**Spring**

Spring Framework представляет собой просто контейнер внедрения зависимостей, с несколькими удобными слоями (например: доступ к базе данных, прокси, аспектно-ориентированное программирование, RPC, веб-инфраструктура MVC). Это все позволяет вам быстрее и удобнее создавать Java-приложения.

Основные принципы

IoC – inversion of control архитектурный подход, при котором сущность не сама создаёт свои зависимости, а когда этой сущности они предоставляются из вне.

* Зависимости очень часто появляются в коде(Один класс зависит от другого)
  + Класс Cat заточен под класс Барсик –> интерфейсы
  + Объекты создаются вручную –> выносим создание в конфигурационный файл
  + Класс Cat сам создает свои зависимости –> Inversion of control
  + Эти зависимости необходимо где-то создавать -> Dependency Injection

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

🡪

Изображение выглядит как текст

Автоматически созданное описаниеDI - Dependency Injection

* XML



Конфигурация Spring

* XML файл
* Аннотации + XML
* Вся конфигурация в коде

Spring Beans

Scope

* Singleton – гарантирует что будет создан 1 экземпляр бина работаем с 1 ссылкой – по умолчанию
* Prototype – создает новый объект при вызове метода getBean
* Request
* Session
* Global – session

Жизненный цикл бина

* Запускается Spring контейнер
* Создается объект бина
* В бин внедряется зависимость
* Вызов метода init-method
* Use
* Destroy method

Init-method – нужен для инициализации бина, подключения к бд … Destroy method – отключится от бд… , оба метода должны находиться внутри класса бина.

Аннотации

* @Autowired – Spring сам ищет нужный бин и автоматически внедряет его
* @Component – Объявление бина
* @Qualifier – указываем id бина, который хотим внедрить