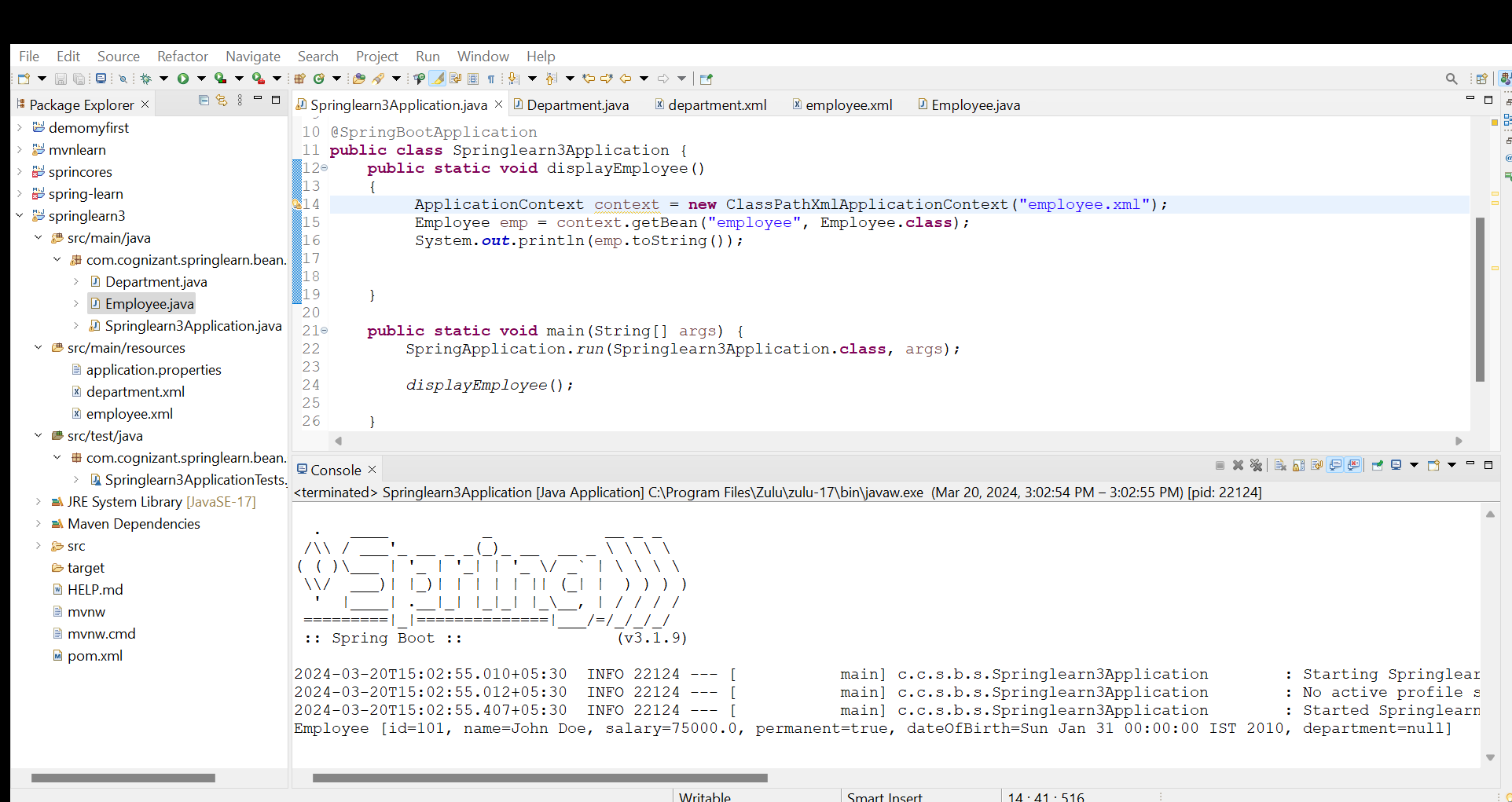
* Include department as an attribute in Employee class
* Include getters/setters for department
* Regenerate toString() method in Employee for inclusion of department
* In employee.xml, using ref attribute include department to employee.
* **package** com.cognizant.springlearn.bean.springlearn3;
* **import** java.util.Date;
* **public** **class** Employee {
* **private** **int** id;
* **private** String name;
* **private** **double** salary;
* **private** **boolean** permanent;
* **private** Date dateOfBirth;
* **private** Department department;
* **public** Employee() {
* **super**();
* }
* **public** **int** getId() {
* **return** id;
* }
* **public** **void** setId(**int** id) {
* **this**.id = id;
* }
* **public** String getName() {
* **return** name;
* }
* **public** **void** setName(String name) {
* **this**.name = name;
* }
* **public** **double** getSalary() {
* **return** salary;
* }
* **public** **void** setSalary(**double** salary) {
* **this**.salary = salary;
* }
* **public** **boolean** isPermanent() {
* **return** permanent;
* }
* **public** **void** setPermanent(**boolean** permanent) {
* **this**.permanent = permanent;
* }
* **public** Date getDateOfBirth() {
* **return** dateOfBirth;
* }
* **public** **void** setDateOfBirth(Date dateOfBirth) {
* **this**.dateOfBirth = dateOfBirth;
* }
* **public** Department getDepartment() {
* **return** department;
* }
* **public** **void** setDepartment(Department department) {
* **this**.department = department;
* }
* @Override
* **public** String toString() {
* **return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent
* + ", dateOfBirth=" + dateOfBirth + ", department=" + department + "]";
* }

* }



* Refer sample spring xml code below.





