1.Create an Address class with the following attributes:-street, city, state, zip, country

Create an Customer class with the following attributes:-customerId, customerName, customerContact, customerAddress.

Inject the Address bean into Customer bean using setter injectionCreate a Test class with main() method, get Customer bean from ApplicationContext object and print details of Customer

.Also write the JUnit Test cases for above program.-Modify the above application and inject the bean using constructor injection-Use XML based Configuraion.

**package** first;

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

**private** String city, street, state, country;

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

**public** String getCity() {

**return** city;

}

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

**this**.city = city;

}

**public** String getStreet() {

**return** street;

}

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

**this**.street = street;

}

**public** String getState() {

**return** state;

}

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

**this**.state = state;

}

**public** String getCountry() {

**return** country;

}

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

**this**.country = country;

}

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

**return** zip;

}

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

**this**.zip = zip;

}

}

**package** first;

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

**private** **int** customerId;

**private** **int** customercontact;

**private** String customerName;

**private** Address customerAddress;

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

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId) {

**this**.customerId = customerId;

}

**public** **int** getCustomercontact() {

**return** customercontact;

}

**public** **void** setCustomercontact(**int** customercontact) {

**this**.customercontact = customercontact;

}

**public** String getCustomerName() {

**return** customerName;

}

**public** **void** setCustomerName(String customerName) {

**this**.customerName = customerName;

}

**public** Address getCustomerAddress() {

**return** customerAddress;

}

**public** **void** setCustomerAddress(Address customerAddress) {

**this**.customerAddress = customerAddress;

}

**public** String toString()

{

**return** "Customer Id: "+getCustomerId()+

", Customer Contact: "+getCustomercontact()+

", Customer Address: {City: "+getCustomerAddress().getCity()+

", Street: "+getCustomerAddress().getStreet()+

", State: "+getCustomerAddress().getState()+

", Country: "+getCustomerAddress().getCountry()+

", Zip: "+getCustomerAddress().getZip()+"},"

+ " Customer Name: "+getCustomerName();

}

}

**package** first;

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

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

**public** **class** main {

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

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

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

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

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

}

}

<?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 = *"Address"* class = *"first.Address"*>

<property name = *"zip"* value = *"120"*/>

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

<property name = *"street"* value = *"chruch Street"*/>

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

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

</bean>

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

<property name = *"customerId"* value = *"123"*/>

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

<property name = *"customerAddress"* ref = *"Address"*/>

<property name = *"customerName"* value = *"thanu"*/>

</bean>

</beans>

Output:

Customer Id: 123, Customer Contact: 123456, Customer Address: {City: bangalore, Street: chruch Street, State: karnataka, Country: india, Zip: 120}, Customer Name: thanu

Using constructor

**public** **class** Address

{

**private** String city, street, state, country;

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

**public** String getCity()

{

**return** city;

}

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

{

**this**.city = city;

}

**public** String getStreet()

{

**return** street;

}

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

{

**this**.street = street;

}

**public** String getState()

{

**return** state;

}

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

{

**this**.state = state;

}

**public** String getCountry()

{

**return** country;

}

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

{

**this**.country = country;

}

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

{

**return** zip;

}

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

{

**this**.zip = zip;

}

**public** Address(String city, String street, String state, String country, **int** zip) {

**super**();

**this**.city = city;

**this**.street = street;

**this**.state = state;

**this**.country = country;

**this**.zip = zip;

}

}

**public** **class** Customer

{

**private** **int** customerId;

**private** **int** customercontact;

**private** String customerName;

**private** Address customerAddress;

**public** Customer(**int** customerId, **int** customercontact, String customerName, Address customerAddress)

{

**this**.customerId = customerId;

**this**.customercontact = customercontact;

**this**.customerName = customerName;

**this**.customerAddress = customerAddress;

}

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

{

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId)

{

**this**.customerId = customerId;

}

**public** **int** getCustomercontact()

{

**return** customercontact;

}

**public** **void** setCustomercontact(**int** customercontact)

{

**this**.customercontact = customercontact;

}

**public** Address getCustomerAddress()

{

**return** customerAddress;

}

**public** **void** setCustomerAddress(Address customerAddress)

{

**this**.customerAddress = customerAddress;

}

**public** String getCustomerName()

{

**return** customerName;

}

**public** **void** setCustomerName(String customerName)

{

**this**.customerName = customerName;

}

**public** String toString()

{

**return** "Customer Id: "+getCustomerId()+ ", Customer Contact: "+getCustomercontact()+ ", Customer Address: {City: "+getCustomerAddress().getCity()+", Street: "+getCustomerAddress().getStreet()+", State: "+getCustomerAddress().getState()+", Country: "+getCustomerAddress().getCountry()+", Zip: "+getCustomerAddress().getZip()+"}, Customer Name: "+getCustomerName();

