Documentation Tool: JavaDoc

Introduction:

1 What is Java?

JavaDoc tool is a document generator tool in Java programming language for generating standard documentation in HTML format. It generates API documentation. It parses the declarations ad documentation in a set of source file describing classes, methods, constructors, and fields.

Javadoc - generate HTML pages of API documentation from Java source files
https://docs.oracle.com/en/java/javase/22/docs/specs/man/javadoc.html

Installation Process:

Video link: https://www.youtube.com/watch?v=CJxMwbJPisw
Blog page: https://medium.com/@tecnicorabi/mastering-javadocs-a-comprehensive-guide-2bd7b9d6d8dc

2 Basic Syntax:

```
import java.util.*;
public class SWE {
    public static void main(String[] args)
    {
        System.out.println("Project!");
    }
}
```

3 Generation of JavaDoc:

To create a JavaDoc you do not need to compile the java file. To create the Java documentation API, you need to write Javadoc followed by file name.

javadoc file_name or javadoc package_name

4 JavaDoc Tags:

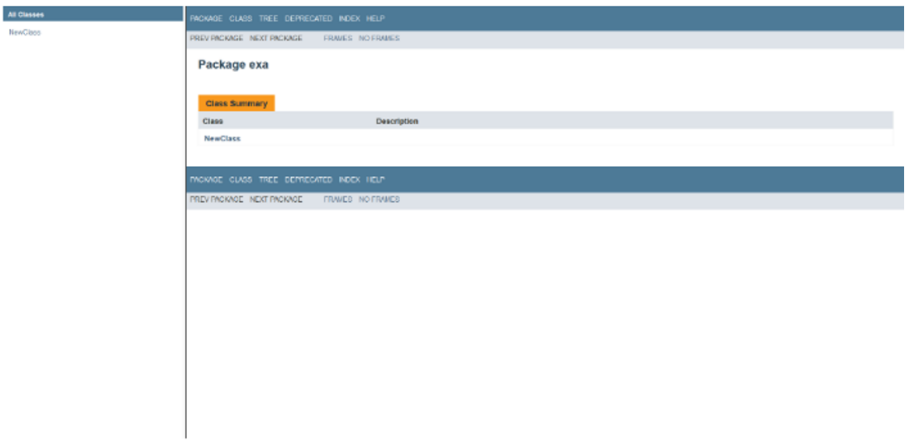
Tag	Parameter	Description
@author	author_name	Describes an author
@param	parameter description	Provide information about method parameter or the input it takes
@see	reference	Generate a link to another element of the document
@version	version-name	Provide version of the class, interface, or enum
@return	return	Provide the return value

5 Example:

```
package exa;
import java.util.Scanner;

/**
*
* @author Yash
*/
public class Example {
        /**
* This is a program for adding two numbers in java.
        * @param args
*/
        public static void main(String[] args)
        {
                /**
                * This is the main method
                * which is very important for
                * execution for a java program.
                */
```

```
                int x, y;
                Scanner sc = new Scanner(System.in);
                /**
                * Declared two variables x and y.
                * And taking input from the user
                * by using Scanner class.
                *
                */

                x = sc.nextInt();
                y = sc.nextInt();
                /**
                * Storing the result in variable sum
                * which is of the integer type.
                */
                int sum = x + y;

                /**
                * Using standard output stream
                * for giving the output.
                * @return null
                */
                System.out.println("Sum is: " + sum);
        }
}
```

Generating document for the above class
javadoc exa



6 Advantages:

- JavaDoc comments provide a quick way to view method details and return values on hover, saving coding time.
- Frequently used in Oracle's predefined classes (e.g., 'ArrayList', 'String', 'Object').
- Reduces the need to constantly reference the API, though checking the API is still recommended for a full list of methods.
- Helps other developers understand the general purpose of a method without reading the entire code.
- Offers a concise overview of functionality, improving code readability.

7 Disadvantages:

- Can clutter the code if not written concisely and clearly.
- Poorly written comments may confuse developers and lead to mistakes.
- Writing and maintaining JavaDoc comments can be time-consuming, especially as the code evolves.