**Hands on 3**

**Spring Core - Include Skill details for Employee**   
  
In continuation to the previous problem, incorporate list of skills for each Employee instance:

* Create new class com.cognizant.spring-learn.bean.Skill
* Create instance variables id and name with data types int and String respectively.
* Include, empty parameter constructor with log, getters, setters and toString()
* Add skills as instance variable of type Skill array in Employee.
* In Employee class, include getters, setters and update the toString() method for inclusion of Skill[]
* Create multiple skill instances in employee.xml and add them to array of Skills.
* Set the array of Skills to skills of Employee bean in configuration xml file
* Load the employee class from employee.xml and display the data.
* Refer sample code below inclusion of skills property for employee in employee.xml

Skill.java:

**package** com.cognizant.springlearn.bean.springlearn3;

**public** **class** Skill {

**private** **int** id;

**private** String name;

@Override

**public** String toString() {

**return** "Skill [id=" + id + ", name=" + name + "]";

}

**public** Skill() {

System.***out***.println("Inside Skill Constructor");

}

**public** **int** getId() {

System.***out***.println("Getting id: " + id);

**return** id;

}

**public** **void** setId(**int** id) {

System.***out***.println("Setting id: " + id);

**this**.id = id;

}

**public** String getName() {

System.***out***.println("Getting name: " + name);

**return** name;

}

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

System.***out***.println("Setting name: " + name);

**this**.name = name;

}

}

Springlearn3Application.java:

**package** com.cognizant.springlearn.bean.springlearn3;

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

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

@SpringBootApplication

**public** **class** Springlearn3Application {

**public** **static** **void** displayEmployee()

{

ApplicationContext context = **new** ClassPathXmlApplicationContext("employee.xml");

Employee emp = context.getBean("employeeSkill",Employee.**class**);

System.***out***.println(emp.toString());

}

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

SpringApplication.*run*(Springlearn3Application.**class**, args);

*displayEmployee*();

}

}

Employee.java:

**package** com.cognizant.springlearn.bean.springlearn3;

**import** java.util.Arrays;

**import** java.util.Date;

**public** **class** Employee {

**private** **int** id;

**private** String name;

**private** **double** salary;

**private** **boolean** permanent;

**private** Date dateOfBirth;

**private** Department department;

**private** Skill[] skills;

**public** Employee() {

**super**();

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** **boolean** isPermanent() {

**return** permanent;

}

**public** **void** setPermanent(**boolean** permanent) {

**this**.permanent = permanent;

}

**public** Date getDateOfBirth() {

**return** dateOfBirth;

}

**public** **void** setDateOfBirth(Date dateOfBirth) {

**this**.dateOfBirth = dateOfBirth;

}

**public** Department getDepartment() {

**return** department;

}

**public** **void** setDepartment(Department department) {

**this**.department = department;

}

**public** Skill[] getSkills() {

**return** skills;

}

**public** **void** setSkills(Skill[] skills) {

**this**.skills = skills;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent

+ ", dateOfBirth=" + dateOfBirth + ", department=" + department + ", skills=" + Arrays.*toString*(skills)

+ "]";

}

}

Employee.xml:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*https://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"skillHistory"* class=*"com.cognizant.springlearn.bean.springlearn3.Skill"*>

<property name=*"id"* value=*"10"*/>

<property name=*"name"* value=*"History"*/>

</bean>

<bean id=*"skillGeography"* class=*"com.cognizant.springlearn.bean.springlearn3.Skill"*>

<property name=*"id"* value=*"20"*/>

<property name=*"name"* value=*"Geography"*/>

</bean>

<bean id=*"skillEcomics"* class=*"com.cognizant.springlearn.bean.springlearn3.Skill"*>

<property name=*"id"* value=*"3"*/>

<property name=*"name"* value=*"Economics"*/>

</bean>

<bean id=*"employeeSkill"* class=*"com.cognizant.springlearn.bean.springlearn3.Employee"*>

