1)

Address.java file

**package** com.helloworldexample;

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

**private** String street;

**private** String city;

**private** String state;

**private** **int** zip;

**private** String country;

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** **int** getZip() {

**return** zip;

}

**public** **void** setZip(**int** zip) {

**this**.zip = zip;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

@Override

**public** String toString() {

**return** "Address [street=" + street + ", city=" + city + ", state=" + state + ", zip=" + zip + ", country="

+ country + "]";

}

}

**package** com.helloworldexample;

**public** **class** Customer {

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

**private** String name;

**private** **long** contact;

**private** Address address;

**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** **long** getContact() {

**return** contact;

}

**public** **void** setContact(**long** contact) {

**this**.contact = contact;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

**void** display() {

System.***out***.println(id+ " " +name+ " " +contact+ " " +address);

}

}

Spring.xml file

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id = *"customer\_address"* class= *"com.helloworldexample.Address"*>

<property name=*"city"* value=*"kandlakoya"*></property>

<property name=*"state"* value=*"telangana"*></property>

<property name=*"zip"* value=*"505504"*></property>

<property name=*"country"* value=*"india"*></property>

</bean>

<bean id = *"customer"* class= *"com.helloworldexample.Customer"*>

<property name=*"id"* value=*"1001"*></property>

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

<property name=*"contact"* value=*"9959173422"*></property>

<property name=*"address"* ref=*"customer\_address"*></property>

</bean>

</beans>

Test.java file

**package** com.helloworldexample;

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

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

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

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

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

Customer customer=(Customer)context.getBean("customer");

customer.display();

}

}

OUTPUT:

1001 rama 9959173422 Address [street=null, city=kandlakoya, state=telangana, zip=505504, country=india]

2)

A)

Question.java file

**import** java.util.Iterator;

**import** java.util.List;

**public** **class** Question {

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

**private** String question;

**private** List<String> answers;

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

**return** id;

}

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

**this**.id = id;

}

**public** String getQuestion() {

**return** question;

}

**public** **void** setQuestion(String question) {

**this**.question = question;

}

**public** List<String> getAnswers() {

**return** answers;

}

**public** **void** setAnswers(List<String> answers) {

**this**.answers = answers;

}

**void** display() {

System.***out***.println(id+ " " +question);

System.***out***.println("answers:");

Iterator<String> list = answers.iterator();

**while**(list.hasNext()) {

System.***out***.println(list.next());

}

}

}

Spring.xml file

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id=*"questions"* class=*"com.helloworldexample.Question"*>

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

<property name=*"question"* value=*"what is spring core"*></property>

<property name=*"answers"*>

<list>

<value>It enables to build applications</value>

<value>by this we can have singleton classes that is only one object</value>

<value>have the bean factory to create beanss</value>

</list>

</property>

</bean>

</beans>

TestClass.java file

**package** com.helloworldexample;

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

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

**public** **class** TestClass {

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

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

Question questions=(Question)context.getBean("questions");

questions.display();

}

}

OUTPUT:

101 what is spring core

answers:

It enables to build applications

by this we can have singleton classes that is only one object

have the bean factory to create beans

B)

Question.java file

**package** com.helloworldexample;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Set;

**public** **class** Question {

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

**private** String question;

**private** Set<String> answers;

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

**return** id;

}

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

**this**.id = id;

}

**public** String getQuestion() {

**return** question;

}

**public** **void** setQuestion(String question) {

**this**.question = question;

}

**public** Set<String> getAnswers() {

**return** answers;

}

**public** **void** setAnswers(Set<String> answers) {

**this**.answers = answers;

}

**void** display() {

System.***out***.println(id+ " " +question);

System.***out***.println("answers:");

Iterator<String> list = answers.iterator();

**while**(list.hasNext()) {

System.***out***.println(list.next());

}

}

}

Spring.xml file

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id=*"questions"* class=*"com.helloworldexample.Question"*>

