

Java Documentation Comments

The javadoc Tags

javadoc Tool

Java comments

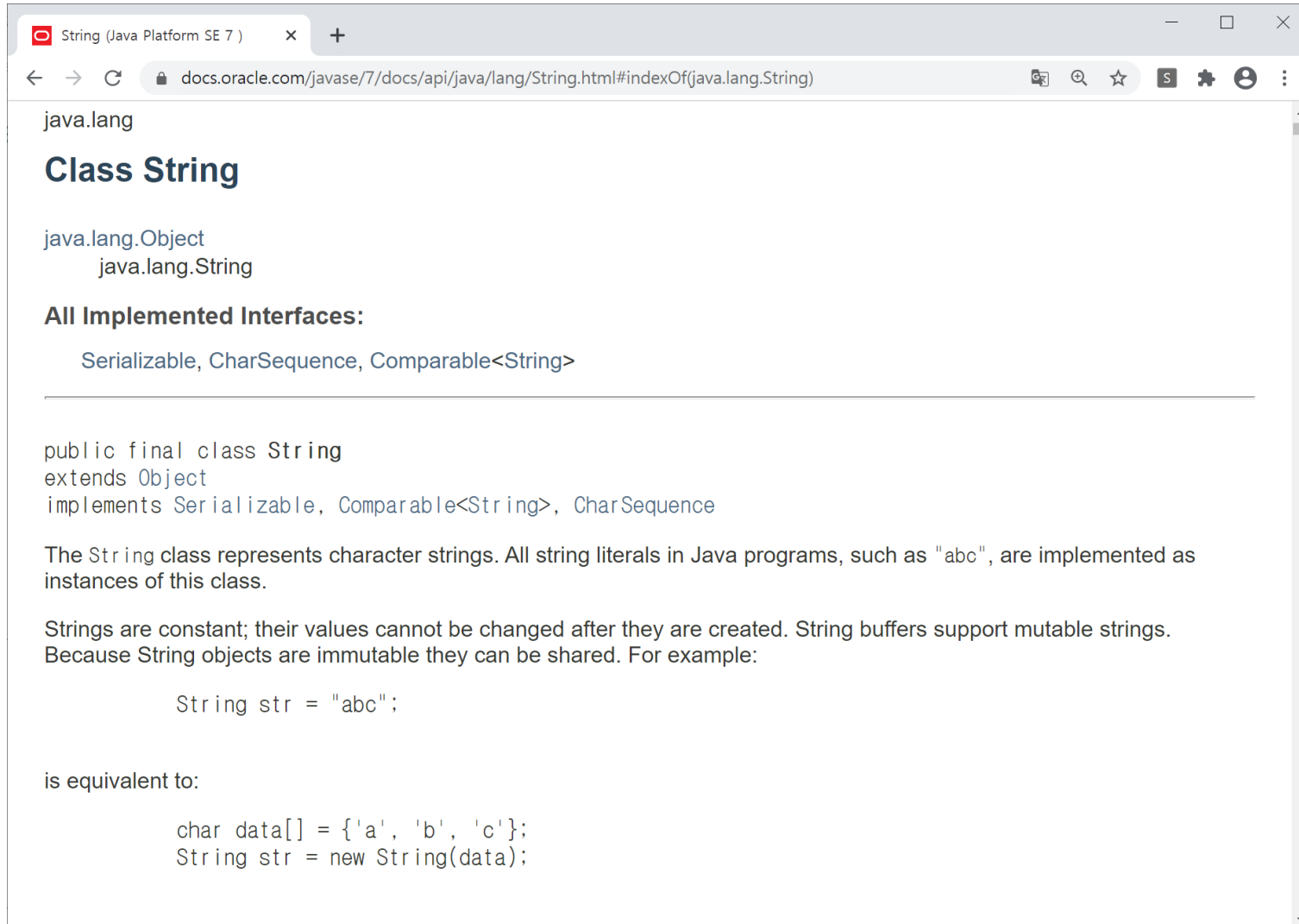
❖ Java Comments

- `/* comments */`
- `// line comment`
- `/** Java documentation comment */`

❖ Java Documentation Comment

- allow you to embed information about your program into the program itself.
- Documentation comments make it convenient to document your programs.
- You can then use the javadoc utility program to extract the information and put it into an HTML file.

