

# JAVADOC

Prasad Perera  
COMP 249

# Comments in Java

2

```
// Can be single line comments
```

```
/* More than one line.  
   Useful to "erase" a block  
   of code from compilation */
```

```
/**
```

```
 * A Javadoc comment  
 * To generate nice HTML documentation  
 */
```

# ¿Qué es Javadoc?

3

- Estándar para documentar proyectos en Java.
- Permite añadir descripciones a las clases, constructores, campos, interfaces y métodos en la documentación HTML escribiendo los comentarios antes de las declaraciones respectivas.
- Ejemplo:

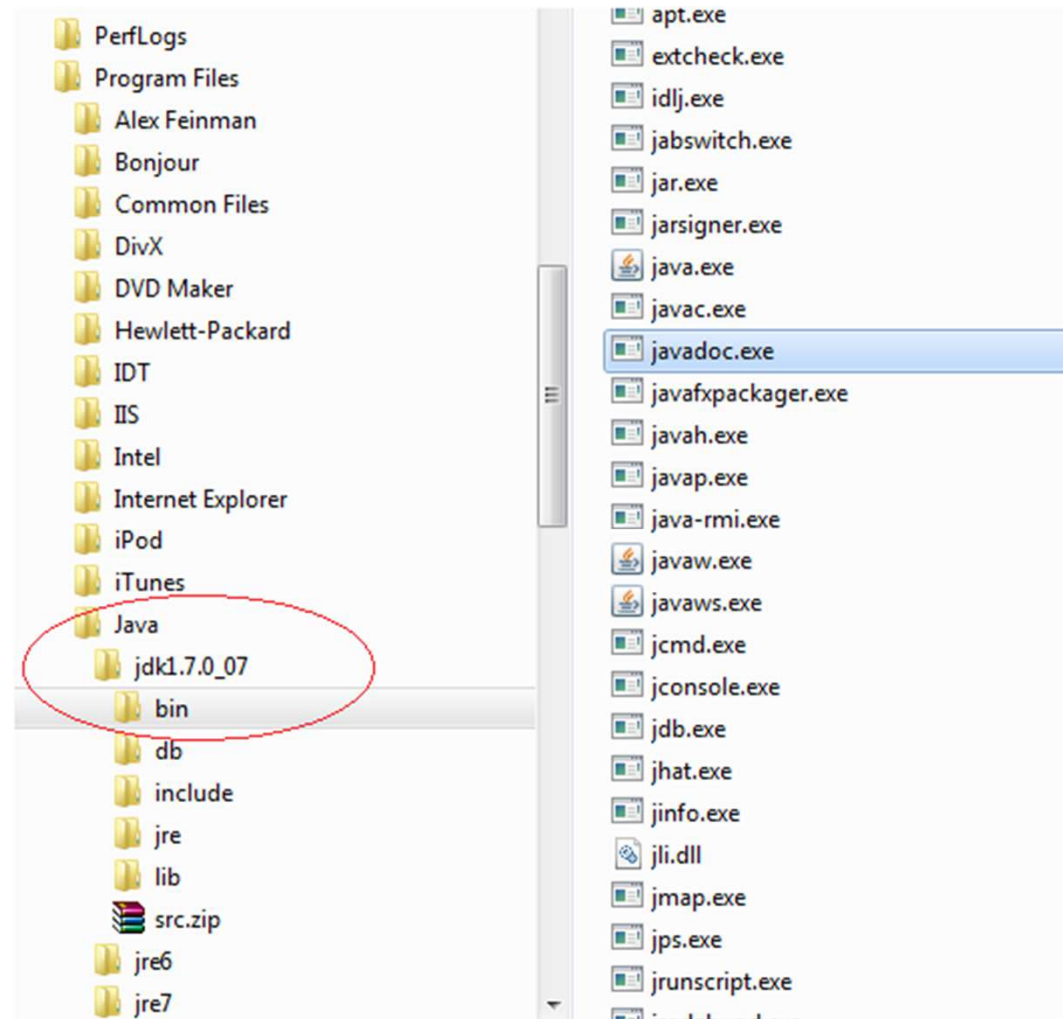
```
/**  
 * Clase que implementa métodos para calcular las calificaciones  
 * de los estudiantes  
 */  
public class Student {
```

# Compilando Javadoc

4

## javadoc.exe

- Genera la documentación de las clases Java en un formato HTML.
- Pertenece al JDK.
- Para cada `X.java`, crea un archivo `X.html` más algunos archivos extra (index, ..)