<property name=*"skills"*>

<array>

<ref bean=*"skillHistory"* />

<ref bean=*"skillGeography"* />

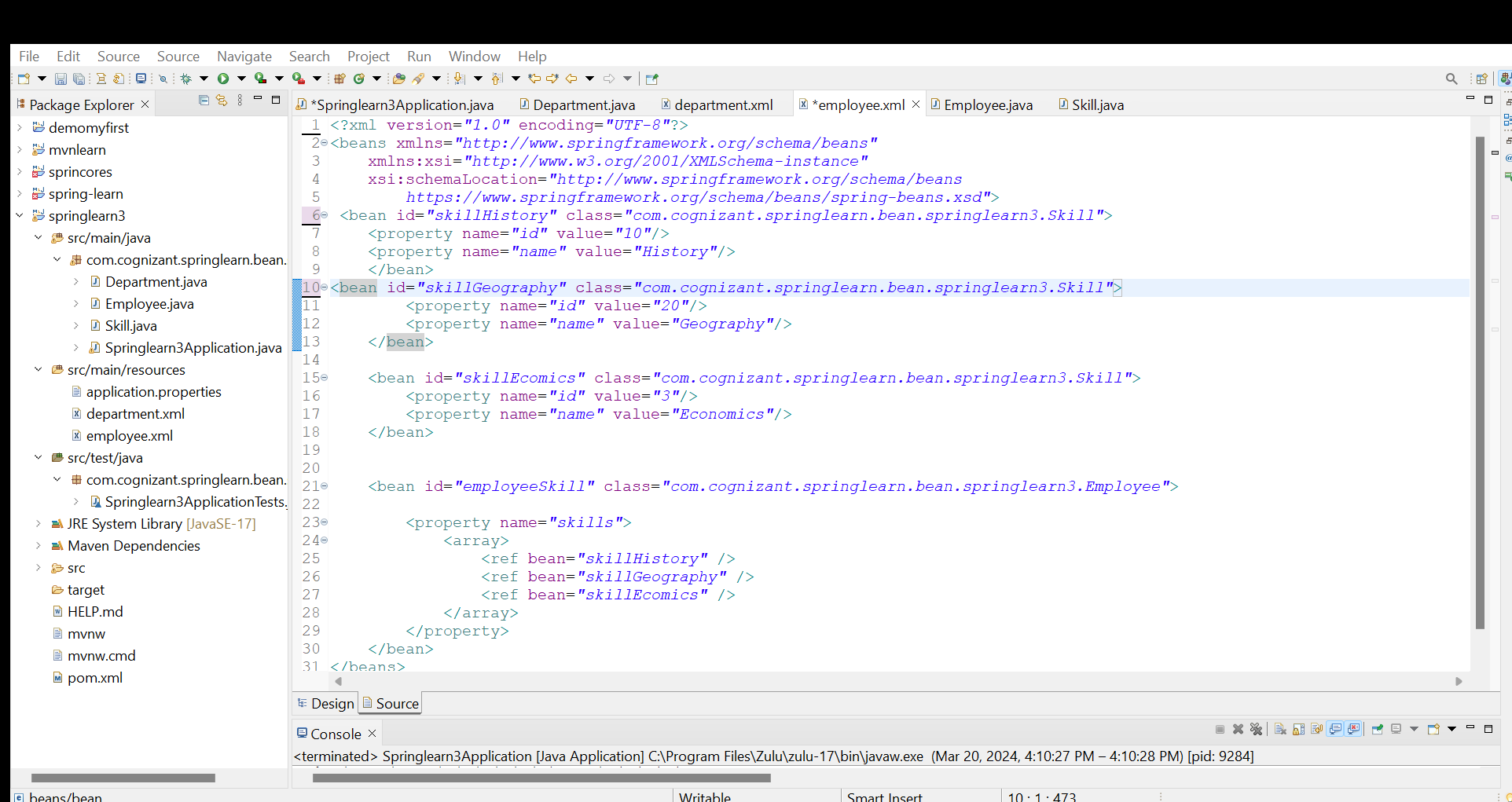
<ref bean=*"skillEcomics"* />

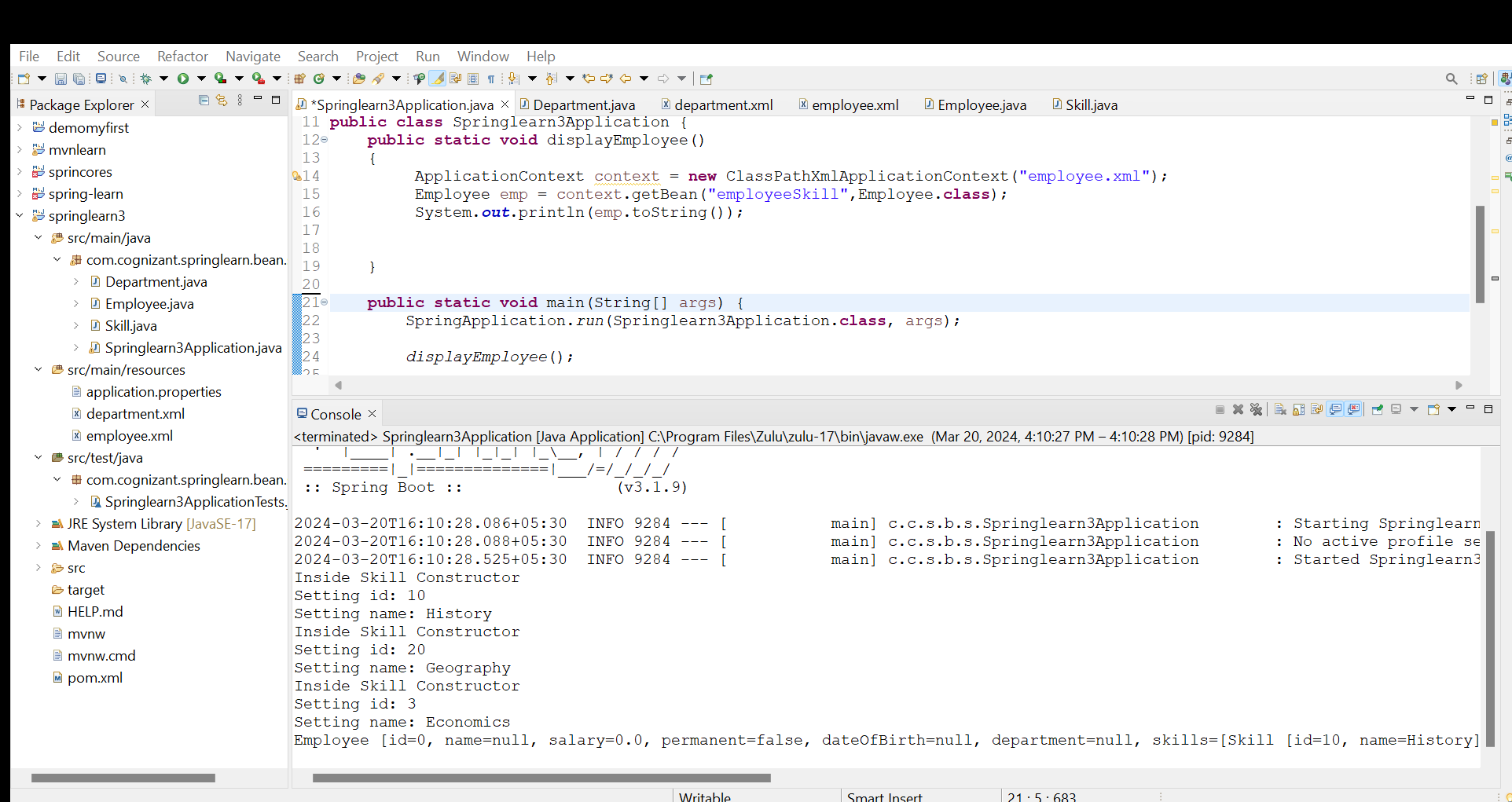
</array>

</property>

</bean>

</beans>





**Hands on 4**

**Spring Core - Inversion of Control / Dependency Injection**   
  
In order to build a web service for managing employee details, controller, service and dao classes needs to be defined.  
  
As part of this hands on, creation of these classes needs to be done using Spring Core framework's Inversion of Control (IoC) and Dependency Injection.  
  
At the same time we need to implement this using annotations without spring configuraton xml file. Refer steps below to gradually implement this. 

* Create all necessary classes based on the below specified class diagram. Do not include any constructor, toString() and getter methods.
* Include logs in constructors and setter methods on all these classes.
* Implement dependency injection in a staged manner as specified below one by one:
  + **Spring XML Configuration:**
    - In employee.xml define the beans for EmployeeController, EmployeeService and EmployeeDao using bean reference.
    - Load the EmployeeController bean in a new method getEmployeeController() in the SpringLearnApplication.java
