ANNOTATIONS AND JAVA DOC

JAVA DOC

- There is a tool called java doc that will help to prepare the documentation for the classes.
- Java provides some tags for java documentation those are known as java doc tool.
- The tags for class or a package are:
  - @author:

    Adds the author name of the class.
  - @version:

    Adds a version subheading with specified version text to generated docs.
  - @since:

    To mention when was the version written or how long it may be valid.
  - @see:

    Adds a see also heading with a link for example to see the references to the given link.
- The tags for methods are:
  - @param:

    To mention the parameters taken by particular product.
  - @return:

    To mention the value returned by the method.
  - @throws/@exception:

    To know the exception thrown by the method.
  - @deprecated:

    To mention whether the method is deprecated or not that is the method may not be used longer.

- @code:

  Displays text in code font without interpreting the text as HTML mark-up or nested java doc tags.

➢ The other tags available are:

- @link:

  To provide the link for particular resource.

- @value:

  To provide value for any static variable or a member.

- @serial:

  To mention the serial id for serialization.

BUILT-IN ANNOTATIONS IN JAVA

➢ Annotations are used for giving attributes or defining the attributes for a class or a interface or methods.

➢ Annotations are useful for giving meta data to class or interface or a method.

➢ Built-in annotations can be categorised into two:

➔ Applied to code:

These are set of annotations applied upon the code.so, this type of annotation gives the hint to the compiler so that it avoids showing errors and warnings.

The in-built annotations applied to the code are:

- @Override:

  It informs the compiler that the element is meant to over ride an element declared in a superclass.

- @deprecated:

  It indicates that the marked element is deprecated and should no longer be used.

- @FunctionalInterface:

  Indicates that the type declaration is intended to be a functional interface

- @SuppressWarnings:

  It tells the compiler to suppress specific warnings that it would otherwise generate.

- @SafeVarArgs: Variable arguments & method should be final/private

  When it is applied to a method or a constructor, it asserts that the code does not perform potentially unsafe operations on its varargs parameter.

➔ Applied to other annotations:

These are set of annotations applied upon user-defined annotation.

The different user-defined annotations are:

```
@interface MyAnno
{
    String name(); default "Mohit"
    String project();
}
@MyAnno(name = "Varu",project = "Bank")
public class AnoDemo{
// can use for method variables interface
}
```

- @Retention:

  It specifies how the marked annotations are stored.

- @Documented:

  It indicates that whenever the specified annotation is used those elements should be documented using java doc tool

- @Target:

  It marks another annotation to restrict what kind of java element the java elements can be applied to.

- @Inherited:

  It indicates that the annotation type can be inherited from super class.

- @Repeatable:

  It indicates that the marked annotation can be applied more than once to the same declaration or type used.

➢ Meta data:

Metadata is "data that provides information about other data".In other words, it is "data about data."