

---

---

# Annotations in Java



*Java Features*  
*Presented by Ramin Zare*

---

---

# @Annotations

1. Annoation ها چه هستند؟
2. چگونه تعریف میشوند؟
3. محدودیت های Annotation ها چیست؟
4. Meta-annotation چیست و چگونه از آنها استفاده کنیم؟
5. Annotation های متداول

# @Annotations

**Martin Webster :**

**“A note added by way of comment or explanation”**

```
[modifiers] @ interface <annotation-type-name> {  
    // Annotation type body goes here  
}
```

# محدودیت ها

## 1. ارث بری در Annotation ها معنی ندارد

هر Annotation به طور ضمنی از کلاسی به نام `java.lang.annotation.Annotation` ارث بری میکند که به شکل زیر تعریف شده است.

```
package java.lang.annotation;
```

```
public interface Annotation {  
    boolean equals(Object obj);  
    int hashCode();  
    String toString();  
    Class<? extends Annotation> annotationType();  
}
```

# محدودیت ها

2. متود ها نمیتوانند پارامتر ورودی داشته باشند

در زمان اجرا در واقع هر متود یک getter محسوب میشود. هد تعریف متود در annotation ها مشخص کردن data element ها میباشد

3. متود ها نمیتوانند گزارش پرتاب استثنا (throws) بدهند

4. خروجی متود ها میتواند یکی از انواع primitive ها ; java.lang.String

; java.lang.Class ; enums ; annotation و آرایه ای از هر کدام باشد

## محدودیت ها

5. نام متود نمی تواند override ای از متودهای کلاس Object باشد
6. Annotation نمی تواند Generic باشد
7. حین استفاده از annotation ها نمیتوانیم از مقادیر null استفاده کنیم

# مختصر نویسی Annotation ها

```
//1
@interface Enabled {
    boolean enable() default true;
}
```

```
//2
@interface Company {
    String value();
}
```

```
@Company("Apple Inc.")
class IPod {

}
```

# Meta-Annotation Types

- **@Target**
- **@Retention**
- **@Inherited**
- **@Documented**
- **@Repeatable (Java 8)**



# @Target

- **ANNOTATION\_TYPE**
- **CONSTRUCTOR**
- **FIELD**
- **LOCAL\_VARIABLE**
- **METHOD**
- **MODULE (JAVA 9)**
- **PACKAGE**
- **PARAMETER**
- **TYPE**
- **TYPE\_PARAMETER (JAVA 8)**
- **TYPE\_USE (JAVA 8)**

# @Retention

- **RetentionPolicy.CLASS**
- **RetentionPolicy.RUNTIME**
- **RetentionPolicy.SOURCE**

# @Inherited

```
@Inherited  
@interface InheritedAnnotation{ }
```

```
@InheritedAnnotation  
classClazz{  
  
}
```

```
class InheritedClass extendsClazz{  
  
}
```

# @Documented

```
@Documented
public @interface Version {
    int major();
    int minor() default 0;
}
```

com.javaland.annotation

## Class Class1

java.lang.Object  
com.javaland.annotation.Class1

---

```
@Version(major=1)
public class Class1
extends java.lang.Object
```

Test Version Annotation

---

# @Repeatable

```
@Query(name = "q1" , query ="from Person")
@Query(name = "q2" , query ="from Student")
public class Class2 {
}
```

```
@Repeatable(Queries.class)
@Retention(RetentionPolicy.RUNTIME)
public @interface Query {
    String name();
    String query();
}
```

```
@Retention(RetentionPolicy.RUNTIME)
public @interface Queries {
    Query[] value();
}
```

# Annotation های متداول

- **@Override**
- **@Deprecated**
- **@SuppressWarnings**
- **@FunctionalInterface**
- **@SafeVarargs**

## @Deprecated

```
public void notDeprecated(){  
}
```

```
@Deprecated(since = "2")  
public void deprecatedMethod(){  
  
}
```

```
@Deprecated(since = "2" , forRemoval = true)  
public void forRemovalMethod(){  
  
}
```

## @SupressWarning

```
@SuppressWarnings({"deprecation", "removal"})  
public static void main(String[] args) {  
    Api api = new Api();  
    api.deprecatedMethod();  
    api.forRemovalMethod();  
}
```

[/\\*https://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftask-suppress\\_warnings.htm\\*/](https://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftask-suppress_warnings.htm)



# Thanks!

Presented by :

Ramin Zare

