ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра компьютерных технологий

Лабораторная работа №2

по предмету «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

ТЕМА: «Inversion of Control и Dependency Injection с помощью аннотаций и кода»

Выполнил:

студент 4 курса ИВТ

Группа ИВТ-1

Очная форма обучения

Мамонтов Владислав Евгеньевич

Проверил:

старший преподаватель

Лихолетов Александр Валерьевич

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Донецк-2021

Листинг Employee

@Component("employeeBean")  
@Scope("prototype")  
public class Employee {  
  
 private IPet pet;  
 private IVehicle vehicle;  
  
 @Value("${employee.name}")  
 private String name;  
  
 @Value("${employee.age}")  
 private int age;  
  
 @Autowired  
 public Employee(@Qualifier("dogBean") IPet pet,  
 @Qualifier("carBean") IVehicle vehicle) {  
 this.pet=pet;  
 this.vehicle=vehicle;  
 System.out.println("Employee bean was created");  
 }  
  
 @PostConstruct  
 public void init(){  
 System.out.println("Class Employee: init method");  
 }  
  
 @PreDestroy  
 public void destroy(){  
 System.out.println("Class Employee: destroy method");  
 }  
  
// @Autowired  
// @Qualifier("carBean")  
 public void setVehicle(IVehicle vehicle) {  
 this.vehicle = vehicle;  
 System.out.println("Class Employee: set Vehicle");  
 }  
  
// @Autowired  
// @Qualifier("dogBean")  
 public void setPet(IPet pet) {  
 this.pet = pet;  
 System.out.println("Class Employee: set Pet");  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 System.out.println("Class Employee: set Age");  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 System.out.println("Class Employee: set Name");  
 }  
  
 public IVehicle getVehicle() {return vehicle;}  
 public IPet getPet() {return pet;}  
 public int getAge() {return age;}  
 public String getName() {return name;}  
  
 public void StartVehicle() {  
 vehicle.GetStarted();  
 }  
  
 public void callYourPet() {  
 System.out.println("Hey!");  
 pet.say();  
 }  
}

Листинг Dog

@Component("dogBean")  
@Scope("prototype")  
public class Dog implements IPet {  
  
 public Dog(){  
 System.*out*.println("Dog bean was created");  
 }  
  
 @Override  
 public void say() {  
 System.*out*.println("гав");  
 }  
  
 @Override  
 @PostConstruct  
 public void init() {  
 System.*out*.println("Class Dog: init method");  
 }  
  
 @Override  
 @PreDestroy  
 public void destroy() {  
 System.*out*.println("Class Dog: destroy method");  
 }  
}

Листинг Car

@Component("carBean")  
@Scope("prototype")  
public class Car implements IVehicle{  
  
 public Car(){  
 System.*out*.println("Car bean was created");  
 }  
  
 @Override  
 public void GetStarted() {  
 System.*out*.println("г-р-р-р-р");  
 }  
  
 @Override  
 @PostConstruct  
 public void init() {  
 System.*out*.println("Class Car: init method");  
 }  
  
 @Override  
 @PreDestroy  
 public void destroy() {  
 System.*out*.println("Class Car: destroy method");  
 }  
}

Листинг applicationContext.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"  
 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="application.properties"/>  
 <context:component-scan base-package="laboratory\_work.laboratory\_oop\_2"/>  
  
</beans>

Листинг application.property

employee.age=20  
employee.name=Vlad

Листинг Task\_2WithoutXMLFile

public class Task\_2 {  
 public static void main(String[] args) {  
 AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(MyConfig.class);  
  
 Employee employee = context.getBean("employeeBean", Employee.class);  
 employee.callYourPet();  
 employee.StartVehicle();  
 System.*out*.println(employee.getName());  
 System.*out*.println(employee.getAge());  
  
 Employee employee1 = context.getBean("employeeBean", Employee.class);  
 employee1.setName("VL");  
 employee1.setAge(50);  
 employee1.callYourPet();  
 employee1.StartVehicle();  
 System.*out*.println(employee1.getName());  
 System.*out*.println(employee1.getAge());  
  
 context.close();  
 }  
}

Листинг Task\_2WithXMLFile

public class Task\_2WithXMLFile {  
 public static void main(String[] args) {  
 ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 Employee employee = context.getBean("employeeBean", Employee.class);  
 employee.callYourPet();  
 employee.StartVehicle();  
 System.*out*.println(employee.getName());  
 System.*out*.println(employee.getAge());  
  
 Employee employee1 = context.getBean("employeeBean", Employee.class);  
 employee1.setName("VL");  
 employee1.setAge(50);  
 employee1.callYourPet();  
 employee1.StartVehicle();  
 System.*out*.println(employee1.getName());  
 System.*out*.println(employee1.getAge());  
  
 context.close();  
 }  
}

Листинг Task\_3

public class Task\_3 {  
 public static void main(String[] args) {  
 AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(MyConfigWithoutScanning.class);  
  
 Employee employee = context.getBean("employeeBean", Employee.class);  
 employee.callYourPet();  
 employee.StartVehicle();  
 System.*out*.println(employee.getName());  
 System.*out*.println(employee.getAge());  
  
 Employee employee1 = context.getBean("employeeBean", Employee.class);  
 employee1.setName("VL");  
 employee1.setAge(50);  
 employee1.callYourPet();  
 employee1.StartVehicle();  
 System.*out*.println(employee1.getName());  
 System.*out*.println(employee1.getAge());  
  
 context.close();  
 }  
}

Листинг MyConfig

@Configuration  
@ComponentScan("laboratory\_work.laboratory\_oop\_2")  
@PropertySource("classpath:application.properties")  
public class MyConfig {  
}

Листинг MyConfigWithoutScanning

@Configuration  
@PropertySource("classpath:application.properties")  
public class MyConfigWithoutScanning {  
  
 @Bean  
 @Scope("prototype")  
 public IPet dogBean(){  
 return new Dog();  
 }  
  
 @Bean  
 @Scope("prototype")  
 public IVehicle carBean(){  
 return new Car();  
 }  
  
 @Bean  
 @Scope("prototype")  
 public Employee employeeBean(){  
 return new Employee(dogBean(), carBean());  
 }  
}