

Dependency Injection

Dependency Injection – принцип, що зменшує складність програм для великих проектів

Конфігурація в XML

- Beans описуються в XML файлах
- Вихідний код класів залишається незмінним
- Можна застосовувати такий підхід і у випадку, коли у вас немає доступу до вихідного коду

XML конфігурація – елементи

<bean id="" class="">

<property name="" ref=""/>

<constructor-arg name="" ref=""/>

</bean>

Annotation-based конфігурація

- Зв'язки описуються анотаціями у вихідному коді
- Якщо у вас немає доступу до вихідного коду – ви не зможете застосувати цей підхід

Annotation - елементи

- `@Configuration` – конфігураційний клас для Spring
- `@Bean` – так позначається Bean. Аналог `<bean>`
- `@Autowired` – позначається конструктор

Задача

Розробити сервіс, який друкуватиме випадкове ім'я. Передбачити локалізацію – щоб сервіс міг друкувати як українські, так і англійські імена

План вирішення задачі - класи

- Створити інтерфейс для друку імені - **PrintNameBean**
- Створити два класи, що друкують ім'я на українській, та на іноземній мові – **UkraineNameBean, EnglishNameBean**
- Створити клас, який приймає **PrintNameBean** в конструкторі, та має метод **printName()**

План вирішення задачі - XML

- Зв'язати класи через xml. Використайте різні XML файли для Bean та Service. Частина коду

```
ApplicationContext context = new  
ClassPathXmlApplicationContext("file2.xml",  
"file2.xml");
```

- Створити клас XmlRunner, де проініціалізувати Spring, отримати екземпляр NameService, та вивести ім'я

План вирішення задачі - анотації

- Створити конфігураційний клас (анотація **@Configuration**), де зробити аналогічні налаштування як у XML
- Створити клас **AnnotationRunner**, де проініціалізувати конфігурацію. Клас - **AnnotationConfigApplicationContext**
- Отримати екземпляр **NameService**, вивести ім'я

Додаткові моменти

- Якщо є декілька кандидатів для injection, потрібний необхідно позначити як **@Primary** (**primary="true"** в XML)
- Деякі Beans потрібно додавати вручну (наприклад, коли лише один конструктор з параметрами) – `ctx.regiser(BeanClass.class)`

Рефакторинг

- Ми створили два класи – **UkrainianNameBean, EnglishNameBean.**
Код обох класів схожий, як можна спростити його. Підказка – можна ввести ще один клас.