Name: Lutika Kolhe

Assignment on Spring Core

1. Create an Address class with the following attributes:- street, city, state, zip, country Create an Customer class with the following attributes:- customerld, customer Name, customerContact, customerAddress. object and print details of Customer. Inject the Address bean into Customer bean using setter injection Create a Test class with main() method, get Customer bean from ApplicationContext Also write the JUnit Test cases for above program.
   * Modify the above application and inject the bean using constructor injection
   * Use XML based Configuraion.

Class1: Address

**package** springCore;

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

**private** Long id;

**protected** String street;

**protected** String city;

**protected** String state;

**protected** String zip;

**protected** String country;

**public** Address() {

**this**.id = id;

**this**.street = street;

**this**.city = city;

**this**.state = state;

**this**.zip = zip;

**this**.country = country;

}

**public** Long getId()

{

**return** id;

}

**public** **void** setId()

{

**this**.id = id;

}

**public** String getStreet()

{

**return** street;

}

**public** **void** setStreet()

{

**this**.street = street;

}

**public** String getCity()

{

**return** city;

}

**public** **void** setCity()

{

**this**.city = city;

}

**public** String getState()

{

**return** state;

}

**public** **void** setState()

{

**this**.state = state;

}

**public** String getZip()

{

**return** zip;

}

**public** **void** setZip()

{

**this**.zip = zip;

}

**public** String getCountry()

{

**return** country;

}

**public** **void** setCountry()

{

**this**.country = country;

}

**public** **void** displayAdd()

{

System.***out***.println("Addresss = " + **this**.street + **this**.city + **this**.state + **this**.zip + **this**.country);

}

}

Class2:Coustomer

**package** springCore;

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

**protected** **int** customer\_id;

**protected** String customer\_name;

**protected** Address customer\_address;

**protected** String customer\_contact;

**public** Customer() {

**this**.customer\_id = customer\_id;

**this**.customer\_name = customer\_name;

**this**.customer\_address = customer\_address;

**this**.customer\_contact = customer\_contact;

}

**public** **int** getCustomerId()

{

**return** customer\_id;

}

**public** **void** setCustomerId()

{

**this**.customer\_id = customer\_id;

}

**public** String getCustomerName()

{

**return** customer\_name;

}

**public** **void** setCustomerName()

{

**this**.customer\_name = customer\_name;

}

**public** Address getCustomerAddress()

{

**return** customer\_address;

}

**public** **void** setCustomerAddress()

{

**this**.customer\_address = customer\_address;

}

**public** String getCustomerContact()

{

**return** customer\_contact;

}

**public** **void** setCustomerContact()

{

**this**.customer\_contact = customer\_contact;

}

**public** **void** display()

{

System.***out***.println("Customer Details - " + **this**.customer\_name + **this**.customer\_id + **this**.customer\_contact);

}

}

Class3:Test

**package** springCore;

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

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

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

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

customer.display();

}

}

Spring1.xml

<bean id = "Customer" class="io.first.Customer">

<property name="customer\_id" value="1"/>

<property name="customer\_name" value="Lutika kolhe"/>

<property name="customer\_contact" value="123"/>

<property name="street" ref="customer\_address"/>

<property name="city" ref="customer\_address"/>

<property name="state" ref="customer\_address"/>

<property name="zip" ref="customer\_address"/>

<property name="country" ref="customer\_address"/>

</bean>

<bean id="customer\_address" class="io.first.Address">

<property name="street" value="8"/>

<property name="city" value="Nagpur"/>

<property name="state" value="Maharashtra"/>

<property name="zip" value="44008"/>

<property name="country" value="India"/>

</bean>

**Output:**

id:1 name:Lutika kolhe Contact:123

Customer\_address: 8

Nagpur

Maharashtra

440008

India

1. Example of Injecting collections (List, Set and Map)

Create a class Question with following attributes: questionid, question, answers. There are 3 cases for above program.

1. Write a program where answers is of type List or String[]
2. Write a program where answers is of type Set
3. Write a program where answers is of type Map<Integer, String>. In case of Map, Integer value represents answer's sequence number.
4. Create a Test class with main() method, get Question bean from ApplicationContext object and print question and its answers.
5. Also write the JUnit Test cases for above program.
   * Use XML based configuration.

