**Spring Core Assignments**

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

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

Inject the Address bean into Customer bean using setter injection

Create 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 Configuration.

**package** org.com;

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

**private** String street;

**private** String city;

**private** String zip;

**private** String state;

**private** String country;

**public** String getState() {

**return** state;

}

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

**this**.state = state;

}

**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 getZip() {

**return** zip;

}

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

**this**.zip = zip;

}

**public** String getCountry() {

**return** country;

}

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

**this**.country = country;

}

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

**super**();

**this**.street = street;

**this**.city = city;

**this**.zip = zip;

**this**.state = state;

**this**.country = country;

}

}

**package** org.com;

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

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

**private** String customerName;

**private** Long customerContact;

**private** Address customerAddress;

**public** Customer(**int** customerId, String customerName, Long customerContact, Address customerAddress) {

**super**();

**this**.customerId = customerId;

**this**.customerName = customerName;

**this**.customerContact = customerContact;

**this**.customerAddress = customerAddress;

}

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

**return** customerId;

}

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

**this**.customerId = customerId;

}

**public** String getCustomerName() {

**return** customerName;

}

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

**this**.customerName = customerName;

}

**public** Long getCustomerContact() {

**return** customerContact;

}

**public** **void** setCustomerContact(Long customerContact) {

**this**.customerContact = customerContact;

}

**public** Address getCustomerAddress() {

**return** customerAddress;

}

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

**this**.customerAddress = customerAddress;

}

@Override

**public** String toString() {

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

**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** org.com;

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

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

**public** **class** AddCust {

@SuppressWarnings("unused")

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

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

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

@SuppressWarnings("resource")

ApplicationContext context = **new** ClassPathXmlApplicationContext("file:src/spring.xml");

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

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

}

}

package org.com;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class CustAdd {

@SuppressWarnings("unused")

private static ApplicationContext context;

public static void main(String[] args) {

// TODO Auto-generated method stub

@SuppressWarnings("resource")

ApplicationContext context1 = new ClassPathXmlApplicationContext("file:src/spring1.xml");

Customer customer2 = (Customer)context1.getBean("Customer1");

System.out.println(customer2);

}

}

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN" "http://www.springframework.org/dtd/spring-beans-2.0.dtd">

<beans>

<bean id=*"Address"* class=*"org.com.Address"*>

<property name=*"zip"* value=*"221"*/>

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

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

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

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

</bean>

<bean id=*"Customer"* class=*"org.com.Customer"*>

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

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

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

<property name=*"customerName"* value=*"Sherlock Holmes"*/>

</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"*

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

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

<bean id=*"Address"* class=*"org.com.Address"*>

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

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

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

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

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

</bean>

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

<constructor-arg index=*"0"* name=*"customerId"* type=*"int"* value=*"123"*/>

<constructor-arg index=*"1"* name=*"customerContact"* type=*"int"* value=*"123456"*/>

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

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

</bean>

</beans>

**Output:** Customer Id: 123, Customer Contact: 123456, Customer Address: {City: London, Street: Baker Street, State: World, Country: England, Zip: 221}, Customer Name: Sherlock Holmes

1. Example of injecting collection(List,Set and Map)

Create a class Question with following attributes : questioneId, question, answers.

There are 3 cases for above program.

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

**package** org.com;

**public** **class** Ques {

**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() {

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

**return**"Question ID: "+questionid+ ", Questions: "+question+ ",Answers: "+answers+"\n";

}}

package org.com;

import java.util.ArrayList;

import java.util.Map;

import java.util.Set;

public class List1 {

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** org.com;

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

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

**public** **class** Quest {

@SuppressWarnings("unused")

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

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

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

@SuppressWarnings("resource")

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

List1 list = (List1) context.getBean("Collections");

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

}

}

<?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=*"Collections"* class=*"org.com.List1"*>

<property name=*"list"*>

<list>

<ref bean=*"answers"*/>

<ref bean=*"answers1"*/>

<ref bean=*"answers2"*/>

</list>

</property>

</bean>

<bean id=*"answers"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers1"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers2"* class=*"org.com.Ques"*>

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

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

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

</bean>

</beans>

**Output:**  List: [Question ID: 123, Questions: What's your name?, Answers: Teja

, Question ID: 456, Questions: What's your name?, Answers: Ritu

, Question ID: 789, Questions: What's your name?, Answers: Manu

]

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

**package** org.com;

**public** **class** Ques {

**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() {

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

**return**"Question ID: "+questionid+ ", Questions: "+question+ ", Answers: "+answers+"\n";

}}

package org.com;

import java.util.ArrayList;

import java.util.Map;

import java.util.Set;

public class List1 {

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** org.com;

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

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

**public** **class** Quest {

@SuppressWarnings("unused")

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

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

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

@SuppressWarnings("resource")

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

/\* List1 list = (List1) context.getBean("Collections");

System.out.println("List: " +list.getList());\*/

List1 set = (List1) context.getBean("Collections");

System.***out***.println("Set : " + set.getSet());

}

}

<?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=*"Collections"* class=*"org.com.List1"*>

<property name=*"set"*>

<set>

<ref bean=*"answers"*/>

<ref bean=*"answers1"*/>

<ref bean=*"answers2"*/>

</set>

</property>

</bean>

<bean id=*"answers"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers1"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers2"* class=*"org.com.Ques"*>

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

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

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

</bean>

</beans>

**Output:**

Set : [Question ID: 123, Questions: What's your name?, Answers: Teja

, Question ID: 456, Questions: What's your name?, Answers: Ritu

, Question ID: 789, Questions: What's your name?, Answers: Manu

]

1. Write a program where answers is of type Map<Integer, String>

In case of Map, Integer value represents answer’s sequence number.

