





OOP with Java

19. Documentation with Javadoc

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区 计算机科学与技术系 应用优化研究所 中国 安徽省 合肥市 蜀山区 230601 经济技术升发区 锦绣大道99号

Outline



- Introduction
- 2 Javadoc
- Javadoc Generation
- 4 Summary





• If you write a program, usually you are not just doing it alone



- If you write a program, usually you are not just doing it alone
- Often, you work in a team



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are
 - You need to tell them what your member variables are storing and how they are used



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are
 - You need to tell them what your member variables are storing and how they are used
- For this, we have comments



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are
 - You need to tell them what your member variables are storing and how they are used
- For this, we have comments:
 - // marks the rest of the current line as comment



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are
 - You need to tell them what your member variables are storing and how they are used
- For this, we have comments:
 - // marks the rest of the current line as comment
 - /* starts a possible multi-line comment that ends with */



- If you write a program, usually you are not just doing it alone
- Often, you work in a team
- Or you write code to be used by other people
- So you need to tell them what your code does:
 - You need to define what a class does and what it is good for
 - You need to state what a method does and what the meaning of its parameters are
 - You need to tell them what your member variables are storing and how they are used
- For this, we have comments:
 - // marks the rest of the current line as comment
 - /* starts a possible multi-line comment that ends with */
 - /** starts a Javadoc comment and */ ends it

Javadoc



• Javadoc comments are comments following a specific structure

Javadoc



- Javadoc comments are comments following a specific structure
- A tool (called Javadoc) can convert them to HTML web pages

Javadoc



- Javadoc comments are comments following a specific structure
- A tool (called Javadoc) can convert them to HTML web pages
- IDEs can use them to display information while programmers are typing code



• A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for
- A Javadoc /** */ comment directly before a static or instance method



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for
- A Javadoc /** */ comment directly before a static or instance method
 - is used to describe what the method does



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for
- A Javadoc /** */ comment directly before a static or instance method
 - is used to describe what the method does
 - can contain a line starting with @param A meaning to describe the
 meaning of parameter A



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for
- A Javadoc /** */ comment directly before a static or instance method
 - is used to describe what the method does
 - can contain a line starting with @param A meaning to describe the
 meaning of parameter A
 - can contain a line starting with @return meaning to describe the meaning of the return value if the method is a function



- A Javadoc /** */ comment directly before a class declaration is used to describe what the class is good for
- A Javadoc /** */ comment directly before a static or instance variable declaration is used to describe what the variable is good for
- A Javadoc /** */ comment directly before a static or instance method
 - is used to describe what the method does
 - can contain a line starting with @param A meaning to describe the
 meaning of parameter A
 - can contain a line starting with @return meaning to describe the meaning of the return value if the method is a function
- A package can contain a file package-info.java with a Javadoc
 /** */ describing the meaning of the package

The ComplexNumber class with Javadoc Annotations



Listing: The ComplexNumber class with Javadoc Annotations

```
package cn.edu.hfuu.iao.math:
public final class ComplexNumber {
  private final double realPart;
  private final double imaginaryPart;
  public ComplexNumber(final double realPart, final double imaginaryPart) {
    this.realPart = _realPart;
    this imaginaryPart = imaginaryPart:
  public final ComplexNumber add(final ComplexNumber x) {
    return new ComplexNumber ((this.realPart + x.realPart), (this.imaginaryPart + x.imaginaryPart)); //(\alpha_x + \alpha_y) + (\beta_x + \beta_y)
  public final ComplexNumber subtract(final ComplexNumber x) {
    return new ComplexNumber ((this, realPart - x, realPart), (this, imaginaryPart - x, imaginaryPart)): //(\alpha_v - \alpha_v) + (\beta_v - \beta_v)
  public final ComplexNumber multiply(final ComplexNumber x)
    double at = this.realPart, bt = this.imaginaryPart, a2 = x.realPart, b2 = x.imaginaryPart;
    return new ComplexNumber(((a1 * a2) - (b1 * b2)), ((a1 * b2) + (b1 * a2))); //(\alpha_x\alpha_y - \beta_x\beta_y) + (\alpha_x\beta_y + \beta_x\alpha_y)
  public ComplexNumber divide(final ComplexNumber x) {
    double at = this.realPart. b1 = this.imaginaryPart. a2 = x.realPart. b2 = x.imaginaryPart:
    (((a2 * b1) - (b2 * a1)) / ((a2 * a2) + (b2 * b2)))); // \frac{\alpha_2\beta_2 - \beta_2\alpha_2}{2^{2} + 3^2}
  public void println() {
    System.out.print(this.realPart);
                                            System.out.print("u+u"); //$NON-NLS-1$
    System.out.print(this.imaginaryPart); System.out.println('1');
```