Ans:

Class1:Question.java

**package** secondQuestion;

**import** java.util.Iterator;

**import** java.util.List;

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

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

**private** String name;

**private** List<String> answers\_list;

**private** Set<String> answers\_set;

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

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

{

**return** id;

}

**public** **void** setId()

{

**this**.id = id;

}

**public** String getName()

{

**return** name;

}

**public** **void** setName()

{

**this**.name = name;

}

**public** ArrayList<Object>getList()

{

**return** answers\_list;

}

**public** **void** setList(ArrayList<Object> answers\_list)

{

**this**.answers\_list = answers\_list;

}

**public** Set<Object>getSet()

{

**return** answers\_set;

}

**public** **void** setSet(Set<Object> answers\_set)

{

**this**.answers\_set = answers\_set;

}

**public** Map<Integer, String>getMap()

{

**return** answers\_map;

}

**public** **void** setList(Map<Integer, String> answer\_map)

{

**this**.answers\_map = answers\_map;

}

**public** **void** displayList(){

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

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

Iterator<String> itr=answers\_list.iterator();

**while**(itr.hasNext()){

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

}

}

**public** **void** displaySet(){

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

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

Iterator<String> iter = answers\_set.iterator();

**while**(iter.hasNext()){

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

}

}

**public** **void** displayMap()

{

answers\_map.forEach((key, value) -> System.out.println(id + ":" + answers\_map));

}

}

Class2:Test

**Package** secondQuestion;

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

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

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

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

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

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

Resource r=**new** ClassPathResource("applicationContext.xml");

BeanFactory factory=**new** XmlBeanFactory(r);

Question q=(Question)factory.getBean("q");

q1.displayList();

Question q1=(Question)factory.getBean("q1");

q1.displaySet();

Question q2=(Question)factory.getBean("q2);

q1.displayMap();

}

}

Spring.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"*

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

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

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"q"* class=*"io.second.Question"*>

<property name=*"id"* value=*"1"*></property>

<property name=*"name"* value=*"What is Java?"*></property>

<property name=*"answers"*>

<list>

<value>Java is a programming language</value>

<value>Java is a platform</value>

<value>Java is an Island</value>

</list>

</property>

</bean>

<bean id=*"q1"* class=*"io.second.Question"*>

<property name=*"id"* value=*"2"*></property>

<property name=*"name"* value=*"Java is a platform"*></property>

<property name=*"answers"*>

<list>

<value>Java is a programming language</value>

<value>Java is a platform</value>

<value>Java is an Island</value>

</list>

</property>

</bean>

<bean id=*"q2"* class=*"io.second.Question"*>

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

<property name=*"name"* value=*"Java is an Island"*></property>

<property name=*"answers"*>

<list>

<value>Java is a programming language</value>

<value>Java is a platform</value>

<value>Java is an Island</value>

</list>

</property>

</bean>

</beans>

1. Example on @Controller, @Service, @Repository, @Autowired, @Configuration and @Bean. Modify the above application, use annotations and java based configuration.

* @Bean

Class1:ApplicationConfiguration

**package** springQuestion4;

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

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

//@Configuration

**public** **class** ApplicationConfiguration {

@Bean(name="demoService")

**public** DemoManager helloWorld()

{

**return** **new** DemoManagerImpl();

}

}

Class2:ApplicationConfiguration

**package** springQuestion4;

**public** **interface** DemoManager {

**public** String getServiceName();

}

Class3:DemoManager

**package** springQuestion4;

**public** **class** DemoManagerImp **implements** DemoManager

{

**public** String getServiceName()

{

**return** "Hello!!!!";

}

}

Class4:VerifySpringCoreFeature

**package** springQuestion4;

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

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

**public** **class** VerifySpringCoreFeature {

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

{

ApplicationContext context = **new** AnnotationConfigApplicationContext(ApplicationConfiguration.**class**);

DemoManager obj = (DemoManager) context.getBean("demoService");

System.***out***.println( obj.getServiceName() );

}

}

OUTPUT:

Addition of first and second = 4

* @contoller

Class1:SpringmainClass

**package** springQuestion4;

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

**public** **class** SpringMainClass {

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

AnnotationConfigApplicationContext context = **new** AnnotationConfigApplicationContext();

context.scan("spring\_exp4");

context.refresh();

MathController ms = context.getBean(MathController.**class**);

**int** result = ms.add(2, 2);

System.***out***.println("Addition of first and second = " + result);

context.close();

