

어노테이션(Annotation)

어노테이션(Annotation)

❖ 어노테이션(Annotation)이란?

- 프로그램에게 추가적인 정보를 제공해주는 메타데이터(metadata)
- 어노테이션 용도
 - 컴파일러에게 코드 작성 문법 에러 체크하도록 정보 제공
 - 소프트웨어 개발 툴이 빌드나 배치 시 코드를 자동 생성하게 정보 제공
 - 실행 시(런타임시) 특정 기능 실행하도록 정보 제공

어노테이션(Annotation)

❖ 어노테이션 타입 정의와 적용

- 어노테이션 타입 정의
- 소스 파일 생성: AnnotationName.java
- 소스 파일 내용

```
public @interface AnnotationName {  
}
```

- 어노테이션 타입 적용
 - 적용 대상 앞에 @AnnotationName을 기술

```
@Override  
public void toString() { ... }
```

어노테이션(Annotation)

❖ 기본 엘리먼트 value

```
public @interface AnnotationName {  
    String value();  
    int elementName() default 5;  
}
```

----- 기본 엘리먼트 선언

- 어노테이션 적용할 때 엘리먼트 이름 생략 가능
@AnnotationName("값")
- 두 개 이상의 속성을 기술할 때에는 value=값 형태로 기술
@AnnotationName(value="값", elementName=3)

어노테이션(Annotation)

❖ 어노테이션 적용 대상

- 코드 상에서 어노테이션을 적용할 수 있는 대상
- `java.lang.annotation.ElementType` 열거 상수로 정의

ElementType 열거 상수	적용 대상
TYPE	클래스, 인터페이스, 열거 타입
ANNOTATION_TYPE	어노테이션
FIELD	필드
CONSTRUCTOR	생성자
METHOD	메소드
LOCAL_VARIABLE	로컬 변수
PACKAGE	패키지

어노테이션(Annotation)

❖ 어노테이션 유지 정책

- 어노테이션 적용 코드가 유지되는 시점을 지정하는 것
- `java.lang.annotation.RetentionPolicy` 열거 상수로 정의

RetentionPolicy 열거 상수	설명
SOURCE	소스상에서만 어노테이션 정보를 유지한다. 소스 코드를 분석할 때만 의미가 있으며, 바이트 코드 파일에는 정보가 남지 않는다.
CLASS	바이트 코드 파일까지 어노테이션 정보를 유지한다. 하지만 리플렉션을 이용해서 어노테이션 정보를 얻을 수는 없다.
RUNTIME	바이트 코드 파일까지 어노테이션 정보를 유지하면서 리플렉션을 이용해서 런타임에 어노테이션 정보를 얻을 수 있다.

어노테이션(Annotation)

❖ 어노테이션 유지 정책

- 리플렉션(Reflection)
 - 런타임에 클래스의 메타 정보를 얻는 기능
 - 클래스가 가지고 있는 필드, 생성자, 메소드, 어노테이션의 정보를 얻을 수 있음
 - 런타임 시 어노테이션 정보를 얻으려면 유지 정책을 RUNTIME으로 설정

어노테이션(Annotation)

❖ 런타임시 어노테이션 정보 사용하기

- 클래스에 적용된 어노테이션 정보 얻기
 - `클래스.class` 의 어노테이션 정보를 얻는 메소드 이용
- 필드, 생성자, 메소드에 적용된 어노테이션 정보 얻기
 - 다음 메소드 이용해 `java.lang.reflect` 패키지의 `Field`, `Constructor`, `Method` 클래스의 배열 얻어냄

리턴타입	메소드명(매개변수)	설명
<code>Field[]</code>	<code>getFields()</code>	필드 정보를 <code>Field</code> 배열로 리턴
<code>Constructor[]</code>	<code>getConstructors()</code>	생성자 정보를 <code>Constructor</code> 배열로 리턴
<code>Method[]</code>	<code>getDeclaredMethods()</code>	메소드 정보를 <code>Method</code> 배열로 리턴

어노테이션(Annotation)

❖ 런타임시 어노테이션 정보 사용하기

- Field, Constructor, Method가 가진 다음 메소드 호출
 - 어노테이션 정보를 얻기 위한 메소드

리턴타입	메소드명(매개변수)
boolean	<code>isAnnotationPresent(Class<? extends Annotation> annotationClass)</code> 지정한 어노테이션이 적용되었는지 여부. Class 에서 호출했을 경우 상위 클래스에 적용된 경우에도 true 를 리턴한다.
Annotation	<code>getAnnotation(Class<T> annotationClass)</code> 지정한 어노테이션이 적용되어 있으면 어노테이션을 리턴하고 그렇지 않다면 null 을 리턴한다. Class 에서 호출했을 경우 상위 클래스에 적용된 경우에도 어노테이션을 리턴한다.
Annotation[]	<code>getAnnotations()</code> 적용된 모든 어노테이션을 리턴한다. Class 에서 호출했을 경우 상위 클래스에 적용된 어노테이션도 모두 포함한다. 적용된 어노테이션이 없을 경우 길이가 0 인 배열을 리턴한다.
Annotation[]	<code>getDeclaredAnnotations()</code> 직접 적용된 모든 어노테이션을 리턴한다. Class 에서 호출했을 경우 상위 클래스에 적용된 어노테이션은 포함되지 않는다.