**package** org.com;

**public** **class** Ques {

**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() {

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

**return**"Question ID: "+questionid+ ", Questions: "+question+ ", Answers: "+answers+"\n";

}}

package org.com;

import java.util.ArrayList;

import java.util.Map;

import java.util.Set;

public class List1 {

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** org.com;

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

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

**public** **class** Quest {

@SuppressWarnings("unused")

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

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

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

@SuppressWarnings("resource")

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

List1 map= (List1) context.getBean("Collections");

System.***out***.println("Map : " + map.getMap());

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

// System.out.println("Set : " + set.getSet());

}

}

<?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=*"Collections"* class=*"org.com.List1"*>

<property name=*"map"*>

<map>

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

<entry key =*"2"* value-ref = *"answers1"* />

<entry key =*"3"* value-ref = *"answers2"* />

</map>

</property>

</bean>

<bean id=*"answers"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers1"* class=*"org.com.Ques"*>

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

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

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

</bean>

<bean id=*"answers2"* class=*"org.com.Ques"*>

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

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

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

</bean>

</beans>

**Output:** Map : {1=Question ID: 123, Questions: What's your name?, Answers: Teja

, 2=Question ID: 456, Questions: What's your name?, Answers: Ritu

, 3=Question ID: 789, Questions: What's your name?, Answers: Manu

}

1. Create a Test class with main() method, get Question bean from ApplicationContext object and print question and its answers
2. Also write the JUnit Test cases for above program.

-Use XML based configuration

1. Example on autowiring

Design and Develop a Banking Application as follows:

1. Create a BankAccount class with following attributes: accounted, accountHolderName, accountType, accountBalance
2. Create an interface BankAccountRepository with following methods:

public double getBalance(long accountId)

public double updateBalance(long accounted, double newBalance)

Note: Above method returns updated balance

1. Create a class BankAccountrepositoryImpl that implements BankAccountRepository Interface.

You can use database or any collection object as persistence store.

1. Create an Interface BankAccountService with following methods:

public double withdraw(long accounted, double balance)

public double deposit(long accounted, double balance)

public double getBalance(long accounted)

public Boolean fundTransfer(long fromAccount, long toAccount, double amount)

1. Create a class BankAccountServiceImpl that implements BankAccountService interface.
2. Create a class BankAccount controller with following operations:

public double withdraw(long accounted, double balance)

public double deposit(long accounted, double balance)

public double getBalance(long accounted)

public0 Boolean fundTransfer(long fromAccount, long toAccount, double amount)

1. Create a Test class main() method, get BankAccountController bean object from ApplicationContext and perform all the operation.
2. Also write the JUnit Test cases for the above program.

Use XML based configuration and perform autowiring with different types.

(byName, byType and constructor). Use one autowiring type at a time.

1. Example on @Controller, @Service, @Repository, @AutoWired, @Configuration and @Bean

Modify the above application, use annotations and Java based configuration

@Bean

**package** org.com;

**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** org.com;

**public interface** DemoManager {

**public** String getServiceName();

}

**package** org.com;

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

{

**public** String getServiceName()

{

**Return** "Hello!!!!";

}

}

**package** org.com;

**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 org.com;

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** org.com;

**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();

MathServicems = 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

@contoller

**package** org.com;

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

**public class** SpringMainClass {

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

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

context.scan("org.com");

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();

}

}

package org.com;

**package** spring\_exp4;

**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:**  Addition of first and second = 4

@Autowired

**Package** org.com;

**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** org.com;

**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** org.com;

**public interface** Shape {

**public void** draw();

}

**package** org.com;

**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"?>

<beansxmlns = "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">

<beanid = "triangle"class="maths\_example.Triangle">

<propertyname="PointA"ref="pointA"/>

<propertyname="PointB"ref="pointB"/>

<propertyname="PointC"ref="pointC"/>

</bean>

<beanid = "pointA"class="maths\_example.Point">

<propertyname="x"value="0"/>

<propertyname="y"value="10"/>

</bean>

<beanid = "pointB"class="maths\_example.Point">

<propertyname="x"value="10"/>

<propertyname="y"value="10"/>

</bean>

<beanid = "pointC"class="maths\_example.Point">

<propertyname="x"value="20"/>

<propertyname="y"value="10"/>

</bean>

<beanid = "center"class="maths\_example.Point">

<propertyname="x"value="20"/>

<propertyname="y"value="10"/>

</bean>

<beanid = "circle"class="maths\_example.Circle">

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

</bean>

<beanclass="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor">

</bean>

</beans>

**Output:** draw circle

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

**@Resource Annotation**

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

**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;

}

@Required

**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() + "]";

}

}

**package** org.com;

**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 + "]";

}

}

**package** org.com;

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

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

**public class** Test {

@SuppressWarnings("resource")

