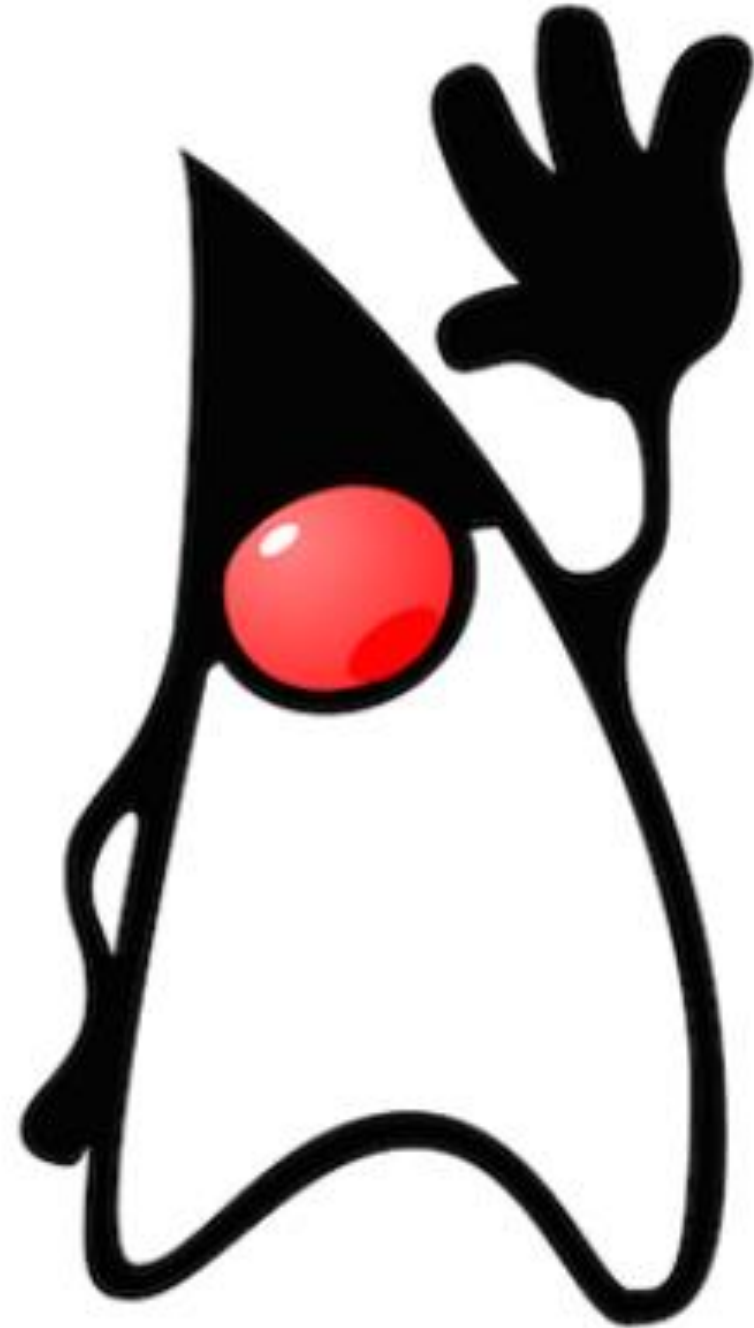


# Javadoc

-VAIBHAV PARIHAR



# Agenda

---

- We will learn about Javadoc and how we can use it to generate useful documentation for Java code.
- API Documentation Tool – Javadoc
  - What and Why is Javadoc?
  - Javadoc Comments
  - Javadoc Tags
  - Javadoc for classes
  - Javadoc for methods
  - How to use it in an IDE?

# What is Javadoc?

---

- Javadoc is a tool which comes with JDK and it is used for generating Java code documentation in HTML format from Java source code, which requires documentation in a predefined format.
- The HTML format is used for adding the convenience of being able to hyperlink related documents together.

# Javadoc Comments

---

❖ The **Javadoc comments structure** looks very similar to a regular multi-line comment, but the key difference is the **extra asterisk** at the beginning:

```
// This is a single line comment
```

```
/*
```

```
* This is a regular multi-line comment
```

```
*/
```

```
/**
```

```
* This is a Javadoc
```

```
*/
```

# Javadoc Tags

---

- ❖ The Javadoc tool recognizes the following tags –
  - **@author** - Adds the author of a class.
  - **@version** - Adds a "Version" subheading with the specified version-text
  - **@param** - Adds a parameter with the specified parameter name followed by the specified description to the "Parameters" section.
  - **@return** - Adds a "Returns" section with the description text.
  - **@throws** - Adds a Throws subheading to the generated documentation, with the class name and description text.
  - **@code** – Displays text in code font without interpreting the text as HTML markup or nested Javadoc tags.
  - **@since** - Adds a "Since" heading with the specified since-text to the generated documentation.
- And a few more...

```
App.java X index.html
package com.test.junitExamples;

/**
 * This class is used to demonstrate implementation of division operation.
 * @version 1.0
 * @author vaibhavparihar
 */
public class App {
```

PACKAGE CLASS USE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD    DETAIL: FIELD | CONSTR | METHOD

**Package** com.test.junitExamples

## Class App

java.lang.Object<sup>↗</sup>  
com.test.junitExamples.App

---

```
public class App
extends Object↗
```

This class is used to demonstrate implementation of division operation.

