

Annotation-based configuration and Spring Dependency Injection

1. Підготувати різні POJO класи з анотацією `@Component`, які розміщені в 3х різних пекіджах:
 - a. `beans1` (біни: `BeanA`, `BeanB`, `BeanC`),
 - b. `beans2` (біни: `RoseFlower`, `CatAnimal`, `NarcissusFlower`),
 - c. `beans3` (біни: `BeanD`, `BeanE`, `BeanF`).
2. Створити 2ва файли конфігурації `@Configuration`.
3. Перша `@Configuration` проводить скан компонентів в пекіджі `beans1`.
4. Друга `@Configuration` сканує усі квітки з `beans2`, а з `beans3` вибирає лише `BeanD` та `BeanF`.
5. В окремому пекіджу створити біни `OtherBeanA`, `OtherBeanB`, `OtherBeanC` з анотацією `@Component`.
6. Створити біни, які будуть вприскувати ці біни за допомогою `@Autowired` лише в:
 - a. конструктор
 - b. сеттери
 - c. поляДеякі з них вприскувати з використанням `@Qualifier`.
7. Створити декілька бінів, які реалізують певний інтерфейс. Створити бін, що буде містити колекцію для створених бінів, які є впорядковані з використанням `@Order`.
8. Для п.7 один з бінів позначити `@Primary`, і вприснути ці біни в новий бін як об'єкти цього інтерфейсу, один з них як основний, а решта по `@Qualifier`.
9. Для п.5 для бінів додати `@Scope("singleton")` та `@Scope("prototype")` та перевірити як вони вприскуються (вивести хеш-коди для їхніх ссилки).
10. Виконати різні конфігурації для 2х різних профайлів:
 - a. Для бінів з використанням `@Bean`
 - b. Для бінів з використанням `@Component`