

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

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Внедрение зависимостей в Spring

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

Ознакомиться с механизмом внедрения зависимостей в Spring.

## 2 Задачи работы

В каждом варианте есть сущность (класс), необходимо создать интерфейс (самостоятельно на усмотрение студента) и классы его имплементирующие. Объекты классов имплементирующих данный интерфейс будут передаваться в качестве зависимостей. Выполнить связывание и получить объекты из контекста. В случае затруднений обратитесь к конспекту второй лекции курса. Необходимо:

- 1) Реализовать внедрение простых значений через конструктор;
- 2) Реализовать внедрение зависимости по ссылке через конструктор;
- 3) Интерфейс должен содержать как минимум один метод;
- 4) Классы, имплементирующие интерфейс должны содержать как минимум одно поле (у разных классов разные);
- 5) Зависимый класс должен содержать метод, который бы на основе вызова метода у зависимости выводил бы некоторое сообщение в консоль (Например для класса Автомобиля, в который внедряются Двигатели. Двигатели могут выдавать свою мощность, а автомобиль может выводить сообщение с какой скоростью он может двигаться.)
- 6) Реализовать внедрение простых значений из внешнего файла через Setter (Вариант 20 - Сад)

## 3 Ход работы

A screenshot of a code editor with a dark theme. The code is written in Java and defines a package and an interface. The package is 'ru.myapplication'. The interface is named 'Garden' and is public. It contains two methods: 'getLength()' and 'getWidth()', both returning a 'double' value. The code is displayed on a grid with line numbers 1 through 7 on the left side.

```
1 package ru.myapplication;
2
3 public interface Garden
4 {
5     double getLength();
6     double getWidth();
7 }
```

Рисунок 1 – Интерфейс

```

1      package ru.myapp.garden;
2
3      public class House {
4
5          private final Garden garden;
6
7          public House(Garden garden) {
8              this.garden = garden;
9          }
10
11         public Garden getGarden()
12         {
13             return this.garden;
14         }
15
16         public double getSquare()
17         {
18             return garden.getLength() * garden.getWidth();
19         }
20     }
21

```

Рисунок 2 – Класс Garden

```

1      package ru.myapp.garden;
2
3      public class BackGarden implements Garden{
4
5          private final double length;
6          private final double width;
7
8          public BackGarden(double length, double width){
9              this.length = length;
10             this.width = width;
11         }
12
13         @Override
14         public double getLength() {
15             return length;
16         }
17
18         @Override
19         public double getWidth() {
20             return width;
21         }
22     }
23

```

Рисунок 3 – Класс BackGarden

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:context="http://www.springframework.org/schema/context"
5      xsi:schemaLocation="http://www.springframework.org/schema/beans
6          http://www.springframework.org/schema/beans/spring-beans.xsd
7          http://www.springframework.org/schema/context
8          http://www.springframework.org/schema/context/spring-context.xsd">
9
10     <bean id="frontGarden" class="ru.myapp.garden.FrontGarden">
11         <constructor-arg name="length" value="11"/>
12         <constructor-arg name="width" value="11"/>
13         <constructor-arg name="lawn" value="${lawn}"/>
14     </bean>
15
16     <context:property-placeholder location="classpath:application.properties"/>
17     <bean id="backGarden" class="ru.myapp.garden.BackGarden">
18         <constructor-arg name="length" value="10"/>
19         <constructor-arg name="width" value="10"/>
20     </bean>
21
22     <bean id="house1" class="ru.myapp.garden.House">
23         <constructor-arg name="garden" ref="frontGarden"/>
24     </bean>
25
26     <bean id="house2" class="ru.myapp.garden.House">
27         <constructor-arg name="garden" ref="backGarden"/>
28     </bean>
29
30 </beans>

```

Рисунок 4 – Внедрение зависимостей

```

Площадь сада house1: 121.0
Площадь сада house2: 100.0

```

Рисунок 5 – Результат выполнения

## 4 Вывод

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