}

Main class

**package** first;

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

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

**public** **class** main {

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

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

//ApplicationContext context = new ClassPathXmlApplicationContext("Address.xml");

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

//System.out.println(customer);

ApplicationContext context = **new** ClassPathXmlApplicationContext("Address1.xml");

Customer constructor = (Customer) context.getBean("Customer1");

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

}

<?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 = *"Address"* class = *"first.Address"*>

<constructor-arg index = *"0"* name = *"city"* type = *"java.lang.String"* value = *"Bangalore"*/>

<constructor-arg index = *"1"* name = *"street"* type = *"java.lang.String"* value = *"Street"*/>

<constructor-arg index = *"2"* name = *"state"* type = *"java.lang.String"* value = *"Karnataka"*/>

<constructor-arg index = *"3"* name = *"country"* type = *"java.lang.String"* value = *"india"*/>

<constructor-arg index = *"4"* name = *"zip"* type = *"int"* value = *"221"*/>

</bean>

<bean id = *"Customer1"* class = *"first.Customer"*>

<constructor-arg index = *"0"* name = *"customerId"* type = *"int"* value = *"121"*/>

<constructor-arg index = *"1"* name = *"customercontact"* type = *"int"* value = *"90876"*/>

<constructor-arg index = *"2"* name = *"customerName"* type = *"java.lang.String"* value = *"shreyan"*/>

<constructor-arg index = *"3"* name = *"customerAddress"* ref = *"Address"*/>

</bean>

</beans>

Output:

Customer Id: 121, Customer Contact: 90876, Customer Address: {City: Bangalore, Street: Street, State: Karnataka, Country: india, Zip: 221}, Customer Name: shreyan

2. 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.

a.Writea program where answers is of type List<String> or String []

b.Write a program where answers is of type Set<String>

c.Write a program where answers is of type Map<Integer, String>In case of Map, Integer value represents answer’s sequence number.d.Create a Test class with main() method, get Questionbean from ApplicationContext object and print question and its answers.e.Also write the JUnit Test cases for above program.-Use XML based configuration.

**package** Second;

**public** **class** Collection {

**private** **int** questionId;

**private** String question, answers;

**public** **int** getQuestionId() {

**return** questionId;

}

**public** **void** setQuestionId(**int** questionId) {

**this**.questionId = questionId;

}

**public** String getQuestion() {

**return** question;

}

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

**this**.question = question;

}

**public** String getAnswers() {

**return** answers;

}

**public** **void** setAnswers(String answers) {

**this**.answers = answers;

}

@Override

**public** String toString() {

**return** "[questionId=" + questionId + ", question=" + question + ", answers=" + answers + "]";

}

}

**package** Second;

**import** java.util.ArrayList;

**import** java.util.Map;

**import** java.util.Set;

**public** **class** list\_set\_map\_class {

**private** ArrayList<Object> list;

**private** Set<Object> set;

**private** Map<Object,Object> map;

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

**return** list;

}

**public** **void** setList(ArrayList<Object> list) {

**this**.list = list;

}

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

**return** set;

}

**public** **void** setSet(Set<Object> set) {

**this**.set = set;

}

**public** Map<Object,Object> getMap() {

**return** map;

}

**public** **void** setMap(Map<Object,Object> map) {

**this**.map = map;

}

}

package Second;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class main\_class {

public static void main(String[] args) {

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

list\_set\_map\_class list1 =(list\_set\_map\_class) context.getBean("ListCollections");

System.out.println(("List: " + list1.getList()));

ApplicationContext con= new ClassPathXmlApplicationContext("set.xml");

list\_set\_map\_class set1 =(list\_set\_map\_class) con.getBean("SetCollections");

System.out.println(("SET: " + set1.getSet()));

ApplicationContext text= new ClassPathXmlApplicationContext("map.xml");

list\_set\_map\_class map1 =(list\_set\_map\_class) text.getBean("MapCollections");

System.out.println(("MAP: " + map1.getMap()));

}

}

<?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"*

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"ListCollections"* class=*"Second.list\_set\_map\_class"*>

<property name=*"list"*>

<list>

<ref bean=*"answers1"* />

<ref bean=*"answers2"* />

<ref bean=*"answers3"* />

</list>

</property>

</bean>

<bean id=*"answers1"* class=*"Second.Collection "*>

<property name=*"answers"* value =*"bindu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers2"* class=*"Second.Collection "*>

<property name=*"answers"* value=*"thanu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers3"* class=*"Second.Collection"*>

<property name=*"answers"* value=*"shreyan"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

<?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"*

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"SetCollections"* class=*"Second.list\_set\_map\_class"*>

