

Java SE & Spring

Module 1: Java SE

14.Java Annotations



What is Java Annotations?

Java annotations are metadata (data about data) for our program source code. They provide additional information about the program to the compiler but are not part of the program itself. These annotations do not affect the execution of the compiled program. Annotations start with @.

Syntax:

1

```
@AnnotationName
```

Let's take an example of **@Override** annotation.

The **@Override** annotation specifies that the method that has been marked with this annotation overrides the method of the superclass with the same method name, return type, and parameter list.

It is not mandatory to use **@Override** when overriding a method. However, if we use it, the compiler gives an error if something is wrong (such as wrong parameter type) while overriding the method.

Java Annotations

- Annotations start with '@'.
- Annotations do not change action of a compiled program.
- Annotations help to associate metadata (information) to the program elements i.e. instance variables, constructors, methods, classes, etc.
- Annotations are not pure comments as they can change the way a program is treated by compiler.

Annotation Formats

Annotations may also include elements (members/attributes/parameters).

1. Marker Annotations

Marker annotations do not contain members/elements. It is only used for marking a declaration. Since these annotations do not contain elements, parentheses can be excluded.

Syntax:

1	@AnnotationName()
2	//or
3	@AnnotationName

Annotation Formats

Annotations may also include elements (members/attributes/parameters).

2. Single Element Annotations

A single element annotation contains only one element.

Syntax:

```
@AnnotationName(elementName = "elementValue")
```

- If there is only one element, it is a convention to name that element as value.

```
@AnnotationName(value = "elementValue")
```

- In this case, the element name can be excluded as well. The element name will be value by default.

```
@AnnotationName("elementValue")
```

Annotation Formats

Annotations may also include elements (members/attributes/parameters).

3. Multiple Element Annotations

These annotations contain multiple elements separated by commas.

Syntax:

```
@AnnotationName(element1 = "value1", element2 = "value2")
```

Annotations Placement

Any declaration can be marked with annotation by placing it above that declaration. As of Java 8, annotations can also be placed before a type.

1. Above declarations

As mentioned above, Java annotations can be placed above class, method, interface, field, and other program element declarations.

Example:

```
1  class Animal {
2      public void displayInfo() {
3          System.out.println("I am an animal.");
4      }
5  }
6
7  class Dog extends Animal {
8      @Override
9      public void displayInfo() {
10         System.out.println("I am a dog.");
11     }
12 }
```


Annotations Placement

2. Type Annotations

Before Java 8, annotations could be applied to declarations only. Now, type annotations can be used as well. This means that we can place annotations wherever we use a type.

Example:

```
1 public void getUsers(@PathVariable int id)
```



Practice Time: Let's Code Together!



Questions ?



Next:Concurrency in Java