Core Java
Annotation

Akshita Chanchlani

Annotations

- *Annotations*, a form of metadata, provide data about a program.
- Annotations have a number of uses, among them:
    1. **Information for the compiler** – Annotations can be used by the compiler to detect errors or suppress warnings.
    2. **Compile-time and deployment-time processing** – Software tools can process annotation information to generate code, XML files, and so forth.
    3. **Runtime processing** – Some annotations are available to be examined at runtime.

Types of Annotation

1. **Marker Annotation**
   - Annotation without element is called marker annotation
   - Example : @Entity, @Id, @override, @Deprecated
2. **Single-value Annotation**
   - Annotation which is having single element is called single value annotation.
   - Example : @Table(name = “employees”)
3. **Multi-value Annotation**
   - Annotation which is having multiple elements is called multi value annotation.
   - Example: @Entity(tableName = “vehicles”, primaryKey=“id”)

Example Single value and Multiple Value Annotation

Single-value Annotation

```
@interface Demo{   int value(); } // declaration
@interface Demo{   int value() default 0; } // declaration along with default value
@Demo(value=10)  // applying the single value annotation
```

Multi-value Annotation

@interface Emp{
int value1();
String value2();
String value3();
}
}
	@interface Emp{
	int value1() **default** 1;
	String value2() **default** "";
	String value3() **default** “ABC";
	}
		@Emp(value1=20,value2="Akshita",value3=“Pune")

Predefined Annotation Types

- A set of annotation types are predefined in the Java SE API. Some annotation types are used by the Java compiler, and some apply to other annotations.
- Annotation Types Used by the Java Language
    1. @Deprecated
    2. @override
    3. @SuppressWarnings
    4. @SafeVarargs
    5. @FunctionalInterface

Predefined Annotation Types

- Annotations that apply to other annotations are called *meta-annotations*.
- There are several meta-annotation types defined in java.lang.annotation package.
- Annotations that apply to other annotations:
    1. @Retention
    2. @Documented
    3. @Target
    4. @Inherited
    5. @Repeatable
    6. @Native

Custom Annotation Type

- The annotation type definition looks similar to an interface definition where the keyword interface is preceded by the at sign (@) (@ = AT, as in annotation type).
- Annotation types are a form of *interface.*
- There are few points that we should remember
    1. Element definition should not have any parameter.
    2. Element definition not have any throws clauses
    3. Element definition should return one of the following:
        1. primitive data types,
        2. String,
        3. Class, enum or array of these data types.
    4. We should attach @ just before interface keyword to define annotation.
    5. It may assign a default value to the method.

Retention Policy

- **RetentionPolicy.SoURCE**
  - The marked annotation is retained only in the source level and is ignored by the compiler.

- **RetentionPolicy.CLASS**
  - The marked annotation is retained by the compiler at compile time, but is ignored by the Java Virtual Machine (JVM).

- **RetentionPolicy.RUNTIME**
  - The marked annotation is retained by the JVM so it can be used by the runtime environment.

Annotation Target

1. ANNOTATION_TYPE
2. CONSTRUCTOR
3. FIELD
4. LOCAL_VARIABLE
5. METHOD
6. PACKAGE
7. PARAMETER
8. TYPE
9. TYPE_PARAMETER
10. TYPE_USE

Thank You.