

Define an Annotation

```
@Retention(RUNTIME) //RetentionPolicy
@Target(TYPE) //ElementType
public @interface MyAnnot {
    boolean value() default false;
    String name();
}
```

Annotation for Classes, Enums and Interfaces available for Reflection

RetentionPolicy

| | |
|---------|--|
| CLASS | Compiled into .class No Reflection |
| RUNTIME | Available for Reflection |
| SOURCE | Only in Source Dropped for Compilation |

ElementType

| | |
|-----------------|-----------------------------------|
| TYPE | Class, Interface, Enum |
| FIELD | Class fields |
| METHOD | Methods no Constructor |
| PARAMETER | Any Param of a Method/Constructor |
| CONSTRUCTOR | Only Constructors |
| LOCAL_VARIABLE | only Local Vars |
| ANNOTATION_TYPE | valid for other Annotations |
| PACKAGE | Annotation for Package |

PACKAGE annotations should only be in `package-info.java`

Extract Annotation from Class

| | |
|---|------------------------------|
| <code>getAnnotation(Class<A> aClass)</code> | in this class or from parent |
| <code>getDeclaredAnnotation(Class<A> aClass)</code> | only declared in this class |
| <code>getAnnotations()</code> | array of Annotations |
| <code>getDeclaredAnnotations()</code> | array of Annotations |

Extract Fields and Methods

```
getDeclaredField(String name)
getDeclaredMethod(String name, Class<?>... parameterTypes)
getConstructors()
getDeclaredFields()
```

```
getDeclaredMethods()
```

`getDeclared...` always points to exactly this Object only including private, without Declared any super can also match but only public ones.

Extract Field information

```
getName()
getType()
getModifiers()
get(Object obj) extract field value from obj
set(Object obj, Object val) set field to value
```

Primitives are available e.g.: `getInt(obj), setInt(obj, val)`

Method Functions

```
getName()
getParameterTypes() Class Array of Types
getReturnType()
invoke(Object obj, Object... args) Call Method
```

Constructors

```
Object.class.newInstance() utilizes Default Constructor of an Class object
const.new Instance( Object.... args) utilizes one of the extracted Constructors
```



By **Chris** (cs8898)
cheatography.com/cs8898/

Published 5th May, 2021.
Last updated 1st February, 2021.
Page 1 of 1.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>