Java API Documentation



The screenshot shows a web browser window with the title "String (Java Platform SE 7)". The address bar displays the URL "docs.oracle.com/javase/7/docs/api/java/lang/String.html#indexOf(java.lang.String)". The page content includes the package "java.lang", the class name "Class String", and its inheritance hierarchy: "java.lang.Object" and "java.lang.String". It lists "All Implemented Interfaces:" as "Serializable, CharSequence, Comparable<String>". The class declaration is shown as "public final class String" extending "Object" and implementing "Serializable, Comparable<String>, CharSequence". A description states: "The String class represents character strings. All string literals in Java programs, such as \"abc\", are implemented as instances of this class." Another paragraph explains: "Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:" followed by the code "String str = \"abc\";". It then states "is equivalent to:" followed by the code "char data[] = {'a', 'b', 'c'};" and "String str = new String(data);".

String (Java Platform SE 7)

docs.oracle.com/javase/7/docs/api/java/lang/String.html#indexOf(java.lang.String)

java.lang

Class String

java.lang.Object
java.lang.String

All Implemented Interfaces:

Serializable, CharSequence, Comparable<String>

```
public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence
```

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};
String str = new String(data);
```

```
package javadoc;  
import java.io.*;
```

```
/**  
 * This class demonstrates documentation comments.  
 * @author Ayan Amhed  
 * @version 1.2  
 */
```

```
public class SquareNum {  
    /**  
     * This method returns the square of num.  
     * This is a multiline description. You can use  
     * as many lines as you like.  
     * @param num The value to be squared.  
     * @return num squared.  
     */  
    public double square(double num) {  
        return num * num;  
    }  
}
```

```

/**
 * This method inputs a number from the user.
 * @return The value input as a double.
 * @exception IOException On input error.
 * @see IOException
 */
public double getNumber() throws IOException {
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader inData = new BufferedReader(isr);
    String str;
    str = inData.readLine();
    return (new Double(str)).doubleValue();
}

/**
 * This method demonstrates square().
 * @param args Unused.
 * Omit @return for methods that return void and for constructors;
 * @exception IOException On input error.
 * @see IOException
 */
public static void main(String args[]) throws IOException
{
    SquareNum ob = new SquareNum();
    double val;
    System.out.println("Enter value to be squared: ");
    val = ob.getNumber();
    val = ob.square(val);
    System.out.println("Squared value is " + val);
}
}

```

How to use javadoc tool

❖ `javadoc -d doc SquareNum.java`

The Generated Documentation

Class SquareNum

java.lang.Object
javadoc.SquareNum