    - Check the logs to see if EmployeeController, EmployeeService and EmployeeDao are loaded
  + **Spring XML Configuration with Autowiring:**
    - Include new bean configuration in employee.xml for the three classes using autowire property.
    - Modify getEmployeeController() method to load the new autowired bean from xml configuration.
    - Check the logs to see if all three classes are loaded.
  + **Loading beans using Annotations:**
    - Include @Component annotation at class level for EmployeeController, EmployeeService and EmployeeDao
    - Include @Autowired annotation in the respective setter method of EmployeeController and EmployeeService
    - Include a new method in SpringLearnApplication.java with below signature:

public static void displayEmployeeControllerAnnotation(ApplicationContext applicationContext)

* Use the below code in this method to load the controller bean:

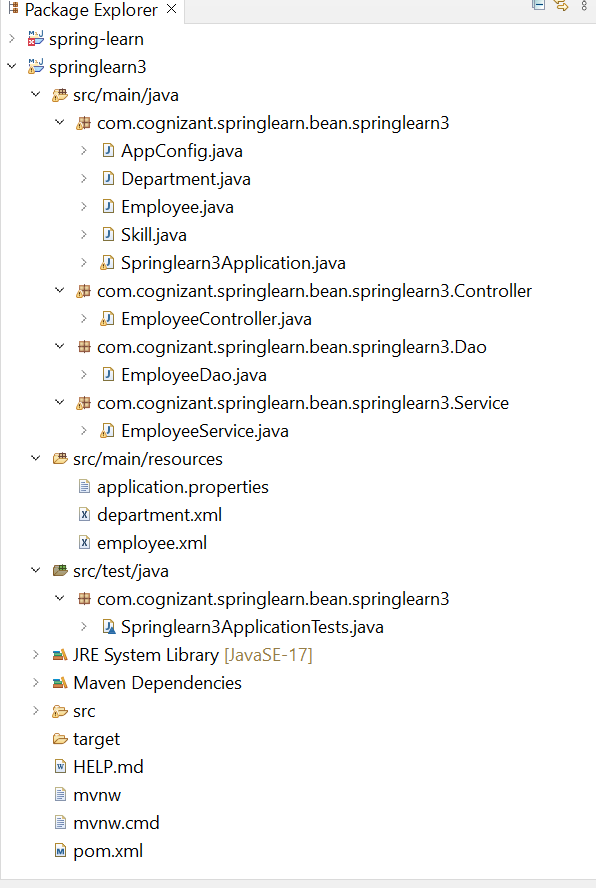
        EmployeeController employeeController = (EmployeeController) applicationContext

                .getBean("employeeController");

* Invoke displayEmployeeControllerAnnotation() method in main method passing the applicaitonContext.

|  |
| --- |
| https://cognizant.e-box.co.in/uploads/Image/01fseangular/ioc1.png |

SME to explain how Inversion of Control works.



