

JFSD-JAVA FULL STACK DEVELOPMENT

LAB05

ID:2000030185

Name:Suprith Ch

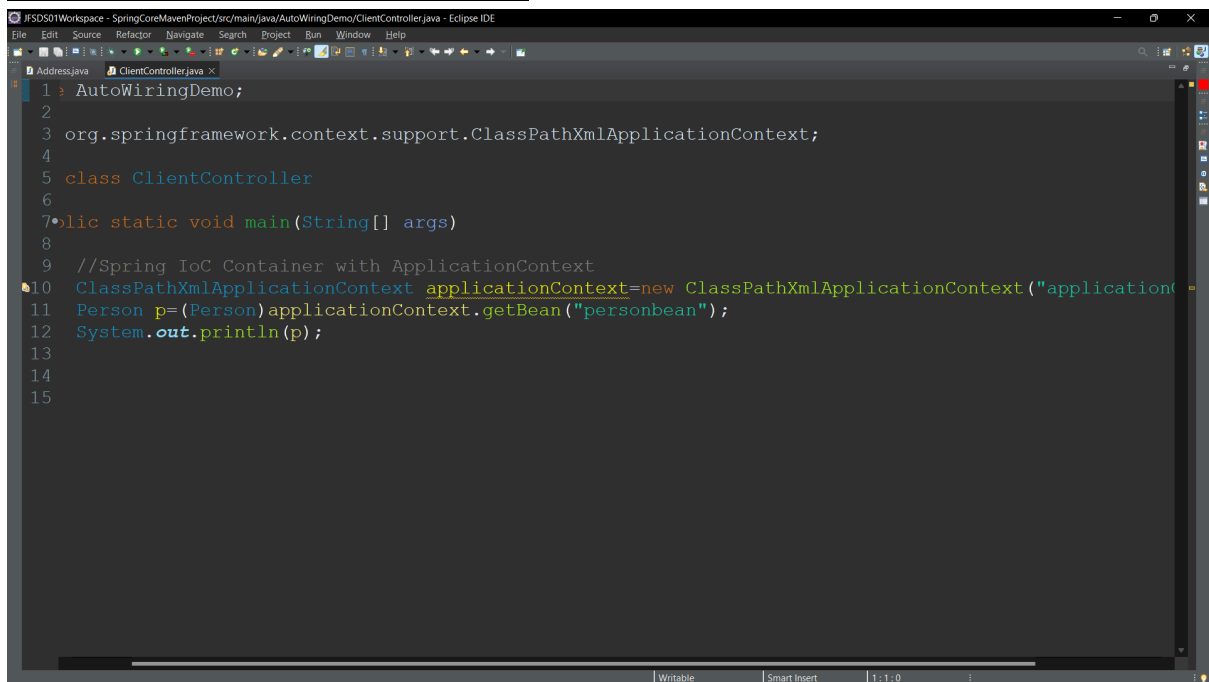
AutoWiringDemo:

Address.java Class:

```
1 package AutoWiringDemo;
2
3 public class Address
4 {
5     private String landmark;//non primitive data type
6     private String city;//non primitive data type
7     private String state;//non primitive data type
8     private String country;//non primitive data type
9     @Override
10    public String toString() {
11        return "Address [landmark=" + landmark + ", city=" + city + ", state=" + state + ", country
12    }
13    public String getLandmark() {
14        return landmark;
15    }
16    public void setLandmark(String landmark) {
17        this.landmark = landmark;
18    }
19    public String getCity() {
20        return city;
21    }
22    public void setCity(String city) {
23        this.city = city;
24    }
25    public String getState() {
26        return state;
```

```
13    public String getLandmark() {
14        return landmark;
15    }
16    public void setLandmark(String landmark) {
17        this.landmark = landmark;
18    }
19    public String getCity() {
20        return city;
21    }
22    public void setCity(String city) {
23        this.city = city;
24    }
25    public String getState() {
26        return state;
27    }
28    public void setState(String state) {
29        this.state = state;
30    }
31    public String getCountry() {
32        return country;
33    }
34    public void setCountry(String country) {
35        this.country = country;
36    }
37 }
38
```