<property name=*"set"*>

<set>

<ref bean=*"answers1"* />

<ref bean=*"answers2"* />

<ref bean=*"answers3"* />

</set>

</property>

</bean>

<bean id=*"answers1"* class=*"Second.Collection "*>

<property name=*"answers"* value =*"bindu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers2"* class=*"Second.Collection "*>

<property name=*"answers"* value=*"thanu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers3"* class=*"Second.Collection"*>

<property name=*"answers"* value=*"shreyan"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

<?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"*

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

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

xmlns:task=*"http://www.springframework.org/schema/task"* xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"MapCollections"* class=*"Second.list\_set\_map\_class"*>

<property name=*"map"*>

<map>

<entry key=*"1"* value-ref=*"answers1"* />

<entry key=*"2"* value-ref=*"answers2"* />

<entry key=*"3"* value-ref=*"answers3"* />

</map>

</property>

</bean>

<bean id=*"answers1"* class=*"Second.Collection "*>

<property name=*"answers"* value =*"bindu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"123"*/>

</bean>

<bean id=*"answers2"* class=*"Second.Collection "*>

<property name=*"answers"* value=*"thanu"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"456"*/>

</bean>

<bean id=*"answers3"* class=*"Second.Collection"*>

<property name=*"answers"* value=*"shreyan"*/>

<property name = *"question"* value = *"What's your name?"*/>

<property name = *"questionId"* value = *"789"*/>

</bean>

</beans>

Output:

List: [[questionId=123, question=What's your name?, answers=bindu], [questionId=456, question=What's your name?, answers=thanu], [questionId=789, question=What's your name?, answers=shreyan]]

SET: [[questionId=123, question=What's your name?, answers=bindu], [questionId=456, question=What's your name?, answers=thanu], [questionId=789, question=What's your name?, answers=shreyan]]

MAP: {1=[questionId=123, question=What's your name?, answers=bindu], 2=[questionId=456, question=What's your name?, answers=thanu], 3=[questionId=789, question=What's your name?, answers=shreyan]}

4)Example on @Controller, @Service, @Repository, @Autowired, @Configuration and @BeanModify the above application, use annotations and java based configuration.

**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();

}

}

**package** spring\_ex4;

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

**public** String getServiceName();

}

**package** spring\_ex4;

**public** **class** DemoManagerImpl **implements** DemoManager

{

**public** String getServiceName()

{

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

}

}

**package** spring\_ex4;

**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() );

}

}

@Service

package spring\_exp4;

import org.springframework.stereotype.Component;

import org.springframework.stereotype.Service;

@Service("ms")

//@Component

public class MathService {

public int add(int x, int y) {

return x + y;

}

} **package** spring\_exp4;

**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:

Jan 25, 2022 2:01:42 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext@20322d26: startup date [Tue Jan 25 14:01:42 IST 2022]; root of context hierarchy

Addition of first and second = 4

Jan 25, 2022 2:01:43 PM org.springframework.context.support.AbstractApplicationContext doClose

INFO: Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@20322d26: startup date [Tue Jan 25 14:01:42 IST 2022]; root of context hierarchy

@contoller

**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();

}

}

;

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

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

//@Service("ms")

@Controller

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

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

**return** x + y;

}

}

Output:

Jan 25, 2022 2:04:52 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext@20322d26: startup date [Tue Jan 25 14:04:52 IST 2022]; root of context hierarchy

Addition of first and second = 4

Jan 25, 2022 2:04:52 PM org.springframework.context.support.AbstractApplicationContext doClose

INFO: Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@20322d26: startup date [Tue Jan 25 14:04:52 IST 2022]; root of context hierarchy

@Autowired

**package** maths\_example;

**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();

}

}

**package** maths\_example;

**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());

}

} **package** maths\_example;

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

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

}

**package** maths\_example;

**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;

}

}

**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;

}

}

<?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:

Jan 25, 2022 2:10:56 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@31610302: startup date [Tue Jan 25 14:10:56 IST 2022]; root of context hierarchy

Jan 25, 2022 2:10:56 PM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [spring.xml]

draw circle

5)Write a program to demonstrate use of @Resource, @Inject, @Requiredannotations

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

public class Demonstrate {

private String msg;

private int msgID;

private ResourceEg recEg;

public int getMsgID() {

return msgID;

}

public void setMsgID(int msgID) {

this.msgID = msgID;

}

public String getMsg() {

return msg;

}

@Required

public void setMsg(String msg) {

this.msg = msg;

}

}

public class ResourceEg {

private String recID;

public String getRecID() {

return recID;

}

public void setRecID(String recID) {

this.recID = recID;