EmployeeController.java:

**package** com.cognizant.springlearn.bean.springlearn3.Controller;

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

**import** org.springframework.stereotype.Controller;

**import** com.cognizant.springlearn.bean.springlearn3.Service.EmployeeService;

@Controller

**public** **class** EmployeeController {

**private** EmployeeService employeeService;

@Autowired

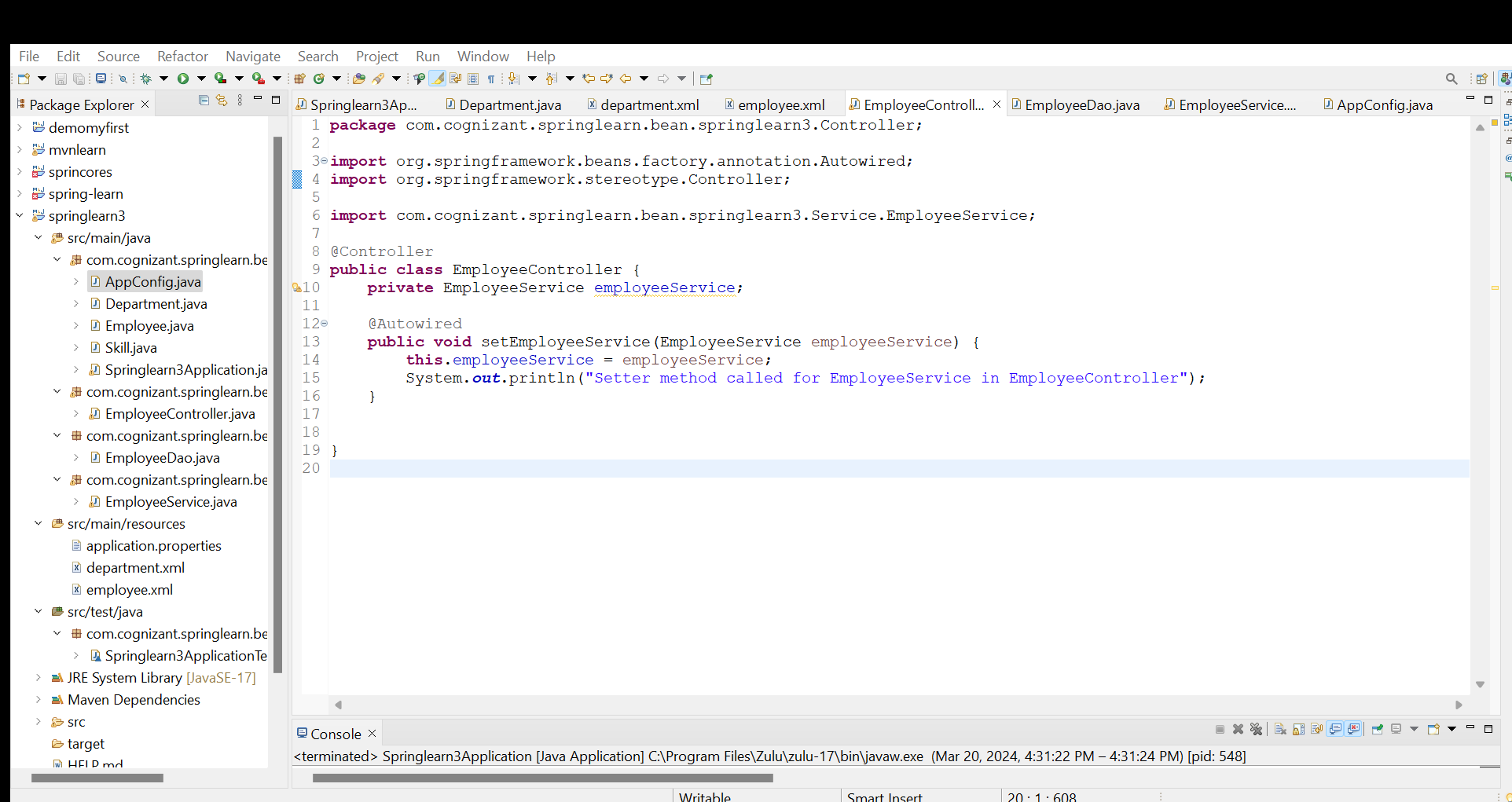
**public** **void** setEmployeeService(EmployeeService employeeService) {

**this**.employeeService = employeeService;

System.***out***.println("Setter method called for EmployeeService in EmployeeController");

}

}



EmployeeDao.java:

**package** com.cognizant.springlearn.bean.springlearn3.Dao;

**import** org.springframework.stereotype.Component;

@Component

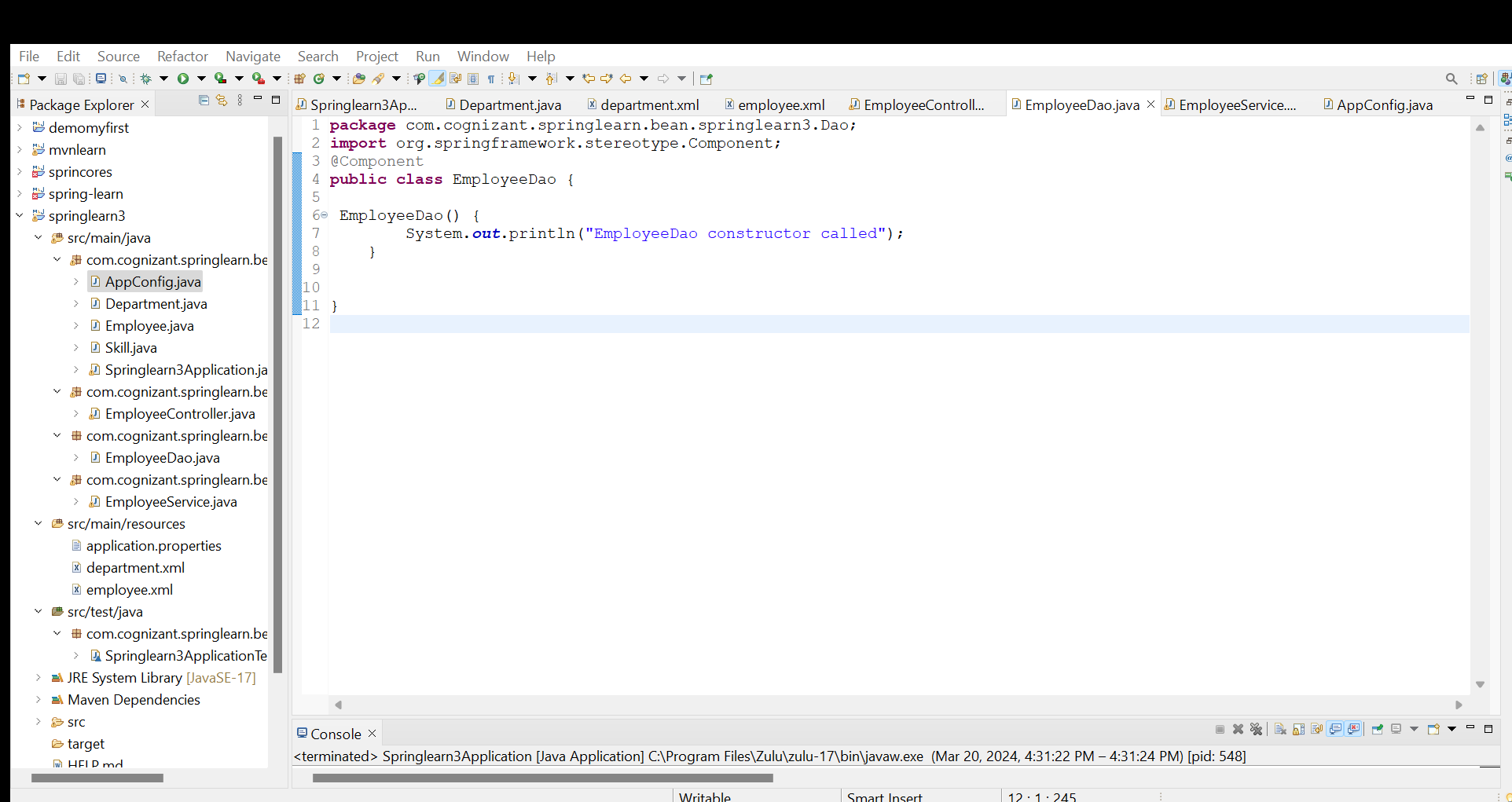
**public** **class** EmployeeDao {

EmployeeDao() {

System.***out***.println("EmployeeDao constructor called");

}

}



EmployeeService.java:

**package** com.cognizant.springlearn.bean.springlearn3.Service;

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

**import** org.springframework.stereotype.Component;

**import** com.cognizant.springlearn.bean.springlearn3.Dao.EmployeeDao;

@Component

**public** **class** EmployeeService {

**private** EmployeeDao employeeDao;

@Autowired

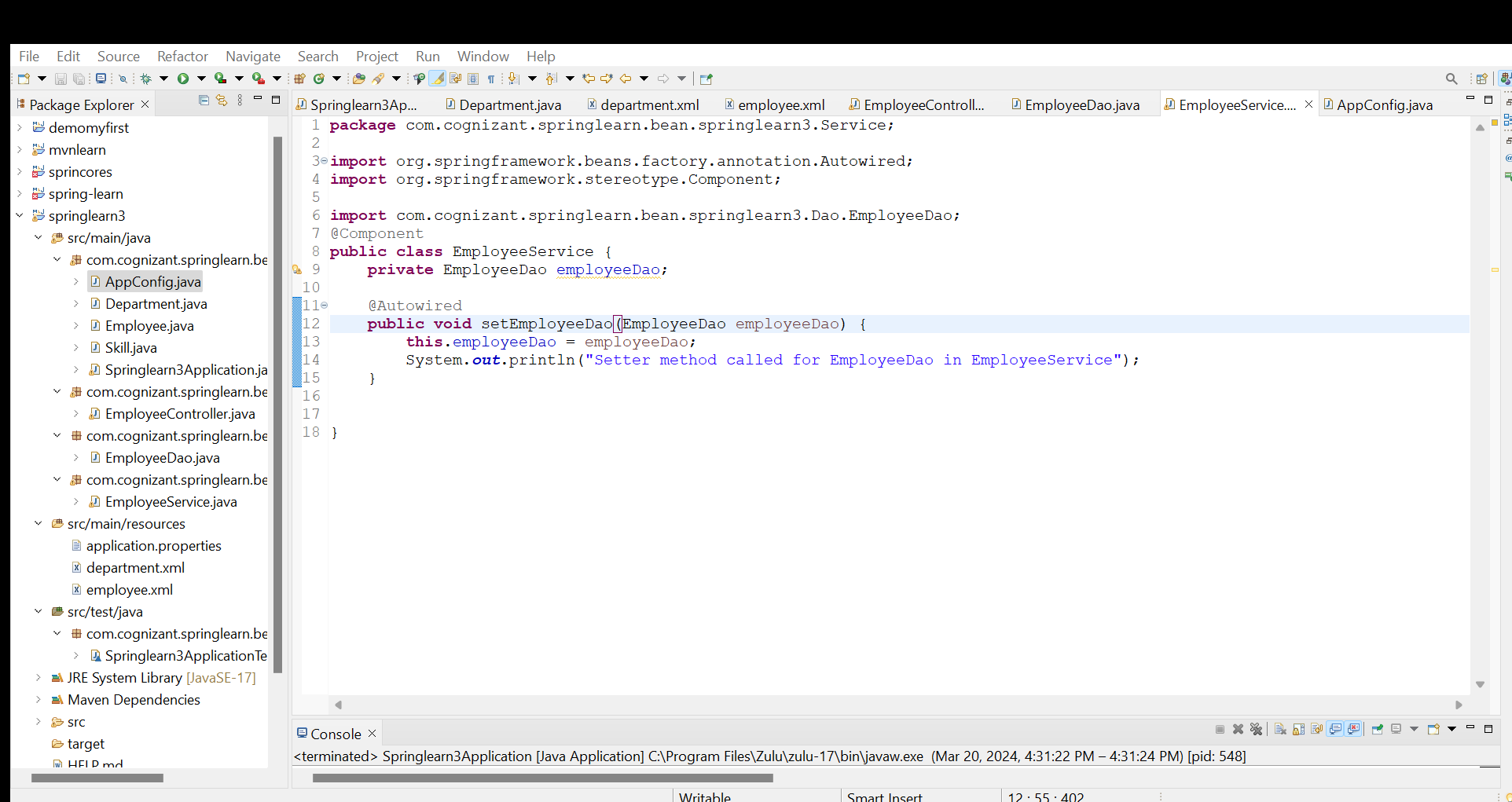
**public** **void** setEmployeeDao(EmployeeDao employeeDao) {