# Javadoc Compilación

5

- Compilación desde la línea de comandos:

- *Javadoc [options] [files]*
- some basic Javadoc options

- author*

- classpath [path] / [path];[path];*

- sourcepath [path] / -sourcepathlist [path];[path];*

- version*

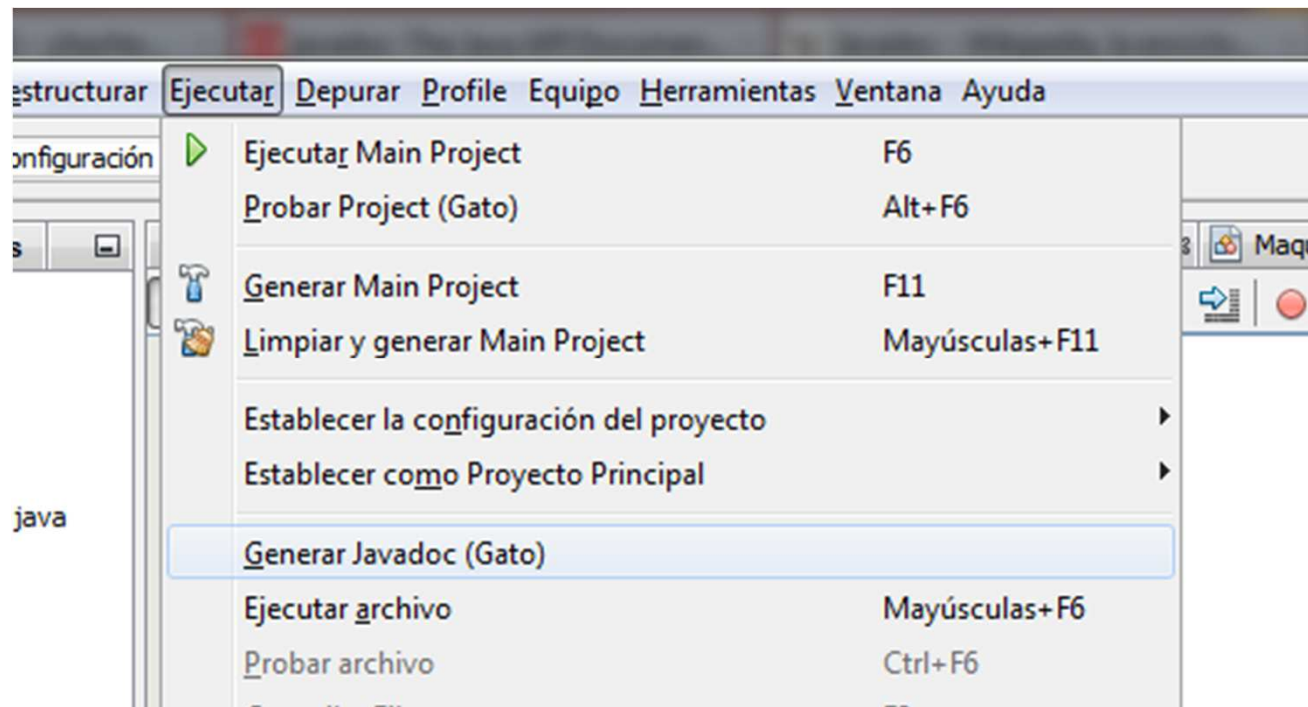
- Ejemplo:

- *c:\> Javadoc -version 1.1 -sourcepath C:\Mysources -  
classpath C:\Mylibs*

# Javadoc Compilación

6

## □ En NetBeans



# Archivos Javadoc

7

The screenshot shows a web browser window titled "Generated Documentation". The address bar displays the file path: `file:///C:/Users/prasad/workspace/Tutorial1/doc/index.html`. The browser's tab bar shows several open tabs, including "Alistair Knott: Public...", "Gifs, Gifs and More ...", "LaTeX MikTeX for du...", "Admin Menu Page -...", and "IMDb Pictures & Photos o...".

The main content area of the browser displays the Javadoc for the `Class Student`. The left sidebar contains a "Packages" section with "<unnamed package>" and "question1", and an "All Classes" section with "Animal" and "Student". The main content area has a navigation bar with tabs: "Overview", "Package", "Class" (selected), "Use", "Tree", "Deprecated", "Index", and "Help". Below the navigation bar, there are links for "Prev Class", "Next Class", "Frames", and "No Frames". The "Summary" section shows "Nested | Field | Constr | Method" and "Detail: Field | Constr | Method".