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

<property name=*"question"* value=*"what is spring core"*></property>

<property name=*"answers"*>

<set>

<value>It enables to build applications</value>

<value>by this we can have singleton classes that is only one object</value>

<value>have the bean factory to create beanss</value>

</set>

</property>

</bean>

</beans>

TestClass.java file

**package** com.helloworldexample;

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

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

**public** **class** TestClass {

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

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

Question questions=(Question)context.getBean("questions");

questions.display();

}

}

OUTPUT::

101 what is spring core

answers:

It enables to build applications

by this we can have singleton classes that is only one object

have the bean factory to create beans

C)

Question.java file

**package** com.helloworldexample;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** Question {

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

**private** String question;

**private** Map<Integer,String> answers;

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

**return** id;

}

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

**this**.id = id;

}

**public** String getQuestion() {

**return** question;

}

**public** **void** setQuestion(String question) {

**this**.question = question;

}

**public** Map<Integer, String> getAnswers() {

**return** answers;

}

**public** **void** setAnswers(Map<Integer, String> answers) {

**this**.answers = answers;

}

**void** display() {

System.***out***.println(id+ " " +question);

System.***out***.println("answers:");

Set<Entry<Integer,String>> map=answers.entrySet();

Iterator<Entry<Integer,String>> list = map.iterator();

**while**(list.hasNext()) {

Entry<Integer,String> entry =list.next();

System.***out***.println("Sequence " +entry.getKey()+ ") " +entry.getValue());

}

}

}

Spring.xml file

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id=*"questions"* class=*"com.helloworldexample.Question"*>

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

<property name=*"question"* value=*"what is spring core"*></property>

<property name=*"answers"*>

<map>

<entry key=*"1"* value=*"It enables to build applications"*>

</entry>

<entry key=*"2"* value=*"by this we can have singleton classes that is only one object"*>

</entry>

<entry key=*"3"* value=*"have the bean factory to create beanss"*>

</entry>

</map>

</property>

</bean>

</beans>

TestClass.java file

**package** com.helloworldexample;

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

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

**public** **class** TestClass {

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

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

Question questions=(Question)context.getBean("questions");

questions.display();

}

}

OUTPUT:

101 what is spring core

answers:

Sequence 1) It enables to build applications

Sequence 2) by this we can have singleton classes that is only one object

Sequence 3) have the bean factory to create beanss

3)

**package** com.helloworldexample;

**public** **class** BankAccount {

**private** **int** accountid;

**private** String accountHolderName;

**private** String accountType;

**private** **double** balance;

**public** **int** getAccountid() {

**return** accountid;

}

**public** **void** setAccountid(**int** accountid) {

**this**.accountid = accountid;

}

**public** String getAccountHolderName() {

**return** accountHolderName;

}

**public** **void** setAccountHolderName(String accountHolderName) {

**this**.accountHolderName = accountHolderName;

}

**public** String getAccountType() {

**return** accountType;

}

**public** **void** setAccountType(String accountType) {

**this**.accountType = accountType;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

}

**package** com.helloworldexample;

**public** **interface** BankAccountRepository {

**public** **double** getBalance(**long** accountid);

**public** **double** updateBalance(**long** accountid,**double** Balance);

}

**package** com.helloworldexample;

**public** **class** BankAccountrepositoryimpl **implements** BankAccountRepository {

@Override

**public** **double** getBalance(**long** accountid) {

// **TODO** Auto-generated method stub

System.***out***.println("actual balance");

**return** updateBalance(0,0);

}

@Override

**public** **double** updateBalance(**long** accountid, **double** Balance) {

// **TODO** Auto-generated method stub

System.***out***.println("update the balance");

**return** updateBalance(0,0);

}

}

**package** com.helloworldexample;

**public** **interface** BankAccountService {

**public** **void** withdraw(**long** accountid,**double** balance);

**public** **void** deposit(**long** accountid,**double** balance);

**public** **void** getBalance(**long** accountid);

**public** **boolean** fundTransfer(**long** fromAccount,**long** toAccount,**double** amount);

}

**package** com.helloworldexample;

**public** **class** BankAccountServiceimpl **implements** BankAccountService {

@Override

**public** **void** withdraw(**long** accountid, **double** balance) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** deposit(**long** accountid, **double** balance) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** getBalance(**long** accountid) {