**this**.employeeDao = employeeDao;

System.***out***.println("Setter method called for EmployeeDao in EmployeeService");

}

}



AppConfig.java:

**package** com.cognizant.springlearn.bean.springlearn3;

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

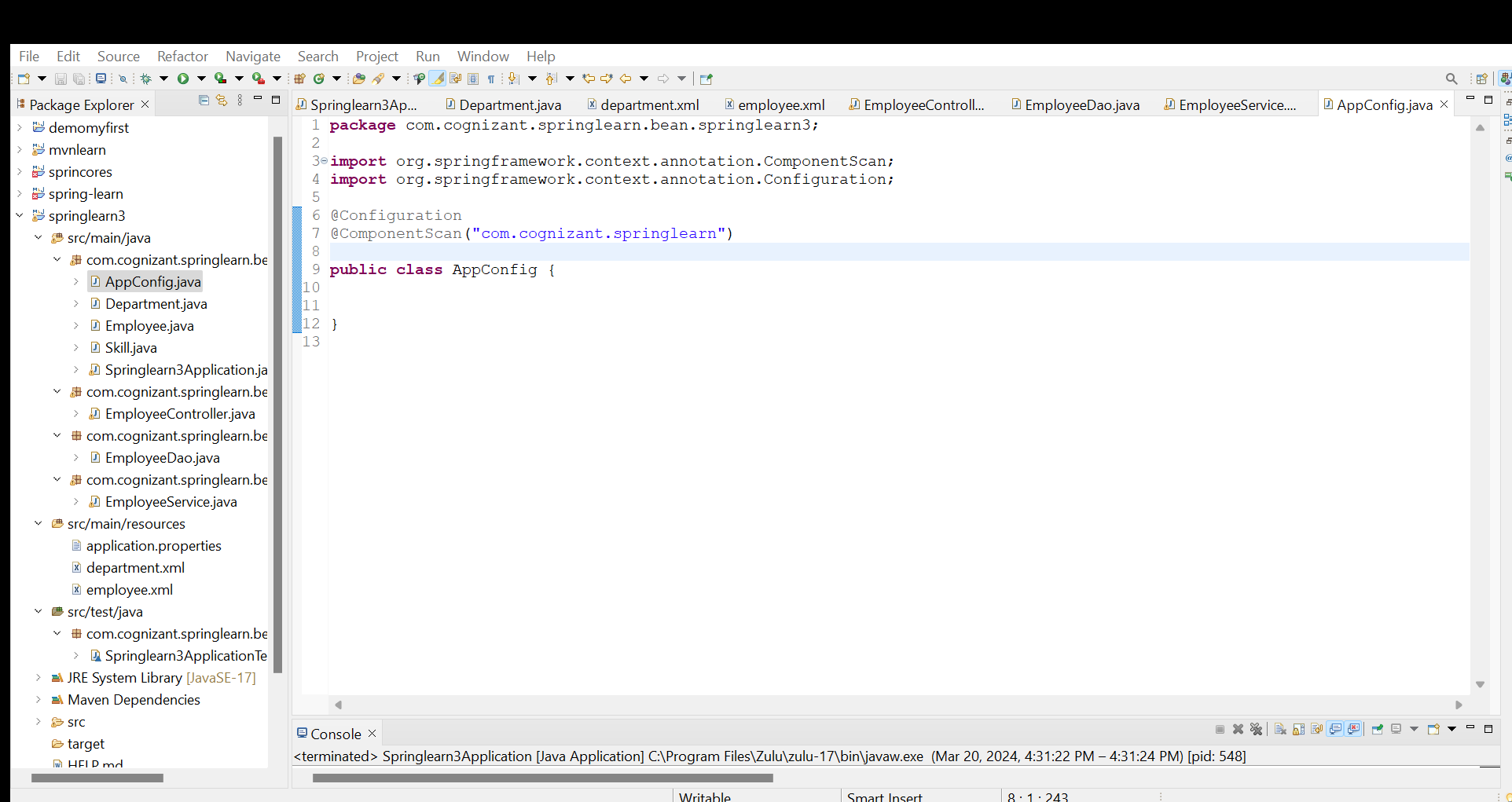
**import** org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan("com.cognizant.springlearn")