}

}

Class2:MathController

**package** springQuestion4;

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

**import** org.springframework.stereotype.Service;

@Controller

**public** **class** MathController {

**public** **int** add(**int** x, **int** y) {

**return** x + y;

}

}

OUTPUT:

Addition of first and second = 4

* @Service

Class1:MathService

**package** springQuestion4;

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

**import** org.springframework.stereotype.Service;

@Service("ms")

**public** **class** MathService {

**public** **int** add(**int** x, **int** y) {

**return** x + y;

}

}

Class1:SpringmainClass

**package** springQuestion4;

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

**public** **class** SpringMainClass {

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

AnnotationConfigApplicationContext context = **new** AnnotationConfigApplicationContext();

context.scan("spring\_exp4");

context.refresh();

MathService ms = context.getBean(MathService.**class**);

**int** result = ms.add(2, 2);

System.out.println("Addition of first and second = " + result);

context.close();

}

}

OUTPUT:

Addition of first and second = 4

* @Autowired

Class1:Mainbean

**package** springQuestion4;

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

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

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

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

**import** org.springframework.stereotype.Repository;

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

//@Repository

**public** **class** Mainbean {

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

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

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

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

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

shape.draw();

}

}

Class2:Triangle

**package** springQuestion4;

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

**private** Point PointA;

**private** Point PointB;

**private** Point PointC;

**public** Point getPointA() {

**return** PointA;

}

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

PointA = pointA;

}

**public** Point getPointB() {

**return** PointB;

}

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

PointB = pointB;

}

**public** Point getPointC() {

**return** PointC;

}

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

PointC = pointC;

}

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

{

System.***out***.println("Draw triangle");

System.***out***.println(getPointA().getX()+ " "+getPointA().getY());

System.***out***.println(getPointB().getX()+ " "+getPointB().getY());

System.***out***.println(getPointC().getX()+ " "+getPointC().getY());

}

}

Class3:Point

**package** springQuestion4;

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

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

}

**package** springQuestion4;

**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;

}

}

Class4:Circle

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

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

**import** org.springframework.stereotype.Repository;

//@Component

//@Repository

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

**private** Point center;

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

{

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

System.***out***.println("circle point" +center.getX() +center.getY());

}

**public** Point getCenter() {

**return** center;

}

//@Autowired

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

**this**.center = center;

}

}

spring.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*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id = *"triangle"* class=*"maths\_example.Triangle"*>

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

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

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

</bean>

<bean id = *"pointA"* class=*"maths\_example.Point"*>

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

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

</bean>

<bean id = *"pointB"* class=*"maths\_example.Point"*>

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

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

</bean>

<bean id = *"pointC"* class=*"maths\_example.Point"*>

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

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

</bean>

<bean id = *"center"* class=*"maths\_example.Point"*>

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

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

</bean>

<bean id = *"circle"* class=*"maths\_example.Circle"*>

<!-- <property names="center" ref="pointA"/> -->

</bean>

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

</bean>

</beans>

OUTPUT:

draw circle

circle point2010

1. Write a program to demonstrate use of @Resource, @inject, @Required annotations

* @Required

Class1:Employee

**package** assignment5;

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

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

**private** String name;

**private** String designation;

**private** String company;

@Required

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

**this**.name = name;

}

**public** String getName() {

**return** name;

}

@Required

**public** **void** setDesignation(String designation) {

**this**.designation = designation;

}

**public** String getDesignation() {

**return** designation;

}

**public** **void** setCompany(String company) {

**this**.company = company;

}

**public** String getCompany() {

**return** company;

}

@Override

**public** String toString() {

**return** "Employee [name=" + name + ", designation=" + designation + ", company=" + company + "]";

}

}

Class2:AppMain

**package** assignment5;

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

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

**import** assignment5.Employee;

**public** **class** AppMain {

@SuppressWarnings("resource")

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

ApplicationContext ac = **new** ClassPathXmlApplicationContext("required-annotation.xml");

Employee emp = ac.getBean("myemployee", Employee.**class**);

