ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра компьютерных технологий

Лабораторная работа №1

по предмету «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

ТЕМА: «Создание проекта. Inversion of Control и Dependency Injection»

Выполнил:

студент 4 курса ИВТ

Группа ИВТ-1

Очная форма обучения

Мамонтов Владислав Евгеньевич

Проверил:

старший преподаватель

Лихолетов Александр Валерьевич

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

Донецк-2021

Листинг LaboratoryOop1Application

package laboratory\_work.laboratory\_oop\_1;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
  
@SpringBootApplication  
public class LaboratoryOop1Application {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(LaboratoryOop1Application.class, args);  
 }  
  
 @GetMapping("/hello")  
 public String Hello(@RequestParam(value = "myName", defaultValue = "World") String name) {  
 return String.*format*("Hello %s", name);  
 }  
  
 @GetMapping("/about")  
 public String About() {  
 return "about us";  
 }  
  
 @GetMapping("/options")  
 public String Options(@RequestParam(value = "myOption", defaultValue = "") String option) {  
 if (option.equals("")) {  
 return "option";  
 } else {  
 return "not a option";  
 }  
 }  
}

Листинг Employee

public class Employee {

private IPet pet;

private ICar car;

private String name;

private int age;

public Employee() {}

public void setCar(ICar car) {this.car = car;}

public void setPet(IPet pet) {this.pet = pet;}

public void setAge(int age) {this.age = age;}

public void setName(String name) {this.name = name;}

public ICar getCar() {return car;}

public IPet getPet() {return pet;}

public int getAge() {return age;}

public String getName() {return name;}

public void StartCar() {

car.GetStarted();

}

public void callYourPet() {

System.out.println("Hey!");

pet.say();

}

}

Листинг Task\_2

public class Task\_2 {

public static void main(String[] args) {

ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

Employee employee = context.getBean("myEmployee", Employee.class);

employee.callYourPet();

employee.StartCar();

System.out.println(employee.getAge());

System.out.println(employee.getName());

context.close();

}

}

Листинг Car

public class Car implements ICar{

@Override

public void GetStarted() {

System.out.println("г-р-р-р-р");

}

}

Листинг Dog

public class Dog implements IPet {

@Override

public void say() {

System.out.println("гав");

}

}

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"/>

<bean id="myPet" class="laboratory\_work.laboratory\_oop\_1.Dog"></bean>

<bean id="myCar" class="laboratory\_work.laboratory\_oop\_1.Car"></bean>

<bean id="myEmployee" class="laboratory\_work.laboratory\_oop\_1.Employee">

<property name="pet" ref="myPet"/>

<property name="car" ref="myCar"/>

<property name="name" value="${employee.name}"/>

<property name="age" value="${employee.age}"/>

</bean>

</beans>

Листинг application.property

employee.age=20

employee.name=Vlad