The class `Student` is shown as a subclass of `java.lang.Object` and `question1.Student`. The class definition is:

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

The class description states: "Student class implements methods to calculate student grades."

The "Constructor Summary" section shows a table with one constructor:

Constructor and Description
<code>Student ()</code>

The "Method Summary" section shows a table with two methods:

Modifier and Type	Method and Description
<code>void</code>	<code>calculateOverallScore ()</code>
<code>void</code>	<code>finalLetterGrade ()</code>

# Forma general de la documentación

8

```
/**  
 * One sentence ending with a period describing the purpose.  
 * Additional lines giving  
 * more details (html tags can be included)  
 *  
 * javadoc tags to specify more specific information,  
 * such as parameters and return values for a method  
 *  
 * @Tag Description ...  
 * @Tag Description ...  
 *  
 */
```



# Etiquetas

9

- Permiten especificar información usada en los archivos HTML
- Etiquetas más comunes:
  - Para archivos, clases e interfaces:  
`@author Autor`  
`@version Número de Version`
  - Para métodos:  
`@param Nombre descripción`  
`@return descripción`  
`@exception exceptionClass descripción`  
`@deprecated descripción`
  - For everything:  
`@see relatedReference (ex. other class name)`

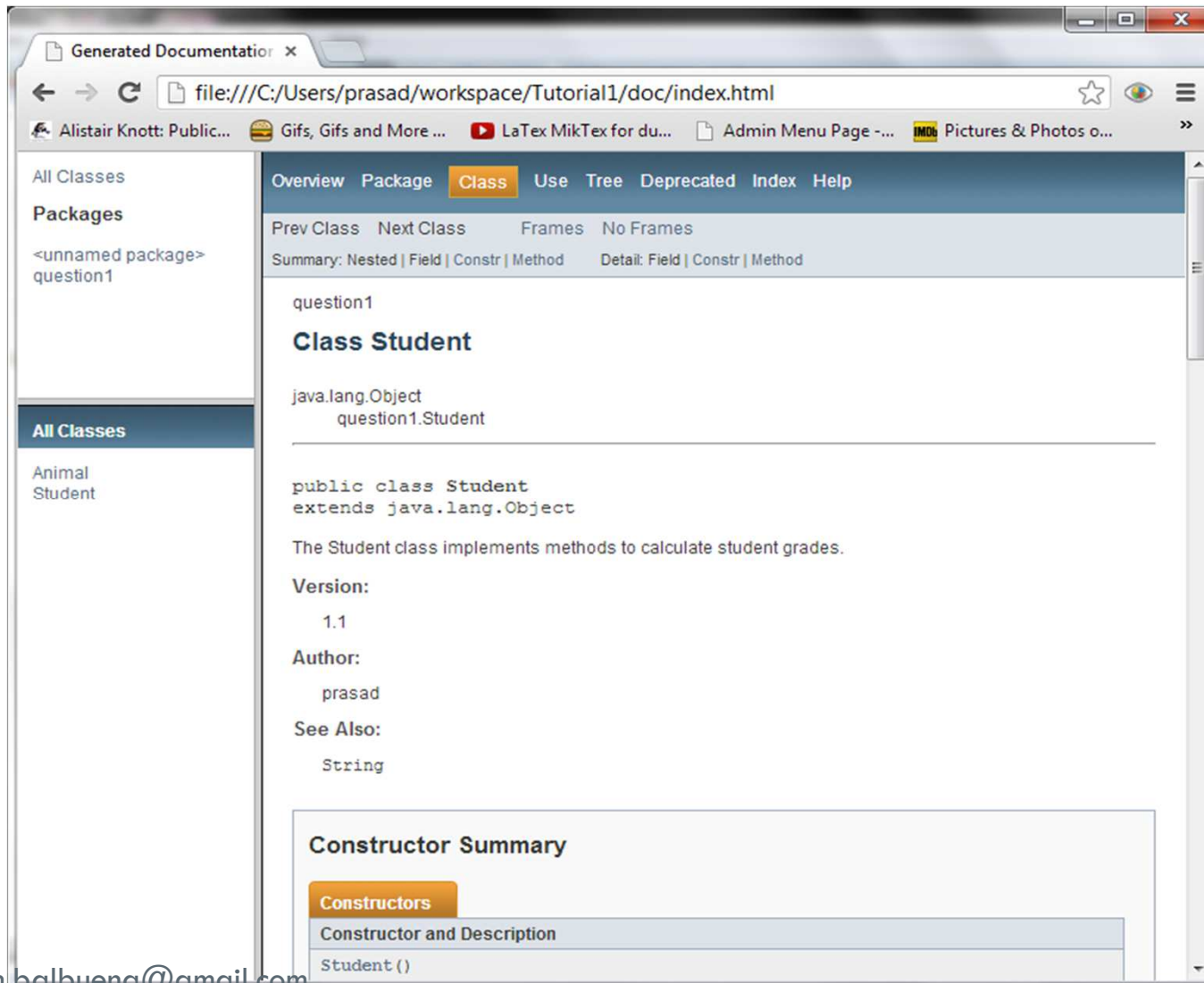
# Documentando Clases e Interfaces

10

```
/**  
 * La clase estudiante implementa métodos para calcular las  
 * calificaciones de los estudiantes.  
 * @author Miriam Balbuena  
 * @version 1.1  
 * @see String  
 */  
public class Student {
```