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

}

}

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

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"*>

<context:annotation-config />

<bean id=*"myemployee"* class=*"com.spring.pojo.Employee"*>

<!-- Required property -->

<property name=*"name"* value=*"Charlotte O' Neil"* />

<!-- Required property -->

<property name=*"designation"* value=*"Technical Leader"* />

<property name=*"company"* value=*"Test Ltd."* />

</bean>

</beans>

* @resource

Class1:Company

**package** assignment5;

**public** **class** Company {

**private** String name;

**private** String location;

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getLocation() {

**return** location;

}

**public** **void** setLocation(String location) {

**this**.location = location;

}

@Override

**public** String toString() {

**return** "Company [name=" + name + ", location=" + location + "]";

}

}

Class2:Employee

**package** assignment5;

**import** javax.annotation.Resource;

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

**private** String id;

**private** String name;

@Resource(name="mycompany")

**private** Company company;

**public** String getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** Company getCompany() {

**return** company;

}

**public** **void** setCompany(Company company) {

**this**.company = company;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", company=" + company.toString() + "]";

}

}

Class2:AppMain

**package** assignment5;

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

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

**import** com.spring.pojo.Employee;

**public** **class** AppMain {

@SuppressWarnings("resource")

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

ApplicationContext ac = **new** ClassPathXmlApplicationContext("resource-annotation.xml");

Employee emp = ac.getBean("myemployee", Employee.**class**);

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

}

}

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

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"*>

<!-- To activate the '@Resource' annotation in the spring framework -->

<context:annotation-config />

<bean id=*"mycompany"* class=*"com.spring.pojo.Company"*>

<property name=*"name"* value=*"Test Pvt. Ltd."* />

<property name=*"location"* value=*"India"* />

</bean>

<bean id=*"myemployee"* class=*"com.spring.pojo.Employee"*>

<property name=*"id"* value=*"123456"* />

<property name=*"name"* value=*"Charlotte O' Neil"* />

</bean>

</beans>

* @insert

Class1:RunMyProgram

**package** com.springexample;

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

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

**public** **class** RunMyProgram {

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

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

StudentHolder studentHolder = (StudentHolder) context.getBean("studentHolder");

studentHolder.displayStudentDetails();

}

}

Class2:StudentHolder

**package** com.springexample;

**import** javax.inject.Inject;

**public** **class** StudentHolder {

/\* Inject annotation wires the property byType by default \*/

@Inject

Student student;

**public** Student getStudent() {

**return** student;

}

**public** **void** setStudent(Student student) {

**this**.student = student;

}

**public** **void** displayStudentDetails(){

System.out.println("Student Details");

System.out.println("---------------");

System.out.println("Student No: "+student.getStudentNo());

System.out.println("Student Name: "+student.getStudentName());

}

}

Class2:Student

**package** com.springexample;

**public** **class** Student {

**private** **int** studentNo;

**private** String studentName;

**public** **int** getStudentNo() {

**return** studentNo;

}

**public** **void** setStudentNo(**int** studentNo) {

**this**.studentNo = studentNo;

}

**public** String getStudentName() {

**return** studentName;

}

**public** **void** setStudentName(String studentName) {

**this**.studentName = studentName;

}

}

spring.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"*

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<bean id=*"stu"* class=*"com.springexample.Student"*>

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

<property name=*"studentName"* value=*"John Peter"* />

</bean>

<bean id=*"studentHolder"* class=*"com.springexample.StudentHolder"* />

<context:annotation-config />

</beans>

1. Write a Java program to demonstrate SPEL (Spring Expression language)

Class2:Test

**import** org.springframework.expression.Expression;

**import** org.springframework.expression.ExpressionParser;

**import** org.springframework.expression.spel.standard.SpelExpressionParser;

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

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

ExpressionParser parser = **new** SpelExpressionParser();

Expression exp = parser.parseExpression("'Hello SPEL'");

String message = (String) exp.getValue();

System.***out***.println(message);

}

}

1. Write a Java program to demonstrate InitializingBean and DisposableBean.

Try Different ways:

* (Use init-method and destroy-method in xml config file)

Class1:CustomerService

**package** Question8;

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

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

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

**private** String msg;

**public** String getMsg() {

**return** msg;

}

**public** **void** setMsg(String msg) {

**this**.msg = msg;

}

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

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

System.out.println("Spring Container is destroy! Customer clean up");

}

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

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

System.out.println("Init method after properties are set : " + msg);

}}

Class2:CustomerService

**package** Question8;

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

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

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

**private** String msg;

**public** String getMsg() {

**return** msg;

}

**public** **void** setMsg(String msg) {

**this**.msg = msg;

}

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

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

System.out.println("Spring Container is destroy! Customer clean up");

