МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК

Лабораторна робота №4

з дисципліни «ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ»

Тема: «Основи Dependency Injection и Spring Framework»

Виконала:
студентка групи КС-33
Рузудженк С.Р.
Перевірив:
ст. викл. Товстокоренко О.Ю.

Харків – 2019

Лабораторна робота №4

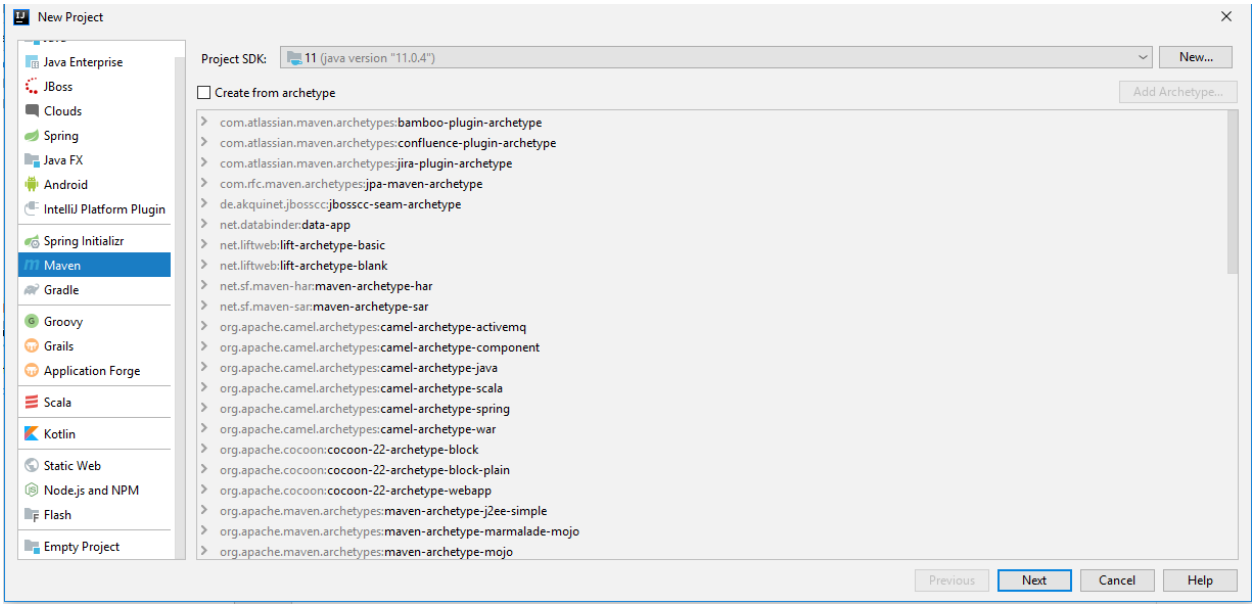
Тема: Основи Dependency Injection и Spring Framework

Мета: ознайомитися із парадигмою IoC та її імплементацією фреймворку Spring завдяки Dependency Injection

Хід роботи

У ході виконання даної лабораторної роботи було створено Spring-проект за допомогою Maven для конфігурації бібліотек, а також проведено ознайомлення з парадигмою IoC та її імплементацією фреймворку Spring завдяки Dependency Injection.

1. Створимо новий Maven-проект, прописуємо GroupID та ArtifactID.

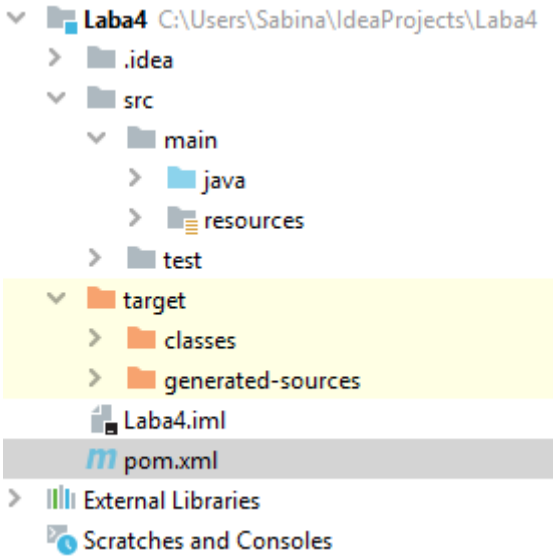


2. До автоматично згенерованого файлу ***pom.xml*** додаємо залежності для використання ***org.springframework***:

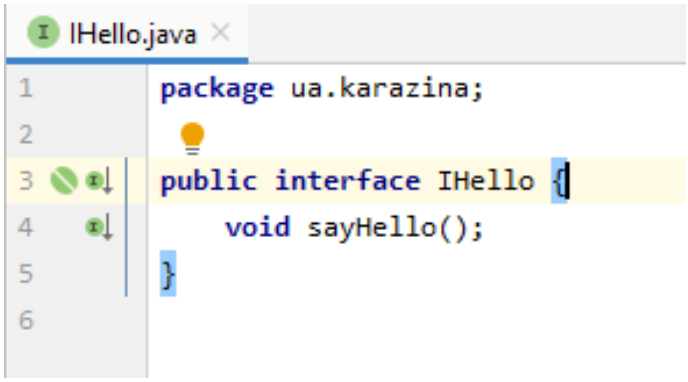
```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>3.0.0.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
```

```
        <artifactId>spring-beans</artifactId>
        <version>3.0.0.RELEASE</version>
    </dependency>
</dependencies>
```

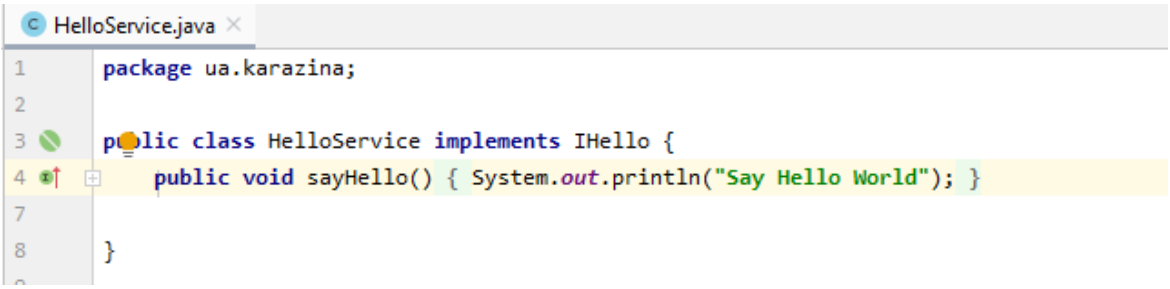
Таким чином, отримано готовий проект з налаштованими залежностями.



3. Створимо інтерфейс IHello.



4. Створимо простий класcс-сервіс, який друку’ «Hello World» і реалізує інтерфейс IHello.



5. Для реєстрації оголошеного компоненту в контейнері Spring використаємо XML, де всередину елементу <beans> вміщуємо повний опис конфігурації Spring, включаючи оголошення <bean>.

ApplicationResources.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 name="hello" class="ua.karazina.HelloService"></bean>
</beans>
```

6. Для тестування створюємо простий клас Executor, у якому створюємо новий об'єкт ApplicationContext, якому передаємо створений раніше XML-файл, у якому зареєстровано оголошений компонент у контейнері Spring.

Executor.java

```
package ua.karazina;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Executor {

    public static void main(String[] args) {
        ApplicationContext ctx = new ClassPathXmlApplicationContext(
                configLocation: "ApplicationResources.xml");
        IHello hello = (IHello)ctx.getBean( s: "hello");
        hello.sayHello();
    }
}
```

7. Збираємо та запускаємо проект.

```
Run: Executor
"C:\Program Files\JetBrains\IntelliJ IDEA 2019.2.4\jbr\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.2.4\lib\idea_rt.jar=58378:C:\Program Files\JetBrains
нов. 18, 2019 12:42:17 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@2437c6dc: startup date [Mon Nov 18 12:42:17 EET 2019]; root of context hierarchy
нов. 18, 2019 12:42:18 PM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
INFO: Loading XML bean definitions from class path resource [ApplicationResources.xml]
нов. 18, 2019 12:42:18 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@727803de: defining beans [hello]; root of factory hierarchy
Say Hello World
```

Висновки

У ході виконання даної лабораторної роботи було створено Spring-проект за допомогою Maven для конфігурації бібліотек, а також проведено ознайомлення з парадигмою IoC та її імплементацією фреймворку Spring завдяки Dependency Injection, створено інтерфейс та клас, що його реалізує, зареєстровано компонент у контейнері Spring та створено клас для тестування. Даний проект було вдало зібрано та запущено.