}

}

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class TestDemonstrate {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext con=new ClassPathXmlApplicationContext("Spring5Eg.xml");

Demonstrate demo=(Demonstrate) con.getBean("msg1");

System.out.println(demo.getMsgID()+" "+demo.getMsg() );

}

}

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

[<beans 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**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**" xmlns="**http://www.springframework.org/schema/beans**">](file:///C:\Users\BINDBR\Downloads\Spring5Eg%20(1).xml)

<context:annotation-config/>

[<bean class="**Demonstrate**" id="**msg1**">](file:///C:\Users\BINDBR\Downloads\Spring5Eg%20(1).xml)

<property value="**2021**" name="**msgID**"/>

<property value="**Spring Q.5**" name="**msg**"/></bean>

</beans>

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

**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 Maven'");

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

System.out.println(message);

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("'Welcome To'.concat('SpringExpression')");

String message = (String) exp.getValue(); System.out.println(message);

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

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

byte[] bytes = (byte[]) exp.getValue();

for(int i=0;i<bytes.length;i++) {

System.out.print(bytes[i]+" ");

}

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp =

parser.parseExpression("new String('hello world').toUpperCase()");

String message = exp.getValue(String.class); System.out.println(message);

\*/

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

Expression exp = parser.parseExpression("This is bindu get the length of it");

**int** length = (Integer) exp.getValue();

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

}

}

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.maven7</groupId>

<artifactId>SPELexpression</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.15</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-expression -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>5.3.15</version>

</dependency>

</dependencies>

</project>

Output:

Hello Maven

Concat : Welcome To SpringExpression

StringtoByte : 72 101 108 108 111 32 87 111 114 108 100

UpperCase : HELLO WORLD

Length : 10

8)Write a Java program to demonstrate InitializingBean and DisposableBean.

Try Different ways:(Use init-method and destroy-method in xml config file)

(Use @PostConstruct and @PreDestroy)

Type1

package Eight;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class FirstMethod {

public static void main(String[] args) {

ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(

new String[] { "Spring.xml" });

method mes = (method) context.getBean("Method");

System.out.println(mes);

context.close();

}

}

package Eight;

import org.springframework.beans.factory.DisposableBean;

import org.springframework.beans.factory.InitializingBean;

public class method implements InitializingBean, DisposableBean {

private String msg;

public String getMsg() {

return msg;

}

public void setMsg(String msg) {

this.msg = msg;

}

public void destroy() throws Exception {

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

}

public void afterPropertiesSet() throws Exception {

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

}

}

<?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=*"Method"* class=*"Eight.method"*>

<property name=*"msg"* value=*"property message"* />

</bean>

</beans>

Output:

Init method after properties are set : property message

Spring Container is destroy! Method clean up

**Type2**

**package** Eight;

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

**public** **class** SecondMethod {

**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();

}

}

**package** Eight;

**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");

}

//@PreDestory

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

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

}

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

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

}

}

**package** Eight;

**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();

}

}

Output:

MyBean instance created

1652807864

1652807864

Closing All Resources

**9. Write a Java program to demonstrate complete Bean Life cycle .**

**public** **class** HelloWorld {

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

{

System.***out***.println(

"Bean HelloWorld has been "

+ "instantiated and I'm "

+ "the init() method");

}

// This method executes

// when the spring container

// is closed

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

{

System.***out***.println(

"Container has been closed "

+ "and I'm the destroy() method");

}

}

**import** org.springframework.context.ConfigurableApplicationContext;

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

**public** **class** Client {

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

// Loading the Spring XML configuration

// file into the spring container and

// it will create the instance of

// the bean as it loads into container

ConfigurableApplicationContext cap = **new** ClassPathXmlApplicationContext("spring.xml");

// It will close the spring container

// and as a result invokes the

// destroy() method

cap.close();

}

}

<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=*"HelloBean"* class=*"Nine.HelloBean"*>

<property name =*"name"* value=*"Bindu"*>

</property>

</bean>

</beans>

**OUTPUT:**

Bean HelloWorld has been instantiated and I'm the init() method

Container has been closed and I'm the destroy() method

**10. Write a Java program to demonstrate ApplicationContextAware interface .**

**ANS:**

**Class 1:**

**package** Ten;

**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 + "]";

}

}

package Test;

import org.springframework.beans.BeansException;

import org.springframework.context.ApplicationContext;

import org.springframework.context.ApplicationContextAware;

public class AppContextAwareImpl implements ApplicationContextAware {

private ApplicationContext applicationContext;

public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {

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"));

}

}

**package** AplicationContextAware;

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

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

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

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

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

applicationContextAwareImpl.displayEmployeeDetails();

applicationContext.close();

}

}

<?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 =*"bindu"* />

</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=bindu]

is employee object Singleton =true