ClientController.java Class:

A screenshot of the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. The main editor window displays the code for ClientController.java. The code starts with a package declaration 'AutoWiringDemo;' followed by an import statement for 'org.springframework.context.support.ClassPathXmlApplicationContext;'. Then, a class 'ClientController' is declared. Inside the class, a static void method 'main' is defined, which takes a 'String[] args' parameter. The main method contains a comment '//Spring IoC Container with ApplicationContext', followed by the creation of a 'ClassPathXmlApplicationContext' object named 'applicationContext'. Then, a 'Person' object 'p' is retrieved from the context using 'getBean("personbean")', and its details are printed to the console using 'System.out.println(p)'. The IDE's status bar at the bottom shows 'Writable', 'Smart Insert', and '1: 1: 0'.

Pojo Class:

Person.java

```
1 package AutoWiringDemo;
2
3 public class Person
4 {
5     private int id; // primitive data type
6     private String name; // non primitive data type
7     private String gender; // non primitive data type
8     private double age; // primitive data type
9     private Address address; // non primitive data type
10    @Override
11    public String toString() {
12        return "Person [id=" + id + ", name=" + name + ", gender=" + gender + ", age=" + age + ", a"
13            + " ]";
14    }
15    public int getId() {
16        return id;
17    }
18    public void setId(int id) {
19        this.id = id;
20    }
21    public String getName() {
22        return name;
23    }
24    public void setName(String name) {
25        this.name = name;
26    }
27
28    public String getGender() {
29        return gender;
30    }
31    public void setGender(String gender) {
32        this.gender = gender;
33    }
34    public double getAge() {
35        return age;
36    }
37    public void setAge(double age) {
38        this.age = age;
39    }
40    public Address getAddress() {
41        return address;
42    }
43    public void setAddress(Address address) {
44        this.address = address;
45    }
46 }
47
```

ApplicationContext.xml:

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: SpringCoreMavenProject, src/main/java (containing AnnotationsDemo, AutoWiringDemo, Address.java, ClientController.java, Person.java, ConstructorDI, SetterDI), src/test/java, and src/main/resources (containing applicationContext.xml). The main editor shows the applicationContext.xml file with the following XML content:

```
36 <value>SERVICENOW</value>
37 </list>
38 </constructor-arg>
39 </bean>
40 <bean id="personbean" class="AutoWiringDemo.Person">
41 <property name="id" value="1"></property>
42 <property name="name" value="Suprith"></property>
43 <property name="gender" value="MALE"></property>
44 <property name="age" value="20"></property>
45 <property name="address" ref="addressbean"></property>
46 </bean>
47 <bean id="addressbean" class="AutoWiringDemo.Address">
48 <property name="landmark" value="greenfields"></property>
49 <property name="city" value="GUNTUR"></property>
50 <property name="state" value="AP"></property>
51 <property name="country" value="India"></property>
52 </bean>
53 </beans>
```

The bottom of the IDE shows the Problems, Console, Javadoc, Declaration, and Properties tabs. The Console tab is active, displaying "No consoles to display at this time."

Output:

Eclipse IDE screenshot showing the console output of a Java application. The output displays a Person object with the following details: `Person [id=1, name=Suprith, gender=MALE, age=20.0, address=Address [landmark=greenfields, city=GUNTUR,`

Eclipse IDE screenshot showing the source code of the `ClientController.java` file. The code is as follows:

```
1 package AutoWiringDemo;
2
3 import org.springframework.context.support.ClassPathXmlApplicationContext;
4
5 public class ClientController
```

The console output below the code shows the full object representation: `ress=Address [landmark=greenfields, city=GUNTUR, state=AP, country=India]]`