Listing: The ComplexNumberTest class with Javadoc Annotations

```
package cn.edu.hfuu.iao.math;
public class ComplexNumberTest {
   * Oparam aras
  public static final void main(String[] args) {
    ComplexNumber a, b, res;
    a = new ComplexNumber(20d, 0d); // instantiate a = 20 + 0i
    a.println(); // print 20 + 0i
    b = new ComplexNumber(1d, -2d); // create b = 1 - 2i
    b.println(): // print 1+2i
    (res = a.multiplv(b)).println(): // set res = a*b = 20*(1-2i) = 20-40i and vrint
    res = a.multiply(b).subtract(b); // we can chain methods by applying subtract to the result of multiply
    res.println(): // print the result of the above computation: (20*(1-2i))-(1-2i)=19*(1-2i)=19-38i
    res.divide(b).println(); // print the result of \frac{19-38i}{11-2i} = 19 = 19 + 0i, but don't store it:
    res.multiply(res).divide(res.multiply(//\frac{(19-38i)*(19-38i)}{(19-28i)*(19-3)} = 28.5 - 9.5i
        new ComplexNumber(1d, -1d))).println();
```

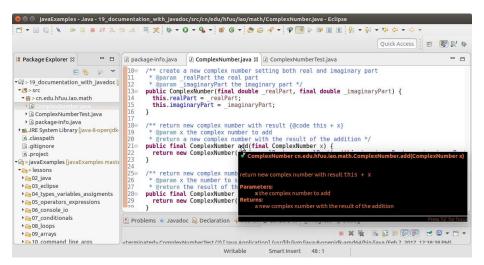


Listing: An Example package-info.java File

```
/** This package contains a mathematical class. */
package cn.edu.hfuu.iao.math;
```

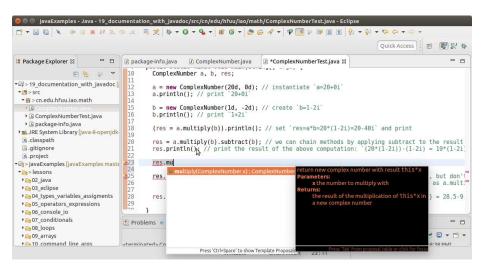
Pop-Up Infos in Eclipse based on JavaDoc



