МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНЙ УНІВЕРСИТЕТ

Кафедра програмного забезпечення автоматизованих систем

ЗВІТ

про виконання

Лабораторної роботи №2

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: Переробити першу лабораторку, додати анотації, створити методи init, destroy

Завдання:

Необхідно переробити створений проект у 1 лабораторній роботі таким чином щоб створення бінів і DI виконувалось за допомогою Spring анотацій (@Component, @Autowired, @Value, @Qualifier і т.д.). Необхідно реалізувати init і destroy методи. Після виконання роботи у проекті не повинно залишитись файлу із xml нотаціями (у якому був опис бінів і т.д.) але повинні зявитись анотації у коді проекту. Проект необхідно розмістити на github і у звіті додати посилання на проект на github-і.

Хід роботи:

1. Видалення файлу spring-config.xml:

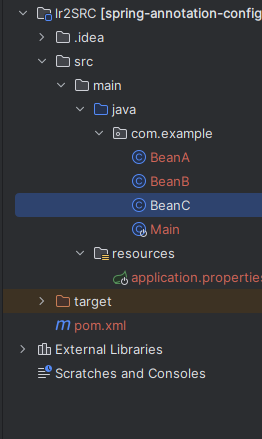


Рисунок 1 – Структура файлів

1. Зміна вмісту файлів, додавання анотацій, додавання методів init,destroy, тощо.
2. Результат роботи програми в терміналі:



Рисунок 2 – Вивід в термінал

Лістинг

Клас BeanA.java:

package com.example;

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

import org.springframework.stereotype.Component;

@Component

public class BeanA {

private final BeanB beanB;

@Autowired

public BeanA(BeanB beanB) {

this.beanB = beanB;

}

public void doSomething() {

beanB.doSomething();

}

}

Клас BeanB.java:

package com.example;

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

import org.springframework.stereotype.Component;

@Component

public class BeanB {

private BeanC beanC;

@Autowired

public void setBeanC(BeanC beanC) {

this.beanC = beanC;

}

public void doSomething() {

beanC.doSomething();

}

}

Клас BeanC.java:

package com.example;

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

import org.springframework.stereotype.Component;

import javax.annotation.PostConstruct;

import javax.annotation.PreDestroy;

@Component

public class BeanC {

@Value("${my.property}")

private String property;

@PostConstruct

public void init() {

System.out.println("BeanC is initialized with property: " + property);

}

@PreDestroy

public void destroy() {

System.out.println("BeanC is being destroyed");

}

public void doSomething() {

System.out.println(property);

}

}

Клас Main.java:

package com.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

@SpringBootApplication

public class Main {

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(Main.class, args);

BeanA beanA = context.getBean(BeanA.class);

beanA.doSomething();

}

}

application.properties:

my.property=Hello, World!

Висновки

В ході виконання цієї роботи було створено проект Spring з використанням XML конфігурації та анотацій. Було реалізовано Dependency Injection (DI) через конструктори та сеттери, а також використання значень з файлу properties. Були додані методи ініціалізації та знищення для бінів.