**Version:**

1.0

**Author:**

vaibhavparihar

# Javadoc for classes

---

```

public class App {
    public static void main(String[] args) {
        System.out.println("Just a basic file explaining Javadoc");
    }

    /**
     * @param x Dividend
     * @param y Divisor
     * @return Quotient (x/y)
     * @throws ArithmeticException : Indicates y is 0
     */
    public static int divide(int x, int y) throws ArithmeticException{
        return x/y;
    }
}

```

## Method Details

### main

```
public static void main(String[] args)
```

### divide

```
public static int divide(int x,
                        int y)
                        throws ArithmeticException
```

#### Parameters:

x - : Dividend

y - : Divisor

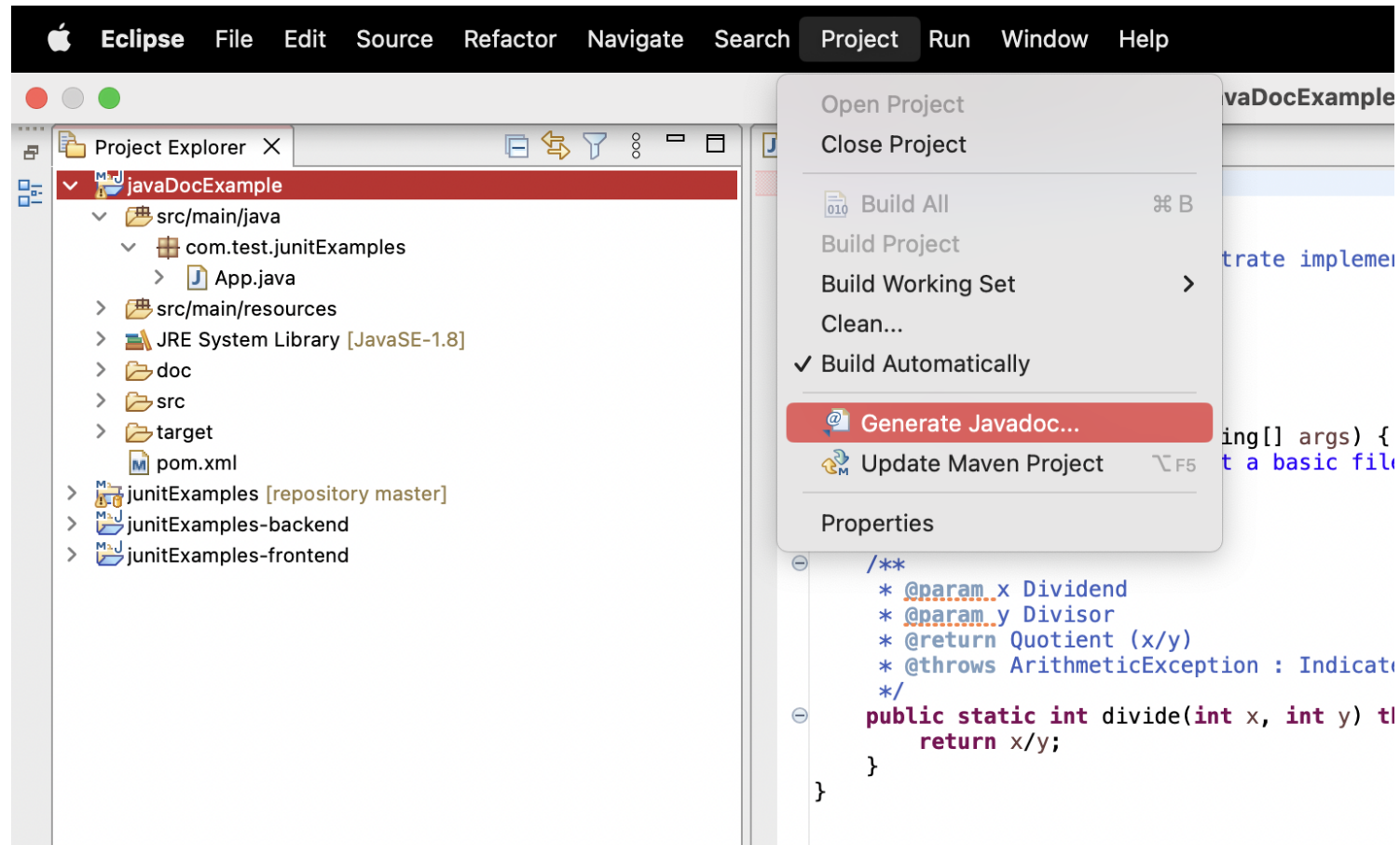
#### Returns:

