07 - Constructor Injection

Constructor Injection is used in Spring when we want to inject values or dependencies into a bean during its creation.

- It is useful when the values or objects must be provided during object initialization.
- In Constructor Injection, values and references are passed to the bean through its constructor.
- Spring offers the **<constructor-arg>** tag to handle this injection.
- <constructor-arg> Tag includes:
    - **value**: Used to pass primitive values.
    - **ref**: Used to pass references to other beans.
- The order and type of arguments must match the constructor's parameters.

Example:

Alien.java

The class takes an age (primitive) and a Laptop object as parameters in its constructor.

```
public class Alien {

	private int age;

	//private Laptop lap = new Laptop();

	private Laptop lap;

	private int salary;


	public Alien() {

		System.out.println("Object Created");

	}
```

```
@ConstructorProperties({"age","lap"})
	public Alien(int age,Laptop lap) {
		System.out.println("Para Constructor Called");
		this.age = age;
		this.lap = lap;
	}
	public void code() {
		System.out.println("Coding");
		lap.compile();
	}
}
```

Laptop.java

```
public class Laptop {
	public Laptop() {
		System.out.println("Laptop object created");
	}
	public void compile() {
		System.out.println("Compiling");
	} }
```

Spring.xml

```
<bean id="alien1" class="com.telusko.Alien" >

        <constructor-arg value="21"></constructor-arg>

        <constructor-arg ref="lap1"></constructor-arg>

</bean>

<bean id="lap1" class="com.telusko.Laptop">

</bean>
```

Here,

- value="21": Injects the age value (primitive) into the constructor.
- ref="lap1": Injects the reference to the Laptop bean.

App.java

```
public class App {

   public static void main( String[] args ) {

        ApplicationContext context = new
ClassPathXmlApplicationContext("spring.xml");

        Alien obj1 = (Alien) context.getBean("alien1");

        System.out.println(obj1.getAge());

        obj1.code();

   }

}
```

Output:

```
Laptop object created
Para Constructor Called
Laptop object created
21
Coding
Compiling
```

👉 **Handling Argument Types and Indexes:**

- **Type Attribute**: When arguments are of different types, the type attribute can be used to specify the exact type of the argument.

```
<constructor-arg type="int" value="21"/>

<constructor-arg type="com.telusko.Laptop" ref="lap1"/>
```

- **Index Attribute**: In some cases, the constructor parameters may be of the same type. The index attribute specifies the 0-based index of the argument.

```
<constructor-arg index="0" value="21"/>

<constructor-arg index="1" ref="lap1"/>
```

- **Name Attribute**: If the constructor parameters have names, the name attribute can be used. It must still follow the order of the parameters unless the @ConstructorProperties annotation is used.

```
<constructor-arg name="age" value="21"/>

<constructor-arg name="lap" ref="lap1"/>
```

👉 Using @ConstructorProperties Annotation

- ➢ If the parameters are not provided in sequence using the name attribute, you can use the @ConstructorProperties annotation to specify the exact names of the constructor arguments.
- ➢ This annotation helps Spring map the arguments to the constructor parameters by name, even if the values are provided in a different order in the XML file.

```
@ConstructorProperties({"age", "lap"})
public Alien(int age, Laptop lap) {
        this.age = age;
        this.lap = lap;
}
```

Code Link:
https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/2%20Exploring%20Spring%20Framework/2.7%20Constructor%20Injection/Spring1