

NCS - Java 011 Annotatoin

1. @ annotation 이란?

- 사전적 의미로 "주석달기" 라는 뜻이며 일반적인 주석과 달리 어노테이션 앞에 "@" 을 이용하여 주석을 달아주는 것이다

- 어노테이션의 사용용도

용도	내용
컴파일러에 정보제공	@Override 어노테이션처럼 컴파일러에 method의 정보를 제공한다 - Override가 아닐 경우 오류
실행시간에 특수기능 제공	스프링 프레임웍에서 @Controller 또는 @Transaction 처럼 실행시간에 특수기능을 적용한다
코드 자동 생성	컴파일과정에 어노테이션 정보를 통해 필요한 코드를 자동으로 생성한다

2. java의 기본 annotation

- java에는 기 정의된 내장 annotation 이 있다

annotation	역할
@Override	메서드가 부모클래스로부터 오버라이드 된 메서드임을 명시한다. 부모클래스에 메서드가 없으면 컴파일 에러가 발생한다.
@Deprecated	가급적 사용을 자제해달라는 의미로 사용된다. 예를 들어 메서드 앞에 @Deprecated가 붙으면, 이 메소드를 사용하거나 오버라이드 할 경우, 컴파일 할 때 경고가 뜬다.
@SupperssWarnings	컴파일러의 경고와 관련된 어노테이션이다. @SuppressWarnings의 의미는 이 부분에 대해서 경고문을 출력하지 말라는 의미이다. 자바에서 주로 등장하는 경고문의 종류에는 deprecation과 unchecked가 있다. 이 둘 중 하나를 @SuppressWarnings옆의 괄호에 넣어야 한다.

3. annotation 만들기

- 어노테이션은 interface를 만드는 것과 비슷한 형태로 정의한다

```
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface Annotation {
    public String val() default "값";
    public String key();
}
```

속성	내용
@Target	- 적용할 대상 CONSTRUCTOR, FIELD, ENUM, LOCAL VARIABLE, METHOD, PACKAGE, PARAMETER, TYPE(Class)
@Retention	- 어노테이션 정보의 유지단계 SOURCE, CLASS, RUNTIME
@Documented	- javadoc등에 문서화되어야하는 엘리먼트일 경우
@Inherited	- 자동으로 상속받는 어노테이션 타입일 경우

4. annotation 사용하기

- @어노테이션명 형태로 사용한다

```
@Annotation(key = "키값추가1")
public class Main {
    public static void main(String args[]){
        Main main = new Main();
        String a = main. getClass().getAnnotation(Annotation.class).val();
        String b = main. getClass().getAnnotation(Annotation.class).key();
        System.out.println(a);
        System.out.println(b);
    }

    public Main(){
        System.out.println("클래스 초기화 됨");
    }
}
```