}

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

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

System.out.println("Init method after properties are set : " + msg);

}}

Spring.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*

*http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"*>

<bean id=*"customerService"* class=*"QuestionEight.CustomerService"*>

<property name=*"msg"* value=*"i'm property message"* />

</bean>

</beans>

Output:

Init method after properties are set : i'm property message

Spring Container is destroy! Customer clean up

* (Use @PostConstruct and @PreDestroy)

Class1:MyBean

**package** Question8;

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

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

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

@Component

**public** **class** MyBean {

**public** MyBean() {

System.out.println("MyBean instance created");

}

@PostConstruct

**private** **void** init() {

System.out.println("Verifying Resources");

}

@PreDestroy

**private** **void** shutdown() {

System.out.println("Shutdown All Resources");

}

**public** **void** close() {

System.out.println("Closing All Resources");

}

}

Class2:MyConfiguration

**package** Question8;

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

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

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

@Configuration

**public** **class** MyConfiguration {

@Bean

@Scope(value = "singleton")

**public** MyBean myBean() {

**return** **new** MyBean();

}

}

Class3:SpringApp

**package** Question8;

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

**public** **class** SpringApp {

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

AnnotationConfigApplicationContext ctx = **new** AnnotationConfigApplicationContext();

ctx.register(MyConfiguration.**class**);

ctx.refresh();

MyBean mb1 = ctx.getBean(MyBean.**class**);

System.out.println(mb1.hashCode());

MyBean mb2 = ctx.getBean(MyBean.**class**);

System.out.println(mb2.hashCode());

ctx.close();

}

}

Output:

MyBean instance created

Verifying Resources

2145970759

2145970759

Shutdown All Resources

Closing All Resources

1. Write a Java program to demonstrate Complete Bean Life cycle.

Class1:Hello

**package** lifecycle.beans;

**public** **class** Hello

{

**public** **void** init() **throws** Exception

{

System.out.println(

"Welcome to SpringCore ");

}

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

{

System.out.println(

"Destroy method created");

}

}

Class2:Main

**package** lifecycle.beans;

**import** org.springframework

.context

.ConfigurableApplicationContext;

**import** org.springframework

.context.support

.ClassPathXmlApplicationContext;

**public** **class** Main {

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

**throws** Exception

{

ConfigurableApplicationContext cap

= **new** ClassPathXmlApplicationContext(

"spring9.xml");

cap.close();

}

}

Spring.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*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"hw"* class=*"lifecycle.beans.Hello"*

init-method=*"init"* destroy-method=*"destroy"*/>

</beans>

Output:

Welcome to SpringCore

Destroy method created

1. Write a java program to demonstrate ApplicationContextAware interface.

Class1:Employee

**package** AplicationContextAware;

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

**private** String Name;

**public** String getName() {

**return** Name;

}

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

Name = name;

}

@Override

**public** String toString() {

**return** "employee [Name=" + Name + "]";

}

}

Class2:AppContextAwareImp

**package** AplicationContextAware;

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

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

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

**public** **class** AppContextAwareImp **implements** ApplicationContextAware {

**private** ApplicationContext applicationContext;

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

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

System.out.println("set Application Context method is called by the spring container");

**this**.applicationContext = applicationContext;

}

**public** **void** displayEmployeeDetails() {

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

System.out.println("Got employee object from the applicationContext(Spring Container)=" + employee);

System.out.println("is employee object Singleton =" + applicationContext.isSingleton("employee"));

}

}

Class3:App

**package** AplicationContextAware;

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

**public** **class** App {

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

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

ClassPathXmlApplicationContext applicationContext= **new** ClassPathXmlApplicationContext("applicationContext.xml");

AppContextAwareImpl applicationContextAwareImpl= applicationContext.getBean("applicationContextAware",AppContextAwareImpl.**class**);

applicationContextAwareImpl.displayEmployeeDetails();

applicationContext.close();

}

}

Spring.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*

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

<bean id=*"employee"* class=*"AplicationContextAware.Employee"*>

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

</bean>

<bean id=*"applicationContextAware"* class=*"AplicationContextAware.AppContextAwareImpl"*></bean>

</beans>

Output:

set Application Context method is called by the spring container

Got employee object from the applicationContext(Spring Container)=employee [Name=peter]

is employee object Singleton =true