МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНЙ УНІВЕРСИТЕТ

Кафедра програмного забезпечення автоматизованих систем

ЗВІТ

про виконання

Лабораторної роботи №1

з дисципліни "Розробка Web-додатків на Java з допомогою Spring фреймворку"

Перевірив роботу: Виконав роботу:

"\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ студент гр. ПЗС-2244

викладач\_\_\_\_\_\_\_\_\_\_\_\_\_ Брус В.М.

Черкаси 2023

Тема: Spring

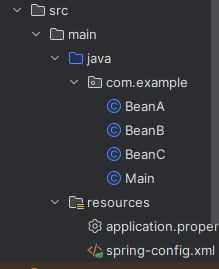
Мета: створити проект у якому реалізувати у самостійно обраній предметній області за допомогою xml нотацій - Spring біни, які повинні мати 3 рівні вкладеності

Завдання:

Створити проект у якому реалізувати у самостійно обраній предметній області за допомогою xml нотацій - Spring біни, які повинні мати 3 рівні вкладеності (тобто один бін повинен містити як поле інший бін і т. д.). У бінах повинен бути реалізований принцип DI за допомогою конструкторів і сеттерів. Також необхідно щоб у Вашому проекті використовувались рядки чи інші значення за допомогою properties файлу.

Хід роботи:

1. Створення файлу properties, класів BeanA, BeanB, BeanC, main, конфігурації spring-config.xml:



1. Результат роботи програми в терміналі:

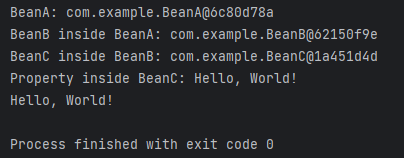


Рисунок 2 – Макет інтерфейсу Player

Лістинг

Клас BeanA.java:

package com.example;  
  
public class BeanA {  
 private BeanB beanB;  
  
 public BeanA(BeanB beanB) {  
 this.beanB = beanB;  
 }  
  
 public void doSomething() {  
 beanB.doSomething();  
 }  
  
 public BeanB getBeanB() { return this.beanB; }  
}

Клас BeanB.java:

package com.example;  
  
public class BeanB {  
 private BeanC beanC;  
  
 public void setBeanC(BeanC beanC) {  
 this.beanC = beanC;  
 }  
 public BeanC getBeanC() {  
 return this.beanC;  
 }  
  
 public void doSomething() {  
 beanC.doSomething();  
 }  
}

Клас BeanC.java:

package com.example;  
  
public class BeanC {  
 private String property;  
  
 public void setProperty(String property) {  
 this.property = property;  
 }  
 public String getProperty() {  
 return this.property;  
 }  
 public void doSomething() {  
 System.*out*.println(property);  
 }  
}

Клас Main.java:

package com.example;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Main {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");  
 BeanA beanA = context.getBean(BeanA.class);  
  
 System.*out*.println("BeanA: " + beanA);  
 System.*out*.println("BeanB inside BeanA: " + beanA.getBeanB());  
 System.*out*.println("BeanC inside BeanB: " + beanA.getBeanB().getBeanC());  
 System.*out*.println("Property inside BeanC: " + beanA.getBeanB().getBeanC().getProperty());  
  
 beanA.doSomething();  
 }  
}

spring-config.xml:

<beans  
 xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd">  
 <context:property-placeholder location="classpath:application.properties"/>  
 <bean id="beanA" class="com.example.BeanA">  
 <constructor-arg ref="beanB"/>  
 </bean>  
  
 <bean id="beanB" class="com.example.BeanB">  
 <property name="beanC" ref="beanC"/>  
 </bean>  
  
 <bean id="beanC" class="com.example.BeanC">  
 <property name="property" value="${my.property}"/>  
 </bean>  
  
</beans>

application.properties:

my.property=Hello, World!

Висновки

У ході виконання даного завдання було створено проект, в якому реалізовано Spring біни у самостійно обраній предметній області за допомогою xml нотацій. Біни мають 3 рівні вкладеності, що демонструє глибоку ієрархію та взаємозв’язок між компонентами системи. Принцип DI було реалізовано за допомогою конструкторів і сеттерів, що забезпечує гнучкість та зручність управління залежностями між бінами. Також у проекті було використано properties файл для збереження рядка.