// **TODO** Auto-generated method stub

}

@Override

**public** **boolean** fundTransfer(**long** fromAccount, **long** toAccount, **double** amount) {

// **TODO** Auto-generated method stub

**return** **false**;

}

}

**package** com.helloworldexample;

**public** **class** BankAccountController {

**private** **int** accountid;

**private** **double** balance;

**private** **double** amount;

**private** **int** fromAccount;

**private** **int** toAccount;

**public** **int** getAccountid() {

**return** accountid;

}

**public** **void** setAccountid(**int** accountid) {

**this**.accountid = accountid;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** **double** getAmount() {

**return** amount;

}

**public** **void** setAmount(**double** amount) {

**this**.amount = amount;

}

**public** **int** getFromAccount() {

**return** fromAccount;

}

**public** **void** setFromAccount(**int** fromAccount) {

**this**.fromAccount = fromAccount;

}

**public** **int** getToAccount() {

**return** toAccount;

}

**public** **void** setToAccount(**int** toAccount) {

**this**.toAccount = toAccount;

}

**public** **double** withdraw() {

System.***out***.println("withdraw money:" +(balance-amount));

**return** balance=balance-amount;

}

**public** **double** deposit() {

System.***out***.println("Total balance:" +(balance+amount));

**return** balance=balance+amount;

}

**public** **boolean** fundtransfer(**long** fromAccount,**long** toAccount,**double** amount) {

**return** **false**;

}

}

**package** com.helloworldexample;

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

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

**public** **class** TestBankAccount {

**private** **static** ApplicationContext *context*;

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

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

BankAccountController controller=(BankAccountController)*context*.getBean("bankaccountcontroller");

controller.withdraw();

controller.deposit();

}

}

<?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*

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

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

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

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

<bean id = *"bankaccountcontroller"* class= *"com.helloworldexample.BankAccountController"* >

<property name=*"accountid"* value=*"1001"*></property>

<property name=*"balance"* value=*"25000"*></property>

<property name=*"amount"* value=*"10000"*></property>

<property name=*"fromAccount"* value=*"1001"*></property>

<property name=*"toAccount"* value=*"1002"*></property>

</bean>

<context:annotation-config/>

</beans>

OUTPUT:

withdraw money:15000.0

Total balance:25000.0

4)

@Autowired:

Triangle.java file:

**package** com.helloworldexample;

**import** java.util.List;

**public** **class** Triangle **implements** Shape {

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**public** Point getPointA() {

**return** pointA;

}

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** Point getPointB() {

**return** pointB;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** Point getPointC() {

**return** pointC;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw() {

System.***out***.println("drawing tiangle");

System.***out***.println("point A =(" +getPointA().getX() +","+getPointA().getY() +")");

System.***out***.println("point B =(" +getPointB().getX() +","+getPointB().getY() +")");

System.***out***.println("point C =(" +getPointC().getX() +","+getPointC().getY() +")");

}

}

Circle.java file:

**package** com.helloworldexample;

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

**public** **class** Circle **implements** Shape {

**private** Point center;

@Override

**public** **void** draw() {

System.***out***.println("drawing circle");

System.***out***.println("Circle point is: (" +center.getX()+"," +center.getY()+ ")");

}

**public** Point getCenter() {

**return** center;

}

@Autowired

**public** **void** setCenter(Point center) {

**this**.center = center;

}

}

Point.java file:

**package** com.helloworldexample;

**public** **class** Point {

**private** **int** x;

**private** **int** y;

**public** **int** getX() {

**return** x;

}

**public** **void** setX(**int** x) {

**this**.x = x;

}

**public** **int** getY() {

**return** y;

}

**public** **void** setY(**int** y) {

**this**.y = y;

}

}

Spring.xml file:

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id=*"pointA"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"0"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id = *"circle"* class= *"com.helloworldexample.Circle"* >

</bean>

<bean class=*"org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"*/>

</beans>

OUTPUT:

drawing circle

Circle point is: (0,0)

5)

@Required annotation::

Drawing.java file::