# Documentando Clases e Interfaces

11



# Documentando métodos

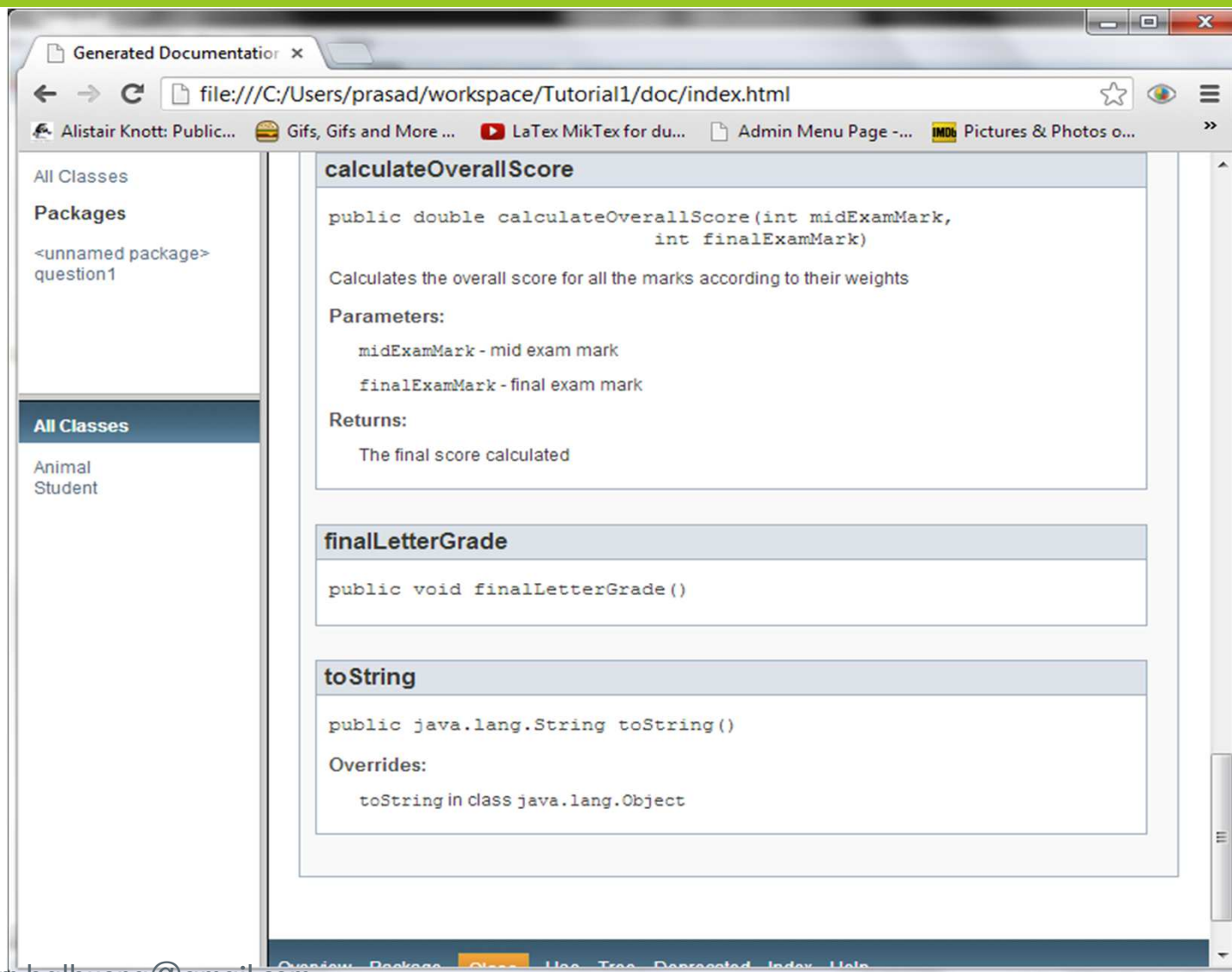
12

- Los comentarios deben estar inmediatamente antes de la definición del método
- Etiquetas comunes:
  - `@param <name of parameter> <description>`
  - `@return <description>`
- Ejemplo:

```
/**
 * Calculates the overall score for all the marks according to their weights
 * @param midExamMark mid exam mark
 * @param finalExamMark final exam mark
 * @return The final score calculated
 */
public double calculateOverallScore(int midExamMark, int finalExamMark){
```

# Documentando métodos

13



# Javadoc Extra

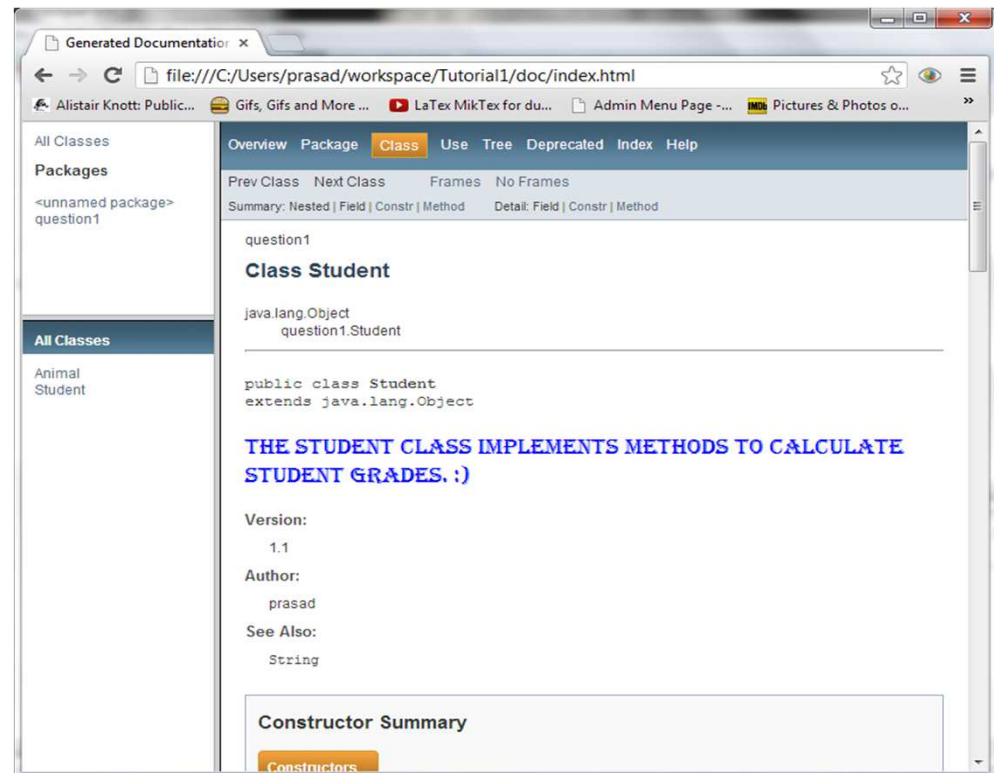
14

- Java doc acepta etiquetas HTML en los comentarios para dar un formato más atractivo en el documento resultante.
  
- Ventajas:
  - Mejor estética
  - Cuando el código cambia, también puede cambiar la documentación:

# Javadoc Extra Cont'd.

15

```
/**
 * <p style="font-family:algerian;color:blue;font-size:20px;">
 * The Student class implements methods to calculate student grades. :)</p>
 * @author prasad
 * @version 1.1
 * @see String
 */
public class Student {
```



# Ejercicio

16

- Documentar las clases del sistema de Agencia de autos.
  - ▣ Documentar todos los métodos con las etiquetas `@param` y `@return` dónde hagan falta.
  - ▣ Generar la documentación

