**每日作业卷答案**

**Spring第1天**

极客营

# 关卡1

## 训练案例

### 训练描述

现在提供了UserServiceImpl和UserDaoImpl实现类，要求使用Spring框架IOC的依赖注入配置文件的方式来管理UserServiceImpl和UserDaoImpl，并且把UserDaoImpl对象UserServiceImpl类中。在demo1测试类中编写测试方法。获取到UserServiceImpl对象，完成方法调用。

### 操作步骤答案

1. 请在applicationContext.xml配置文件中编写UserServiceImpl类的配置文件  
   <bean id="userService" class="com.igeek.service.impl.UserServiceImpl">

<property name="userDao" ref="userDao"/>

</bean>

1. 请在applicationContext.xml配置文件中编写UserDaoImpl类的配置文件  
   <bean id="userDao" class=" com.igeek.dao.impl.UserDaoImpl"> </bean>
2. 把UserDaoImpl注入到UserServiceImpl对象中  
   <bean id="userService" class=" com.igeek.service.impl.UserServiceImpl">

<property name="userDao" ref="userDao"/>

</bean>

1. 在demo1类中完成测试方法，获取到UserServiceImpl对象，调用save方法，在控制台输出结果。  
   @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

UserService us = (UserService) ac.getBean("userService");

us.save();

}

# 关卡2

## 训练案例

### 训练描述

现在提供了User类，该类中存在属性，请使用Spring框架的依赖注入的方式给这些属性赋值，值可以任意。在demo1类的测试方法中从IOC容器中获取到User对象，并且在控制台上打印结果。

### 操作步骤答案

1. 请在applicationContext.xml配置文件中编写User类的配置文件

<bean id="user" class=" com.igeek.domain.User"></bean>

1. 请在applicationContext.xml配置文件中给User类的属性赋值  
   <bean id="user" class=" com.igeek.domain.User">

<property name="name" value="张三" />

<property name="list">

<list>

<value>哈哈</value>

<value>呵呵</value>

</list>

</property>

<property name="pros">

<props>

<prop key="aaa">赵四</prop>

<prop key="bbb">老五</prop>

</props>

</property>

</bean>

1. 在demo1类中完成测试方法，获取到User对象，在控制台输出结果。  
   @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

User u = (User) ac.getBean("user");

System.out.println(u);

}

# 关卡3

## 训练案例1

### 训练描述

使用Spring框架IOC的依赖注入方式，给User类的三个属性赋值，使用ApplicationContext工厂获取到User类的对象，重写User类的toString方法，在控制台上打印结果。

### 操作步骤描述

1. 请在applicationContext.xml配置文件中编写User类的配置文件，并且完成依赖注入  
   <bean id="user" class=" com.igeek.demo1.User">

<property name="name" value="igeek/>

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

<property name="hobby">

<array>

<value>篮球</value>

<value>足球</value>

</array>

</property>

</bean>

1. 请在Demo1类中的run1()测试方法中编写从IOC容器中获取User对象的方法，在控制台打印结果。  
   public class Demo1 {

@Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

User u = (User) ac.getBean("user");

System.out.println(u);

}

}

# 关卡4

## 训练案例1

### 训练描述

### 现在有UserServiceImpl实现类和UserDaoImpl实现类，要求把这个类都使用Spring的IOC进行管理，并且需要把UserDaoImpl注入到UserServiceImpl属性中，在测试方法中获取到UserServiceImpl实现类对象，完成方法的调用

### 操作步骤描述

1. 请在applicationContext.xml配置文件中配置UserServiceImpl和UserDaoImpl，并且需要把dao对象注入到service属性中去，完成方法调用  
   <bean id="userService" class=" com.igeek.service.impl.UserServiceImpl">

<property name="userDao" ref="userDao"/>

</bean>

<bean id="userDao" class=" com.igeek.dao.impl.UserDaoImpl"/>

1. 请在Demo1类中的run1()测试方法中编写从IOC容器中获取UserServiceImpl实现类对象，调用该对象的save方法，结果打印在控制台上  
   @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

UserService us = (UserService) ac.getBean("userService");

us.save();

}

# 关卡5

## 训练案例

### 训练描述

使用Spring和Hibernate框架，把User对象的数据存入到数据库中。提供了hibernate.cfg.xml配置文件，提供了HibernateUtils工具类，提供applicationContext.xml配置文件，提供了User类，提供了UserServiceImpl和UserDaoImpl两个实现类

### 操作步骤描述

1. 先在MySQL数据库中创建名称为spring\_day01的数据库  
   create database spring\_day01;
2. 根据User类编写User.hbm.xml的映射配置文件  
   <?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name=" com.igeek.domain.User" table="t\_user">

<id name="id" column="id">

<generator class="native"/>

</id>

<property name="name" column="name"/>

<property name="age" column="age"/>

</class>

</hibernate-mapping>

1. 在hibernate.cfg.xml配置文件引入User.hbm.xml的映射配置文件  
   <mapping resource="com/igeek/domain/User.hbm.xml"/>
2. 在UserDaoImpl类的save方法编写代码，把User对象存入到数据库中  
   Session session = HibernateUtils.getCurrentSession();

session.save(user);

1. 在UserServiceImpl类的save方法编写代码，开启事务，调用dao保存User对象  
   public void save(User user) {

Session session = HibernateUtils.getCurrentSession();

Transaction tx = session.beginTransaction();

try {

userDao.save(user);

tx.commit();

} catch (Exception e) {

e.printStackTrace();

tx.rollback();

}

}

1. 在applicationContext.xml配置文件中编写UserServiceImpl和UserDaoImpl的IOC配置，并且在service中注入dao对象  
   <bean id="userService" class=" com.igeek.service.impl.UserServiceImpl">

<property name="userDao" ref="userDao"/>

</bean>

<bean id="userDao" class=" com.igeek.dao.impl.UserDaoImpl"/>

1. 在applicationContext.xml配置文件中配置User类，并且给属性注入值  
   <!-- 配置User类 -->

<bean id="user" class=" com.igeek.domain.User">

<!-- 给name属性注入值 -->

<property name="name" value="igeek"/>

<!-- 给age属性注入值 -->

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

</bean>

1. 在Demo1类的测试方法run1方法中，从IOC容器中获取到User对象和Service的对象，调用保存的方法，把User对象数据保存到数据库中  
   @Test

public void run1(){

ApplicationContext ac = new ClassPathXmlApplicationContext("classpath:applicationContext.xml");

UserService us = (UserService) ac.getBean("userService");

User user = (User) ac.getBean("user");

us.save(user);

}