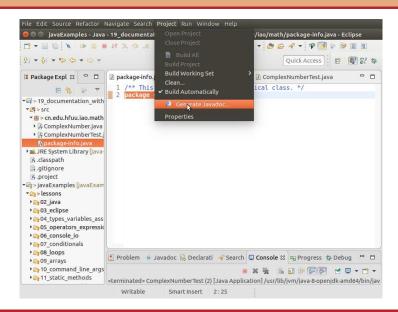


Pop-Up Infos in Eclipse based on JavaDoc











Select types for Javadoc generation. Javadoc command: [/ust/lib/jvm/java-8-openjdk-amd64/bin/javadoc Select types for which Javadoc will be generated: 19_documentation_with_javadoc Create Javadoc for members with visibility: Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse Use custom doclet	Javadoc command: /usr/lib/jym/java-8-openjdk-amd64/bin/ja Select types for which Javadoc will be gene 19_documentation_with_javadoc Create Javadoc for members with visibility Private Package Private: Generate Javadoc for all classe: Use standard doclet	erated:
(ust/lib/jvm/java-8-openjdk-amd64/bin/javadoc Select types for which Javadoc will be generated: ✓ □ 19_documentation_with_javadoc Create Javadoc for members with visibility: ● Private ● Package ● Protected ● Public Private: Generate Javadoc for all classes and members. ● Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse	/usr/lib/jvm/java-8-openjdk-amd64/bin/ja Select types for which Javadoc will be gene 19_documentation_with_javadoc 19_documentation_with_javadoc Create Javadoc for members with visibility Private Package Private: Generate Javadoc for all classes Use standard doclet	erated:
Select types for which Javadoc will be generated: 19_documentation_with_javadoc 19_documentation_with_javadoc Create Javadoc for members with visibility: Private Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: //home/tweise/local/programming/java/javaExamples/lessc Browse	Select types for which Javadoc will be gene 19_documentation_with_javadoc Create Javadoc for members with visibility Private Package Private: Generate Javadoc for all classe: Use standard doclet	erated:
Create Javadoc for members with visibility: Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc	Create Javadoc for members with visibility Private Prackage Private: Generate Javadoc for all classes Use standard doclet	y:
Create Javadoc for members with visibility: Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc	Create Javadoc for members with visibility Private Prackage Private: Generate Javadoc for all classes Use standard doclet	y:
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
 Private Package Protected Public Private: Generate Javadoc for all classes and members. Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse 	Private Private: Generate Javadoc for all classes Use standard doclet	
Use standard doclet Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse	O Use standard doclet	
Destination: /home/tweise/local/programming/java/javaExamples/lessc Browse		es and members.
	Destination: /home/tweise/local/	
		/programming/java/javaExamples/lessc Browse
Doclet name:	Poclet name:	
Doclet class path:	Doclet class path:	



vadoc Generation			@
configure Javadoc argume	ents for standard doclet		ر ا
Document title:			
Basic Options		Document these tags	
Generate use page		author @author	
Generate hierarchy tre	ee	@version	
Generate navigator ba	ır	@deprecated	
Generate index		deprecated list	
Separate index per	letter		
elect referenced archives		3	
# 19_documentation_	with Javadot - not com	igured	Select All
🗌 <text> charsets.jar - https:,</text>	//docs.oracle.com/javas	9	Select All
 ☐ Gharsets.jar - https:, ☐ ☐ Cldrdata.jar - not co ☐ ☐ Hospital Confidence 	//docs.oracle.com/javas nfigured igured	9	Clear All
 ☐ charsets.jar - https:, ☐ documentation ☐ documentation	//docs.oracle.com/javas nfigured igured not configured	9	Select All Clear All Browse
in the charsets.jar - https: in the charsets.jar - not co in the conficient of	//docs.oracle.com/javas nfigured igured not configured	e/8/docs/api/	Clear All
in the charsets.jar - https: in the charsets.jar - not co in the conficient of	//docs.oracle.com/javas nfigured igured not configured figured	e/8/docs/api/	Clear All



