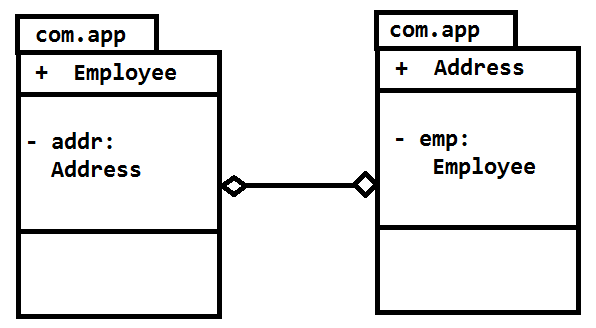
Spring Circular Dependency:

If two classes are dependent on each other, then creating object will be very difficult. In this case spring provides best solution.

First it creates objects using default constructor then it performs injections.

Example:

Consider Employee and Address are the two classes which are having dependency on each other.



Java code is:

**package** com.app;

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

**private** Address addr;

**public** Employee() {

**super**();

System.*out*.println("In Employee Default Constructor");

}

**public** Address getAddr() {

**return** addr;

}

**public** **void** setAddr(Address addr) {

**this**.addr = addr;

System.*out*.println("In Employee class , Address setter");

}

}

**package** com.app;

**public** **class** Address {

**private** Employee emp;

**public** Address() {

**super**();

System.*out*.println("In Address Default Constructor");

}

**public** Employee getEmp() {

**return** emp;

}

**public** **void** setEmp(Employee emp) {

**this**.emp = emp;

System.*out*.println("In Address class , Employee setter");

}

}

Config.xml

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

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

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

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

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

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<bean class=*"com.app.Employee"* name=*"empObj"*>

<property name=*"addr"*>

<ref bean=*"addrObj"*/>

</property>

</bean>

<bean class=*"com.app.Address"* name=*"addrObj"*>

<property name=*"emp"*>

<ref bean=*"empObj"*/>

</property>

</bean>

</beans>

**package** com.app;

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

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

**import** com.app.Employee;

**public** **class** Test {

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

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

Employee cons =context.getBean("empObj", Employee.**class**);

System.*out*.println(cons);

}

}

Output:

In Employee Default Constructor

In Address Default Constructor

In Address class , Employee setter

In Employee class , Address setter

[com.app.Employee@a761fe](mailto:com.app.Employee@a761fe)

Step by Step Process

