

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**  
**(СПбГУТ)**

---

Институт магистратуры  
Кафедра информационных управляющих систем  
«Создание объектов при помощи Spring»  
Дисциплина «Аспектно-ориентированное программирование»

Отчет  
к лабораторной работе № 3

Выполнил:  
Студент группы ИСТ-012м  
Тарасенко А.А.

Принял:  
Ст. преподаватель Антонов В.В.

Санкт-Петербург  
2020

Задание: изучить методы создания объектов в Spring при помощи beans.

Ход работы:

Spring позволяет разработчику осуществлять управление зависимостями bean-компонентов с помощью конфигурирования на основе XML путем использования XML-файла контекста приложения. Этот файл является внешним для приложения, и в нем содержатся определения bean-компонентов и их зависимостей для этого приложения.

Рассмотрим создание объектов на примере проекта.

```
public class Bird {
    public Head head;
    public Body body;
    public Legs legs;

    public Bird() {}

    public Bird(Head head, Body body, Legs legs)
    {
        this.head = head;
        this.body = body;
        this.legs = legs;
    }

    public void sound()
    {
        head.sound();
    }

    public void isFlying()
    {
        float weight = head.getWeight() + body.getWeight() +
        legs.getWeight();
        if (weight <= body.WingsForce())
            System.out.println("My " + body.wingsName() + "
        wings are great, so I can aviate!");
        else
            System.out.println("My " + body.wingsName() + "
        wings are small, so I'd better stroll.");
    }

    public void isSwimming()
    {
        if (legs.canSwim())
```

```

        System.out.println("My legs are wide, so I can
glide.");
    else
        System.out.println("Me legs so slim, I can not
swim.");
    }
}

```

В классе Bird объявлены поля head, legs и body, тип которых является интерфейсом, то есть они могут быть объектами разных классов.

В main происходит обращение к XML-файлу с описанием bean-ов, преобразование в объект класса Bird и обращение к методам класса.

```

public static void main(String[] args) {
    SpringApplication.run(Lr3Application.class, args);

    //ApplicationContext ctx = new
AnnotationConfigApplicationContext(AppContext_old.class);
    ApplicationContext ctx = new
ClassPathXmlApplicationContext("file:src/main/java/SUT/IST012m/T
arasenko/LR3/AppContext.xml");
    Bird bird = ctx.getBean(Bird.class);
    bird.sound();
    bird.isFlying();
    bird.isSwimming();
}

```

XML-файл выглядит следующим образом:

```

<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.xsd">

    <bean id="head"
class="SUT.IST012m.Tarasenko.LR3.Heads.PigeonHead"></bean>
    <bean id="body"
class="SUT.IST012m.Tarasenko.LR3.Bodies.GooseBody"></bean>
    <bean id="legs"
class="SUT.IST012m.Tarasenko.LR3.Legs.PigeonLegs"></bean>

    <bean id="bird" class="SUT.IST012m.Tarasenko.LR3.Bird">
        <constructor-arg index="0" ref="head"/>
        <constructor-arg index="1" ref="body"/>
        <constructor-arg index="2" ref="legs"/>
        <!--<property name="head" value="GooseHead"/>-->
    </bean>
</beans>

```

В первом bean указывается класс для объекта головы, во втором – для тела, в третьем – для ног; в четвёртом же создаётся объект типа Bird при этом в конструктор передаются объекты, созданные раньше.

```
2020-10-09 18:58:23.060 INFO 3444 --- [
2020-10-09 18:58:23.062 INFO 3444 --- [
2020-10-09 18:58:23.590 INFO 3444 --- [
Quack-quack
My pigeon wings are small, so I'd better stroll.
My legs are wide, so I can glide.

Process finished with exit code 0
```

Рисунок 1 – запуск проекта с первым набором параметров

```
2020-10-09 19:00:01.412 INFO 6564 --- [
2020-10-09 19:00:01.415 INFO 6564 --- [
2020-10-09 19:00:01.931 INFO 6564 --- [
Coo-coo
My pigeon wings are great, so I can aviate!
Me legs so slim, I can not swim.

Process finished with exit code 0
```

Рисунок 2 – запуск проекта со вторым набором параметров