1. JAVADOC

(http://www.oracle.com/technetwork/articles/java/index-137868.html)

A doc comment is written in HTML and must precede a class, field, constructor or method declaration. It consists a description part followed by block tags. The description part should be a summary sentence, containing a concise but complete description of the API item.

Documentation comments (doc comments) are delimited by the /** ... */ delimiters. These comments are processed by the Javadoc tool to generate the API docs.

The javadoc tags in their order:

- @author (classes and interfaces only, required)
- @version (classes and interfaces only, required)
- @param (methods and constructors only)
- @return (methods only)
- @exception (@throws is a synonym added in Javadoc 1.2)
- @see
- @since
- @serial
- @deprecated

Example:

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param  url  an absolute URL giving the base location of the image
 * @param  name the location of the image, relative to the url argument
 * @return      the image at the specified URL
 * @see         Image
 */
public Image getImage(URL url, String name) {
        try {
            return getImage(new URL(url, name));
        } catch (MalformedURLException e) {
            return null;
        }
}
```

Run the Javadoc tool

getImage
public Image getImage(URL url,
String name)
Returns an Image object that can then be painted on the screen. The url argument must specify an absolute URL. The name argument is a specifier that is relative to the url argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

Parameters:
url - an absolute URL giving the base location of the image.

name - the location of the image, relative to the url argument.

Returns:
the image at the specified URL.

See Also:
Image

Generate Javadoc document from Eclipse: Project -> generate Javadoc

2. Saving to pdf files

https://dzone.com/articles/creating-pdf-documents-with-apache-pdfbox-2

https://pdfbox.apache.org/