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

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

Employee emp=(Employee) context.getBean("myemployee");

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

}

}

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"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/>

<beanid=*"mycompany"*class=*"New.Company"*>

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

<propertyname=*"location"*value=*"India"*/>

</bean>

<beanid=*"myemployee"*class=*"New.Employee"*>

<propertyname=*"id"*value=*"126"*/>

<propertyname=*"name"*value=*"Nam"*/>

</bean></beans>

@Inject Annotataion

**public class** Employee {

**private** String id;

**private** String name;

@Inject

**private** Company company;

**public** String getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return**name;

}

@Required

**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() + "]";

}

}

CLASS 2

**package**New;

**publicclass** Company {

**private** String name;

**private** String location;

**public** String getName() {

**return**name;

}

**publicvoid**setName(String name) {

**this**.name = name;

}

**public** String getLocation() {

**return**location;

}

**publicvoid**setLocation(String location) {

**this**.location = location;

}

@Override

**public** String toString() {

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

}

}

CLASS 3

**package** ;

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

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

**public class** Test {

@SuppressWarnings("resource")

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

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

Employee emp=(Employee) context.getBean("myemployee");

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

}

}

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"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)

*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/>

<beanid=*"mycompany"*class=*"New.Company"*>

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

<propertyname=*"location"*value=*"India"*/>

</bean>

<beanid=*"myemployee"*class=*"New.Employee"*>

<propertyname=*"id"*value=*"126"*/>

<propertyname=*"name"*value=*"Nam"*/>

</bean>

</beans>

**Output:** Employee [id=126, name=Nam, company=Company [name=Test Pvt. Ltd., location=India]]

1. Example of @Component, @Value, @PropertySource & Environment
2. Create a dbConfig.properties file which contains database configuration details like driver class name, dburl, username, password
3. Create a Java class in which you have to read all properties and display on a console. (Use @Component, @Value or Environment and @PropertyResource).
4. Write a Java Program to demonstrate SPEL (Spring Expression Language)

**package** com.maven7;

**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);

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

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

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

System.out.println(message);

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

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

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

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

}

}

<projectxmlns="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

1. Write a Java Program to demonstrate initializingBean and DisposalBean.

Try Different Ways:

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

(Use @PostConstruct and @PreDestroy)

1st Way

**package** org.com;

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

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

**public** **class** CustServ **implements** InitializingBean, DisposalBean {

**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);

}

}

package org.com;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Snip {

public static void main(String[] args) {

// TODO Auto-generated method stub

ConfigurableApplicationContext context = new ClassPathXmlApplicationContext(new String[] { "SpringCust.xml" });

CustServ cust = (CustServ) context.getBean("customerService");

System.out.println(cust);

context.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-2.5.xsd"*>

<bean id=*"customerService"* class=*"org.com.CustServ"*>

<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

2nd Way

packageorg.com;

importjavax.annotation.PostConstruct;

importjavax.annotation.PreDestroy;

importorg.springframework.beans.factory.annotation.Autowired;

importorg.springframework.stereotype.Component;

@Component

public classMyBean {

publicMyBean() {

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

}

@PostConstruct

private voidinit() {

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

}

@PreDestroy

private void shutdown() {

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

}

public void close() {

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

}

}

packageorg.com;

importorg.springframework.context.annotation.Bean;

importorg.springframework.context.annotation.Configuration;

importorg.springframework.context.annotation.Scope;

@Configuration

public **class** MyConfiguration {

@Bean

@Scope(value = "singleton")

publicMyBean myBean() {

return newMyBean();

}

}

packageorg.com;

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();

MyBeanmb1 = ctx.getBean(MyBean.**class**);

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

MyBeanmb2 = 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.

**package** org.com;

**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");

}

}

**package** org.com;

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

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

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

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

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

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

cap.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=*"HelloBean"* class=*"org.com.HelloWorld"*>

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

</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

1. Write a Java Program to demonstrate ApplicationContextAware interface

**package** org.com;

**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 org.com;

import org.springframework.beans.BeansException;

import org.springframework.context.ApplicationContext;

import org.springframework.context.ApplicationContextAware;

public class AppContextAwareImpl implements ApplicationContextAware {

private ApplicationContextapplicationContext;

public void setApplicationContext(ApplicationContextapplicationContext) 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"));

}

}

**package** org.com;

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

**public class** App {

**public staticvoid** main(String[] args) {

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

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

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

applicationContextAwareImpl.displayEmployeeDetails();

applicationContext.close();

}

}

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"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"*>

<beanid=*"employee"*class=*"AplicationContextAware.Employee"*>

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

</bean>

<beanid=*"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