SPRING

Spring – это фреймворк, предназначенный для более быстрого и простого построения приложений на Java

**Фреймворк** – это платформа, которая определяет структуру приложения.

Появился в 2004 году, но изначально не был так популярен. На 2021 год у него уже 5-я версия (вышла в 2017), и он является очень популярным (если не самым популярным вообще) фреймворком.

Оф сайт – spring.io

Среди проектов Spring (которые сами по себе тоже фреймворки) есть

Spring Boot – очень быстро можно развернуть приложение

Spring Data – непосредственно для работы с б/д

Spring Security – для безопасности приложений

Spring for Android – для написание андроид-приложений

и др.

+ docs.spring.io – сайт с документацией по Spring

При создании проекта необходимо проверить, чтобы была создана **папка resources** (в самой корневой папке проекта, на одном уровне с src) и чтобы она была Mark Directory as -> Resources root

Если есть **папка java** (синяя), - если через maven создаем - то в ней все классы хранятся, которые мы создаем. И их имя начинается со след пакета, после java (обычно там промежуточные пакеты есть).

В **папке lib** – jarы с подгруженными компонентами приложения

Inversion of Control (IoC)

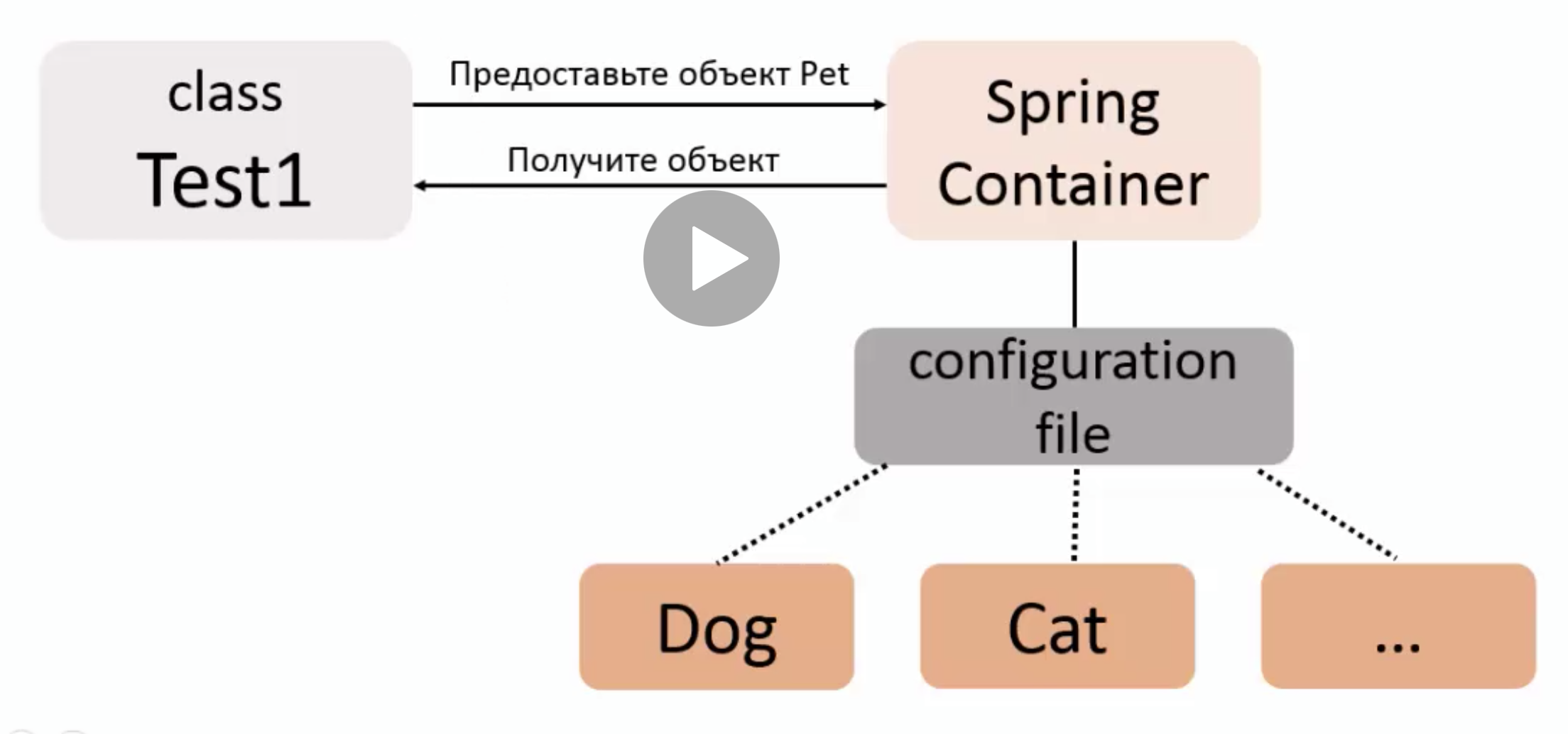
**IoC** – это аутсорсинг создания и управления объектами. Т.е. передача программистом прав на создание объектов Spring’у.

Делается это через **Spring Container**, основные функции которого:

1. *Создание и управление объектами через* ***IoC***
2. *Внедрение зависимостей через* ***Dependency Injection***

**Способы конфигурации Spring Container:**

1. XML-file
2. XML-файл + аннотации
3. Java code



**Конфигурационный файл** нужен для того, чтобы Спринг Контейнер понимал, какие объекты ему создавать.

**Через XML – файл:**

Container прочитает конф-файл в формате XML, создаст объекты, поле чего создастся сам Container, из которого будут получены объекты

Конфигурац файл обычно называется applicationContext.xml (как и класс контенера, создание которого описано ниже) и лежит в папке resourses. Но можно как хочешь называть

Его шапка обычно в таком виде:

<?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

http://www.springframework.org/schema/context/spring-context.xsd">

**\***

</beans>

**\*** Тут как раз мы и создаем бин нашего объекта:

<bean id = “myPet” //- по этому айди мы будем его создавать и вызывать (ИДЕНТИФИКАТОР БИНА)

class = “com.joss.Dog”> // - это класс, который написали и ассоциируем с бином (ЕГО ПОЛНОЕ ИМЯ ОТ СИНЕЙ ПАПКИ)

</bean>

**Spring Bean** - это объект, который создается и управляется через Spring Container

class в bean id – это непосредственно имя класса, объект которого будет содаваться.

Теперь при создании объекта не нужно использовать new, тк объекты уже якобы созданы Spring Container’ом, и их надо только вызвать

**ApplicationContext** – представляет собой Spring Container. Поэтому для создания контейнера, нужно создать ApplicationContext:

ClassPathXmlApplicationContext context = new ClassPathApplicationContext (“applicationContext.xml”);

Вот так он и создается – уже в классе .java *c main?*

Теперь можно создавать объект класса через наш бин:

Pet pet = context.getBean(“myPet”, Pet.class);

– в параметрах указываем id и тип объекта (можно Dog, тк Pet – это интерфейс, который имплементит Pet)

pet.say(); - теперь можно работать с ним как с объектом, например, вызывая метод

context.close(); - обязательно нужно закрывать Контейнер



Кажется, что это наоборот более громоздкое создание объектов, но потом будет понятно, что так лучше