<http://javaconceptoftheday.com/duplicate-characters-in-a-string-in-java/>

deutsche bank

# An Interview Question on Spring Singletons

Please tell me what the output of the following program would be."

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="scopeTest" class="com.example.scope.Scope" scope="singleton">

<property name="name" value="Shamik Mitra"/>

</bean>

<bean id="scopeTestDuplicate" class="com.example.scope.Scope" scope="singleton">

<property name="name" value="Samir Mitra"/>

</bean>

</beans>

Scope.java:

package com.example.scope;

public class Scope {

private String name;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Scope [name=" + name + "]";

}

}

Main class:

package com.example.scope;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

public static void main(String[] args) {

ApplicationContext ctx = new ClassPathXmlApplicationContext(

"configFiles/Scope.xml");

Scope scope = (Scope) ctx.getBean("scopeTest");

Scope scopeDuplicate = (Scope) ctx.getBean("scopeTestDuplicate");

System.out.println(scope == scopeDuplicate); //false

System.out.println(scope + "::" + scopeDuplicate);

}

}

Output:

Reference Check ::false

Scope [name=Shamik Mitra]::Scope [name=Samir Mitra]

# A Spring Singleton does not work like a Java Singleton

The reference check will return false, which means Spring Singletons don't work like they said earlier. (**A few**)

2) Final class of Hibernate entity

3)java filename what happend

4) how class loader load class where path is specefied

5) override Static method with non staic method

6) what happend if we don't override hashcode and equals method

7)Generalization and Specilization

8) Association and Composition

9) which is best Array.sort or Collections.sort

There are 5 differences between Array and Collection as given below :

1. Array is Fixed in Size. Where , Collection is Grow able in nature.
2. Array stores homogeneous data . Where , Collection store both homogeneous as well as Heterogeneous data.
3. In Array , there are no underlined Data Structures, whereas ,Collection has Underlined DS.
4. Array is recommended in performance , whereas Collection is not.
5. Array use more memory space compare to Collection.

10) where configuration defined to run java code from any where in machine using Command prompt

11) Transaction Manager configuration in Spring

The JtaTransactionManager does not need to know about the DataSource, or any other specific resources, because it uses the container’s global transaction management infrastructure.

|  |
| --- |
| <!-- START : Transaction manager and AOP configurations -->  <bean id=*"transactionManager"* class=*"org.springframework.orm.jpa.JpaTransactionManager"*>  <property name=*"entityManagerFactory"* ref=*"entityManagerFactory"* />  <property name=*"dataSource"* ref=*"dataSource"* />  </bean>  <!-- CP Manager Transactions -->  // **public** **interface** CustomerProfileManager  <aop:config>  <aop:pointcut id=*"customerProfileManagerPointCut"*  expression=*"execution(\* \*..CustomerProfileManager.\*(..))"* />  <aop:advisor advice-ref=*"cpManagerTransactionAdvice"*  pointcut-ref=*"customerProfileManagerPointCut"* />  </aop:config>  <tx:advice id=*"cpManagerTransactionAdvice"* transaction-manager=*"transactionManager"*>  <tx:attributes>  <tx:method name=*"createCustomerProfile"* propagation=*"REQUIRED"* />  <tx:method name=*"updateCustomerProfile"* propagation=*"REQUIRED"*  rollback-for=*"Exception"* />  <tx:method name=*"retrieveCustomerProfile"* propagation=*"REQUIRED"* />  <tx:method name=*"getCustomerEvents"* propagation=*"REQUIRED"* />  </tx:attributes>  </tx:advice> |

[**Propagation**](http://static.springsource.org/spring/docs/3.0.x/javadoc-api/org/springframework/transaction/annotation/Propagation.html)

Defines how transactions relate to each other. Common options

* Required: Code will always run in a transaction. Create a new transaction or reuse one if available.
* Requires\_new: Code will always run in a new transaction. Suspend current transaction if one exists.

12) Cascade Type Available:

## org.hibernate.annotations  Enum CascadeType

[java.lang.Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)

extended by [java.lang.Enum](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Enum.html?is-external=true)<[CascadeType](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html" \o "enum in org.hibernate.annotations)>

extended by **org.hibernate.annotations.CascadeType**

**All Implemented Interfaces:**

[Serializable](http://java.sun.com/j2se/1.5.0/docs/api/java/io/Serializable.html?is-external=true), [Comparable](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Comparable.html?is-external=true)<[CascadeType](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html" \o "enum in org.hibernate.annotations)>

public enum **CascadeType**

extends [Enum](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Enum.html?is-external=true)<[CascadeType](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html" \o "enum in org.hibernate.annotations)>

Cascade types (can override default EJB3 cascades

|  |  |
| --- | --- |
| **Enum Constant Summary** | |
| [**ALL**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#ALL) |  |
| [**DELETE**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#DELETE) |  |
| [**DELETE\_ORPHAN**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#DELETE_ORPHAN)            **Deprecated.** *use @OneToOne(orphanRemoval=true) or @OneToMany(orphanRemoval=true)* |  |
| [**DETACH**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#DETACH) |  |
| [**EVICT**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#EVICT)            **Deprecated.** *use javax.persistence.CascadeType.DETACH* |  |
| [**LOCK**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#LOCK) |  |
| [**MERGE**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#MERGE) |  |
| [**PERSIST**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#PERSIST) |  |
| [**REFRESH**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#REFRESH) |  |
| [**REMOVE**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#REMOVE) |  |
| [**REPLICATE**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#REPLICATE) |  |
| [**SAVE\_UPDATE**](https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/annotations/CascadeType.html#SAVE_UPDATE) |  |

Barclays:

1) clone of singleton object

2)Lazy intialization

3)coditional operator in side comparator

4)clone and clonable

5)design pattern

6)put and post difference

7)runable and tread extend from same class

8) polimorphism with example

9) Authentication implementation in Rest service

10) sort in desending order based n empyee

11) Singleton and protype in Spring

12) IOC and DI advantage

13) collection ArrayList and Linked list internal implementation advantage and disadvantage

14) method with wrapper class arg and int argmenet

15) anotation means