

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

Spring, работа с аннотациями

тема

Преподаватель

\_\_\_\_\_  
дата, подпись

А.С. Черниговский

\_\_\_\_\_  
инициалы, фамилия

Студент

КИ18-16Б

\_\_\_\_\_  
номер группы, зачётной книжки

\_\_\_\_\_  
дата, подпись

К.И. Оглезнев

\_\_\_\_\_  
инициалы, фамилия

Красноярск 2021

## 1 Цель работы

Познакомиться с аннотациями в Spring.

## 2 Задачи работы

Взять практическое задание №2 и сконфигурировать контекст приложения без использования xml, а только при помощи аннотаций и java-файла конфигурации. Снабдить ранее спроектированные классы init и destroy методами, а также использовать фабричный метод для любого из классов.

## 3 Ход работы

```
@Configuration
public class ConfigurationGarden {

    @Bean(value = "frontGarden")
    public FrontGarden frontGarden() { return new FrontGarden( length: 11, width: 11, lawn: true); }

    @Bean(value = "backGarden")
    public BackGarden backGarden() { return new BackGarden( length: 10, width: 10); }

    @Bean(value = "house1")
    public House house1() { return new House( name: "house1", frontGarden()); }

    @Bean(value = "house2", initMethod = "turnOn", destroyMethod = "turnOff")
    public House house2() { return new House( name: "house2", backGarden()); }

    @Bean(value = "houseFactory", initMethod = "turnOn", destroyMethod = "turnOff")
    public House houseFactory() { return House.houseFactory(); }

}
```

Рисунок 1 – java-файл конфигурации

```

1  package ru.myapp.garden;
2
3  import lombok.Data;
4  import org.springframework.beans.factory.annotation.Qualifier;
5
6
7  @Data
8  public class House {
9
10     private final Garden garden;
11     private final String name;
12
13     public House(String name, @Qualifier("frontGarden") Garden garden)
14     {
15         this.garden = garden;
16         this.name = name;
17     }
18
19     @ public static House houseFactory(){
20         return new House( name: "Test", new BackGarden( length: 1, width: 1));
21     }
22
23     public double getSquare()
24     {
25         return garden.getLength() * garden.getWidth();
26     }
27
28     public void turnOn()
29     {
30         System.out.println("Создание " + this.name);
31     }
32
33     public void turnOff()
34     {
35         System.out.println("Разрушение " + this.name);
36     }
37 }

```

Рисунок 2 – Класс House

```

public class TestSpring {
    public static void main(String[] args){

        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(ConfigurationGarden.class);

        House house1 = context.getBean( ConfigurationGarden.class).house1();
        House house2 = context.getBean( ConfigurationGarden.class).house2();

        System.out.println("Площадь сада " + house1.getName() + ": " + house1.getSquare());
        System.out.println("Площадь сада " + house2.getName() + ": " + house2.getSquare());

        house1 = context.getBean( name: "house1", House.class);
        house2 = context.getBean( name: "house2", House.class);

        System.out.println("Площадь сада " + house1.getName() + ": " + house1.getSquare());
        System.out.println("Площадь сада " + house2.getName() + ": " + house2.getSquare());

        House houseFactory = context.getBean( name: "houseFactory", House.class);

        System.out.println("Площадь сада " + houseFactory.getName() + ": " + houseFactory.getSquare());

        context.close();
    }
}

```

Рисунок 3 – main

```
C:\Users\kiry_\.jdk\openjdk-15\bin\java.exe ...  
Создание house2  
Создание Test  
Площадь сада house1: 121.0  
Площадь сада house2: 100.0  
Площадь сада house1: 121.0  
Площадь сада house2: 100.0  
Площадь сада Test: 1.0  
Разрушение Test  
Разрушение house2  
  
Process finished with exit code 0
```

Рисунок 4 – Результат выполнения

#### 4 Вывод

В ходе данной лабораторной работы был изучен механизм внедрения зависимостей в Spring.