**public** **class** AppConfig {

}



Springlearn3Application.java:

**package** com.cognizant.springlearn.bean.springlearn3;

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

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.cognizant.springlearn.bean.springlearn3.Controller.EmployeeController;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

@SpringBootApplication

**public** **class** Springlearn3Application {

**public** **static** **void** displayEmployee()

{

ApplicationContext context = **new** ClassPathXmlApplicationContext("employee.xml");

Employee emp = context.getBean("employeeSkill",Employee.**class**);

System.***out***.println(emp.toString());

}

**public** **static** **void** displayEmployeeControllerAnnotation(ApplicationContext applicationContext) {

EmployeeController employeeController = (EmployeeController) applicationContext.getBean("employeeController");

System.***out***.println("Loaded EmployeeController bean: " + employeeController);

}

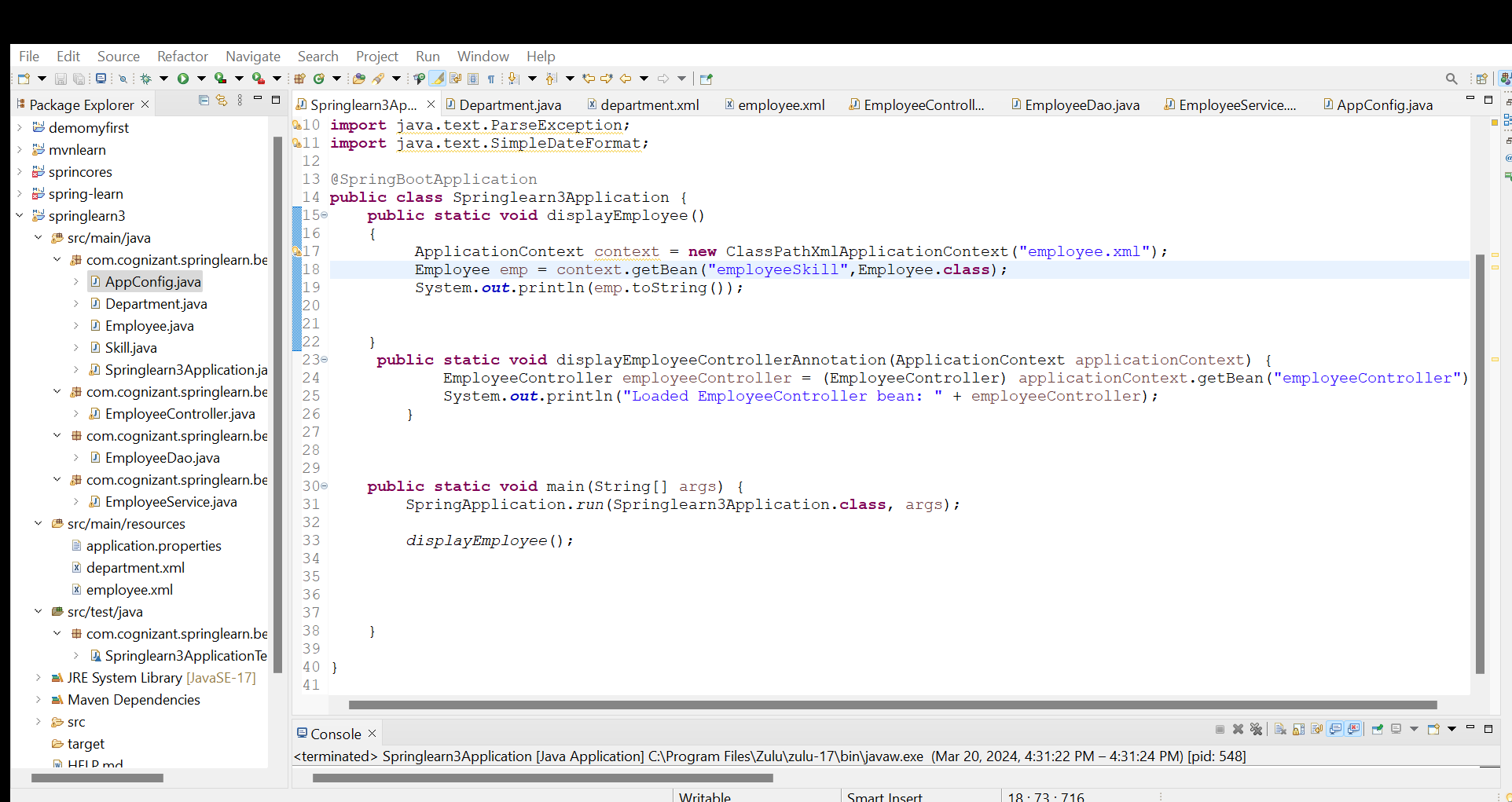
**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Springlearn3Application.**class**, args);

*displayEmployee*();

}

}



Output:

