@Component @Autowired

컴포넌트 스캔과 의존관계 자동 주입 시작하기

- 지금까지 스프링 빈을 등록할 때는 자바 코드의 @Bean이나 XML의 <bean> 등을 통해서 설정 정보에 직접 등록할 스프링 빈을 나열했다.
- 예제에서는 몇개가 안되었지만, 이렇게 등록해야 할 스프링 빈이 수십, 수백개가 되면 일일이 등록하기도 귀찮고, 설정 정보도 커지고, 누락하는 문제도 발생한다. 역시 개발자는 반복을 싫어한다.(무엇보다 귀찮다 ㅠㅠ)

wow.. oDo

- 그래서 스프링은 설정 정보가 없어도 자동으로 스프링 빈을 등록하는 컴포넌트 스캔이라는 기능을 제공한다.
- 또 의존관계도 자동으로 주입하는 `@Autowired` 라는 기능도 제공한다.

AutoAppConfig.java

```
package hello.core;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.FilterType;

import static org.springframework.context.annotation.ComponentScan.*;

@Configuration
@ComponentScan(
        excludeFilters = @Filter(type = FilterType.ANNOTATION, classes =
Configuration.class))
public class AutoAppConfig {

}
```

이름대로 @Component 를 찾아다님

끝. 안에 @Bean 코드 없음

- 컴포넌트 스캔을 사용하려면 먼저 `@ComponentScan` 을 설정 정보에 붙여주면 된다.
- 기존의 AppConfig와는 다르게 @Bean으로 등록한 클래스가 하나도 없다!

RateDiscountPolicy @Component 추가

`@Component` 스프링빈으로 등록하고 싶은 곳에 붙여주기

```
public class RateDiscountPolicy implements DiscountPolicy {}
```

OrderServiceImpl @Component, @Autowired 추가

그러면 AppConfig에서 했던 의존성 주입은 어떻게?

```
@Component
public class OrderServiceImpl implements OrderService {

    private final MemberRepository memberRepository;
    private final DiscountPolicy discountPolicy;

    @Autowired
```

이거 붙어있으면
필요한 빈을 자동으로
주입해줌.

(여기선 MemberRepository, 타입을 갖는 빈을 자동주입)
DiscountPolicy

의존성 주입

```
    public OrderServiceImpl(MemberRepository memberRepository, DiscountPolicy
discountPolicy) {
        this.memberRepository = memberRepository;
        this.discountPolicy = discountPolicy;
    }
}
```

- `@Autowired` 를 사용하면 생성자에서 여러 의존관계도 한번에 주입받을 수 있다.

컴포넌트 스캔 기본 대상

컴포넌트 스캔은 `@Component` 뿐만 아니라 다음과 내용도 추가로 대상에 포함한다.

- `@Component` : 컴포넌트 스캔에서 사용
- `@Controlller` : 스프링 MVC 컨트롤러에서 사용
- `@Service` : 스프링 비즈니스 로직에서 사용

- `@Repository` : 스프링 데이터 접근 계층에서 사용
- `@Configuration` : 스프링 설정 정보에서 사용

해당 클래스의 소스 코드를 보면 `@Component` 를 포함하고 있는 것을 알 수 있다.

```
@Component
public @interface Controller {
}

@Component
public @interface Service {
}

@Component
public @interface Configuration {
}
```

중복 등록과 충돌

컴포넌트 스캔에서 같은 빈 이름을 등록하면 어떻게 될까?

다음 두가지 상황이 있다.

1. 자동 빈 등록 vs 자동 빈 등록
2. 수동 빈 등록 vs 자동 빈 등록

①자동 빈 등록 vs 자동 빈 등록

- 컴포넌트 스캔에 의해 자동으로 스프링 빈이 등록되는데, 그 이름이 같은 경우 스프링은 오류를 발생시킨다.
  - `ConflictingBeanDefinitionException` 예외 발생

수동 빈 등록시 남는 로그 스프링에서 단순 오류가 많이 발생하니까 최근엔 compile 오류를 띄움

```
Overriding bean definition for bean 'memoryMemberRepository' with a different
definition: replacing
```

물론 개발자가 의도적으로 이런 결과를 기대했다면, 자동 보다는 수동이 우선권을 가지는 것이 좋다. 하지만 현실은 개발자가 의도적으로 설정해서 이런 결과가 만들어지기 보다는 여러 설정들이 꼬여서 이런 결과가 만들어지는 경우가 대부분이다!

그러면 정말 잡기 어려운 버그가 만들어진다. 항상 잡기 어려운 버그는 애매한 버그다.

그래서 최근 스프링 부트에서는 수동 빈 등록과 자동 빈 등록이 충돌나면 오류가 발생하도록 기본 값을 바꾸었다.