**import** org.springframework.beans.factory.BeanFactory;

**import** org.springframework.beans.factory.xml.XmlBeanFactory;

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

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

**import** org.springframework.core.io.FileSystemResource;

**public** **class** DrawingApp {

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

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

Shape shape=(Shape)context.getBean("circle");

shape.draw();

}

}

Shape.java file::

**package** com.helloworldexample;

**public** **interface** Shape {

**public** **void** draw();

}

Circle.java file:

**package** com.helloworldexample;

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

**public** **class** Circle **implements** Shape {

**private** Point center;

@Override

**public** **void** draw() {

System.***out***.println("drawing circle");

System.***out***.println("Circle point is: (" +center.getX()+"," +center.getY()+ ")");

}

**public** Point getCenter() {

**return** center;

}

@Required

**public** **void** setCenter(Point center) {

**this**.center = center;

}

}

Spring.xml file::

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id=*"pointA"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"0"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id = *"circle"* class= *"com.helloworldexample.Circle"* >

</bean>

<bean class=*"org.springframework.beans.factory.annotation.RequiredAnnotationBeanPostProcessor"*>

</bean>

</beans>

OUTPUT::

org.springframework.beans.factory.BeanInitializationException: Property 'center' is required for bean 'circle'

@Resource annotation::

Spring.xml file:

<?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*

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

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

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

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

<bean id = *"triangle"* class= *"com.helloworldexample.Triangle"* >

<property name=*"pointA"* ref=*"pointA"*/>

<property name=*"pointB"* ref=*"pointB"*/>

<property name=*"pointC"* ref=*"pointC"*/>

</bean>

<bean id=*"pointA"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"0"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id=*"pointB"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"-20"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id=*"pointC"* class=*"com.helloworldexample.Point"*>

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

<property name=*"y"* value=*"0"*/>

</bean>

<bean id = *"circle"* class= *"com.helloworldexample.Circle"* >

</bean>

<context:annotation-config/>

</beans>

Circle.java file:

package com.helloworldexample;

import org.springframework.context.annotation.ImportResource;

import javax.annotation.Resource;

public class Circle implements Shape {

private Point center;

@Override

public void draw() {

System.out.println("drawing circle");

System.out.println("Circle point is: (" +center.getX()+"," +center.getY()+ ")");

}

public Point getCenter() {

return center;

}

@Resource(name="pointC")

public void setCenter(Point center) {

this.center = center;

}

}

Drawing.java file:

**import** org.springframework.beans.factory.BeanFactory;

**import** org.springframework.beans.factory.xml.XmlBeanFactory;

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

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

**import** org.springframework.core.io.FileSystemResource;

**public** **class** DrawingApp {

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

//Triangle t=new Triangle();

//BeanFactory factory=new XmlBeanFactory(new FileSystemResource("spring.xml"));

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

Shape shape=(Shape)context.getBean("circle");

shape.draw();

}

}

OUTPUT::

drawing circle

Circle point is: (20,0)

8)

SPRING.XML:

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id = "triangle" **class**= "com.helloworldexample.Triangle" init-method="myInit" destroy-method="cleanUp">

<property name="pointA" ref="zeroPoint"/>

<property name="pointB" ref="onePoint"/>

<property name="pointC" ref="twoPoint"/>

</bean>

<bean id="zeroPoint" **class**="com.helloworldexample.Point">

<property name="x" value="0"/>

<property name="y" value="0"/>

</bean>

<bean id="onePoint" **class**="com.helloworldexample.Point">

<property name="x" value="-20"/>

<property name="y" value="0"/>

</bean>

<bean id="twoPoint" **class**="com.helloworldexample.Point">

<property name="x" value="20"/>

<property name="y" value="0"/>

</bean>

</beans>

**package** com.helloworldexample;

**import** java.util.List;

**public** **class** Triangle {

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**public** Point getPointA() {

**return** pointA;

}

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** Point getPointB() {

**return** pointB;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** Point getPointC() {

**return** pointC;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw() {

System.***out***.println("point A =(" +getPointA().getX() +","+getPointA().getY() +")");

System.***out***.println("point B =(" +getPointB().getX() +","+getPointB().getY() +")");

System.***out***.println("point C =(" +getPointC().getX() +","+getPointC().getY() +")");

}

**public** **void** myInit() {

System.***out***.println("my init method called for triangle");

}

**public** **void** cleanUp() {

System.***out***.println("my cleanUp method called for triangle");

}

}

OUTPUT:

my init method called **for** triangle

point A =(0,0)

point B =(-20,0)

point C =(20,0)

my cleanUp method called **for** triangle

BY USING @PostConstruct and @PreDestroy::

Circle.java file.