```
public class SquareNum
extends java.lang.Object
```

This class demonstrates documentation comments.

Version:

1.2

Author:

Ayan Amhed

Constructor Summary

Constructors

Constructor and Description

SquareNum()

Method Summary

All Methods

Static Methods

Instance Methods

Concrete Methods

Modifier and Type

Method and Description

double

getNumber()

This method inputs a number from the user.

static void

main(java.lang.String[] args)

This method demonstrates square().

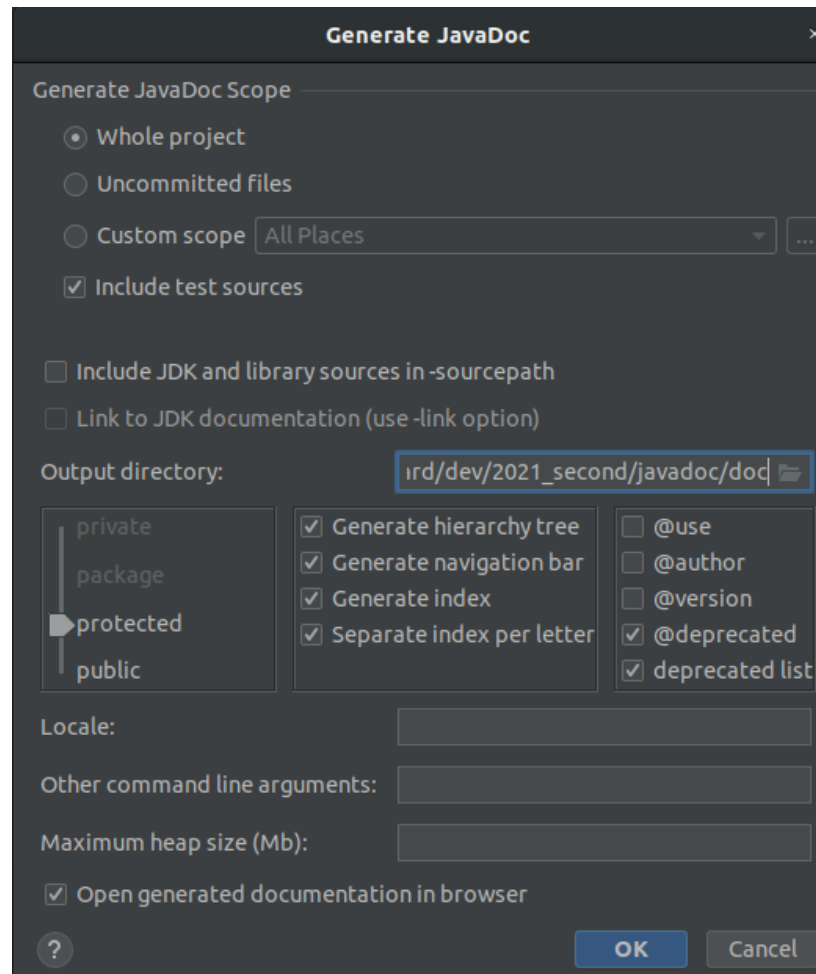
double

square(double num)

This method returns the square of num.

Invoking javadoc in IntelliJ

❖ Tools – Generate JavaDoc ...



The screenshot shows the 'Generate JavaDoc' dialog box in IntelliJ IDEA. The dialog is titled 'Generate JavaDoc' and has a close button (X) in the top right corner. It contains several sections for configuring the documentation generation process.

Generate JavaDoc Scope

- ☒ Whole project
- ☐ Uncommitted files
- ☐ Custom scope: All Places (dropdown menu with a plus icon)
- ☒ Include test sources
- ☐ Include JDK and library sources in -sourcepath
- ☐ Link to JDK documentation (use -link option)

Output directory: ird/dev/2021_second/javadoc/doc (with a folder icon)

Visibility

- private
- package
- protected** (selected with a slider)
- public

Options

- ☒ Generate hierarchy tree
- ☒ Generate navigation bar
- ☒ Generate index
- ☒ Separate index per letter
- ☐ @use
- ☐ @author
- ☐ @version
- ☒ @deprecated
- ☒ deprecated list

Locale: (empty text field)

Other command line arguments: (empty text field)

Maximum heap size (Mb): (empty text field)

☒ Open generated documentation in browser

Buttons: ? (help), OK, Cancel

JavaDoc Tags: Summary

Tag & Parameter	Usage	Applies to
@author <i>name</i>	Describes an author.	Class, Interface, Enum
@version <i>version</i>	Provides software version entry. Max one per Class or Interface.	Class, Interface, Enum
@since <i>since-text</i>	Describes when this functionality has first existed.	Class, Interface, Enum, Field, Method
@see <i>reference</i>	Provides a link to other element of documentation.	Class, Interface, Enum, Field, Method
@param <i>name description</i>	Describes a method parameter.	Method
@return <i>description</i>	Describes the return value.	Method
@exception <i>classname description</i> @throws <i>classname description</i>	Describes an exception that may be thrown from this method.	Method

Doclet

- ❖ Doclets are programs written in the Java™ programming language that use the doclet API to specify the content and format of the output of the Javadoc tool.
- ❖ By default, the Javadoc tool uses the "standard" doclet to generate API documentation in HTML form.
- ❖ However, you can supply your own doclets to customize the output of Javadoc as you like
- ❖ New "Doclet API" to supersede the previous API, which was becoming increasingly difficult to update to support new language features (Since JDK 9)

References

❖ **Java Documentation Comments** from Tutorials Point

- http://www.tutorialspoint.com/java/java_documentation.htm

❖ **How to Write Doc Comments for the Javadoc Tool**

- <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

❖ **Javadoc Tool**

- <http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html>

❖ **Doclet Overview**

- <https://openjdk.java.net/groups/compiler/using-new-doclet.html>