Javadoc Generation				@_
Configure Javadoc argun	nents.			47
Overview:				Browse
VM options (prefixed with	n '-J', e.gJ-Xmx180m for l	arger heap spa	ce):	
Extra Javadoc options (pa	ath names with white space	es must be encl	osed in quotes):	
- Control of Options (pr	annumes man mine space	25 111052 52 21121	osco III quotes,	
JRE source compatibility:	1.8	-		
	1.8 Is Javadoc export as an Ar	▼ nt script:		
Save the settings of th		astronomerono.	/lessons/19_do	Browse
	is Javadoc export as an Ar e/local/programming/java	astronomerono.	/lessons/19_doi	Browse







```
javaExamples - Java - 19 documentation with javadoc/src/cn/edu/hfuu/iao/math/ComplexNumber.java - Eclipse
🖰 + 🔚 🔚 🔌 🕩 III 🔳 18 3. 🤧 19 18 3. 🦈 4 + 🔾 + 🚱 + 😭 + 😭 + 🤔 6 4 + + 19
0 - 0 - 15 C - 0 -

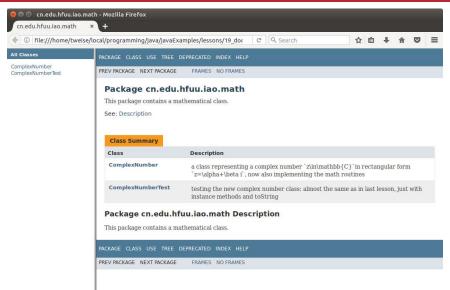
□ Package Explorer 
□
                                 D package-info.iava

☑ ComplexNumber.java 
☒ ☐ ComplexNumberTest.java

                                                                                                                                      - -
                                       package cn.edu.hfuu.iao.math;
▼2 > 19 documentation with java
                                       /** a class representing a complex number `z\in\mathbb{C}`in rectangular form `z=\
                                       nublic final class ComplexNumbe
 ▼(地 > SEC
                                         /** the real part `\alpha` of the complex number */
  *# > cn.edu.hfuu.iao.math
                                         private final double realPart;
   ▶ ☐ ComplexNumber.java
                                         /** the imaginary part '\beta' */
   ▶ A ComplexNumberTest.iava
   ▶ □ package-info.iava
                                 😰 Problems @ Javadoc 🙆 Declaration 🦑 Search 📮 Console 🛭 🖷 Progress 🏇 Debug
                                                                                                                                      - B
▶ ■ JRE System Library [java-8-op
 + @> doc
  12 .classpath
                                 <terminated> Javador Generation
  agitignore.
                                 Building tree for all the packages and classes...
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfuc
  7. project
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
▼ > iavaExamples [iavaExamples
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
 ▼ (m> lessons
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/constant-v
  ▶ € 02 java
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
  ▶ € 03 eclipse
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/cn/edu/hfu
  ▶ 6 04 types variables assigme
                                 Building index for all the packages and classes...
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/overview-t
  ▶ 1 05 operators expressions
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
  ▶ € 06 console io
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
  ▶ ≥ 07 conditionals
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
  ► 08 loops
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
  > 09 arrays
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/index-file
  > 10 command line args
                                 Generating /home/tweise/local/programming/jaya/jayaExamples/lessons/19 documentation with jayadoc/doc/index-file
                                 Generating /home/tweise/local/programming/java/javaExamples/lessons/19 documentation with javadoc/doc/deprecated-
  ▶ 11 static methods
                                 Building index for all classes...
                                 Generating /home/tweise/local/orogramming/java/javaExamples/lessons/19 documentation with javadoc/doc/allclasses.
```

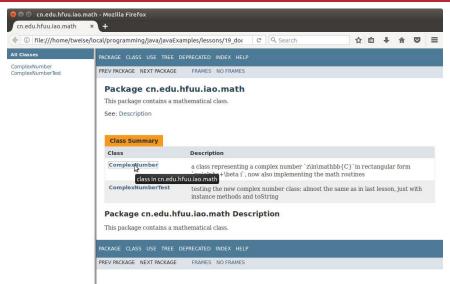
The Generated Javadoc





The Generated Javadoc

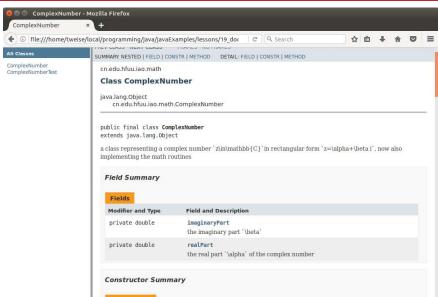




 $file: ///home/tweise/local/programming/java/javaExamples/lessons/19_documentation_with_javadoc/doc/cn/edu/hfuu/iao/math/ComplexNumber.html$

The Generated Javadoc





Summary



- In order to make our code useful, we need to document it
- Good programmers write the documentation while they are coding, because you are never going to do it afterwards, even though you think/claim you would
- Javadoc allows us to write comments in a structured way
- IDEs can exploit this structured information to display useful pop-up infos to the programmer
- We can also generate HTML documentations from the documented code via the javadoc tool
- Style checkers can automatically whether all code is properly documented



谢谢 Thank you

Thomas Weise [汤卫思] tweise@hfuu.edu.cn http://iao.hfuu.edu.cn

Hefei University, South Campus 2 Institute of Applied Optimization Shushan District, Hefei, Anhui, China