**package** com.helloworldexample;

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

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

**import** javax.annotation.Resource;

**public** **class** Circle **implements** Shape {

**private** Point center;

@Override

**public** **void** draw() {

System.***out***.println("drawing circle");

System.***out***.println("Circle point is: (" +center.getX()+"," +center.getY()+ ")");

}

**public** Point getCenter() {

**return** center;

}

@Resource(name="pointC")

**public** **void** setCenter(Point center) {

**this**.center = center;

}

@PostConstruct

**public** **void** initializecircle() {

System.***out***.println("Init of the circle");

}

@PreDestroy

**public** **void** destroycircle() {

System.***out***.println("destroy the circle");

}

}

Spring.xml file::

<?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*

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

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

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

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

<bean id = *"triangle"* class= *"com.helloworldexample.Triangle"* >

<property name=*"pointA"* ref=*"pointA"*/>

<property name=*"pointB"* ref=*"pointB"*/>

<property name=*"pointC"* ref=*"pointC"*/>

</bean>

<bean id=*"pointA"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"0"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id=*"pointB"* class=*"com.helloworldexample.Point"*>

<property name=*"x"* value=*"-20"*/>

<property name=*"y"* value=*"0"*/>

</bean>

<bean id=*"pointC"* class=*"com.helloworldexample.Point"*>

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

<property name=*"y"* value=*"0"*/>

</bean>

<bean id = *"circle"* class= *"com.helloworldexample.Circle"* >

</bean>

<context:annotation-config/>

</beans>

Shape.java file::

**package** com.helloworldexample;

**public** **interface** Shape {

**public** **void** draw();

}

Drawing.java file::

package com.helloworldexample;

/\*import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.core.io.FileSystemResource;

public class DrawingApp {

public static void main(String[] args) {

//Triangle triangle=new Triangle();

//BeanFactory factory= new XmlBeanFactory(new FileSystemResource("spring.xml"));

AbstractApplicationContext context=**new** ClassPathXmlApplicationContext("spring.xml");

context.registerShutdownHook(); triangle.draw();

}

}\*/

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.core.io.FileSystemResource;

**public** **class** DrawingApp {

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

//Triangle t=new Triangle();

//BeanFactory factory=new XmlBeanFactory(new FileSystemResource("spring.xml"));

AbstractApplicationContext context=**new** ClassPathXmlApplicationContext("spring.xml");

context.registerShutdownHook();

Shape shape=(Shape)context.getBean("circle");

shape.draw();

}

}

OUTPUT::

Init of the circle

drawing circle

Circle point is: (20,0)

destroy the circle

9)

INITIALIZATION callback method

TRIANGLE.JAVA FILE

**package** com.helloworldexample;

**import** java.util.List;

**import** org.springframework.beans.factory.InitializingBean;

**public** **class** Triangle **implements** InitializingBean{

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**public** Point getPointA() {

**return** pointA;

}

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** Point getPointB() {

**return** pointB;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** Point getPointC() {

**return** pointC;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw() {

System.***out***.println("point A =(" +getPointA().getX() +","+getPointA().getY() +")");

System.***out***.println("point B =(" +getPointB().getX() +","+getPointB().getY() +")");

System.***out***.println("point C =(" +getPointC().getX() +","+getPointC().getY() +")");

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

System.***out***.println("Initializing Beans init method called for Triangle");