: Quotient (x/y)

#### Throws:

`ArithmeticException` - : Indicates y is 0

# Javadoc for methods

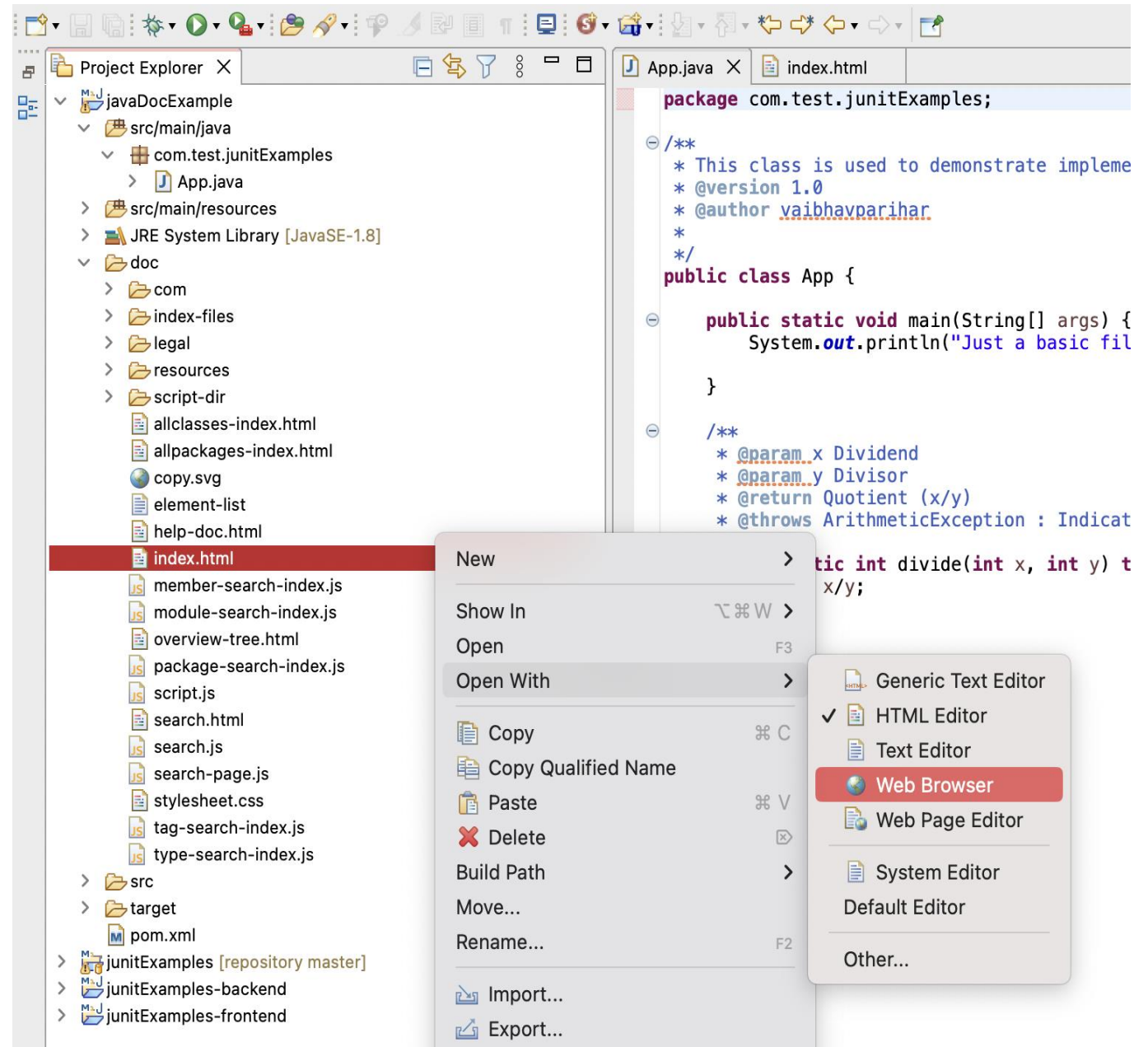


Let's generate Javadoc using Eclipse IDE



## ❖ Once you will click on “Generate Javadoc”:

- A “*doc*” folder will be generated inside your project folder..
- Search for “*Index.html*” file inside it.
- Open it with your default web browser.



***Thank you for your time!***