}

}

SPRING.XML FILE

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

<bean id = "triangle" **class**= "com.helloworldexample.Triangle" >

<property name="pointA" ref="zeroPoint"/>

<property name="pointB" ref="onePoint"/>

<property name="pointC" ref="twoPoint"/>

</bean>

<bean id="zeroPoint" **class**="com.helloworldexample.Point">

<property name="x" value="0"/>

<property name="y" value="0"/>

</bean>

<bean id="onePoint" **class**="com.helloworldexample.Point">

<property name="x" value="-20"/>

<property name="y" value="0"/>

</bean>

<bean id="twoPoint" **class**="com.helloworldexample.Point">

<property name="x" value="20"/>

<property name="y" value="0"/>

</bean>

</beans>

OUTPUT:

Initializing Beans init method called **for** Triangle

point A =(0,0)

point B =(-20,0)

point C =(20,0)

DisposableBean ***METHOD***::

**package** com.helloworldexample;

**import** java.util.List;

**import** org.springframework.beans.factory.DisposableBean;

**import** org.springframework.beans.factory.InitializingBean;

**public** **class** Triangle **implements** InitializingBean,DisposableBean{

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**public** Point getPointA() {

**return** pointA;

}

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** Point getPointB() {

**return** pointB;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** Point getPointC() {

**return** pointC;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw() {

System.***out***.println("point A =(" +getPointA().getX() +","+getPointA().getY() +")");

System.***out***.println("point B =(" +getPointB().getX() +","+getPointB().getY() +")");

System.***out***.println("point C =(" +getPointC().getX() +","+getPointC().getY() +")");

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

System.***out***.println("Initializing Beans init method called for Triangle");

}

@Override

**public** **void** destroy() **throws** Exception {

System.***out***.println("DesposableBean destroy init method called for Triangle");

}

}

OUTPUT::

Initializing Beans init method called **for** Triangle

point A =(0,0)

point B =(-20,0)

point C =(20,0)

Aug 16, 2021 2:36:47 PM org.springframework.context.support.AbstractApplicationContext doClose

INFO: Closing org.springframework.context.support.ClassPathXmlApplicationContext@5e91993f: startup date [Mon Aug 16 14:36:46 IST 2021]; root of context hierarchy

DesposableBean destroy init method called **for** Triangle

10)

Triangle.java file::

**package** com.helloworldexample;

**import** java.util.List;

**import** org.springframework.beans.BeansException;

**import** org.springframework.beans.factory.BeanNameAware;

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

**import** org.springframework.context.ApplicationContextAware;

**public** **class** Triangle **implements** ApplicationContextAware, BeanNameAware {

**private** Point pointA;

**private** Point pointB;

**private** Point pointC;

**private** ApplicationContext context=**null**;

**public** Point getPointA() {

**return** pointA;

}

**public** **void** setPointA(Point pointA) {

**this**.pointA = pointA;

}

**public** Point getPointB() {

**return** pointB;

}

**public** **void** setPointB(Point pointB) {

**this**.pointB = pointB;

}

**public** Point getPointC() {

**return** pointC;

}

**public** **void** setPointC(Point pointC) {

**this**.pointC = pointC;

}

**public** **void** draw() {

System.***out***.println("drawing tiangle");

System.***out***.println("point A =(" +getPointA().getX() +","+getPointA().getY() +")");

System.***out***.println("point B =(" +getPointB().getX() +","+getPointB().getY() +")");

System.***out***.println("point C =(" +getPointC().getX() +","+getPointC().getY() +")");

}

@Override

**public** **void** setApplicationContext(ApplicationContext context) **throws** BeansException {

**this**.context=context;

}

@Override

**public** **void** setBeanName(String beanName) {

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

}

}

Drawing.java file::

package com.helloworldexample;

/\*import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.core.io.FileSystemResource;

public class DrawingApp {

public static void main(String[] args) {

//Triangle triangle=new Triangle();

//BeanFactory factory= new XmlBeanFactory(new FileSystemResource("spring.xml"));

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

Triangle triangle=(Triangle)context.getBean("triangle");

triangle.draw();

}

}\*/

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.core.io.FileSystemResource;

public class DrawingApp {

public static void main(String[] args) {

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

Shape shape=(Shape)context.getBean("circle");

shape.draw();

}

}

OUTPUT:

Bean name:triangle

Init of the circle

drawing circle

Circle point is: (20,0)