CSC 7003 : Basics of Software Engineering

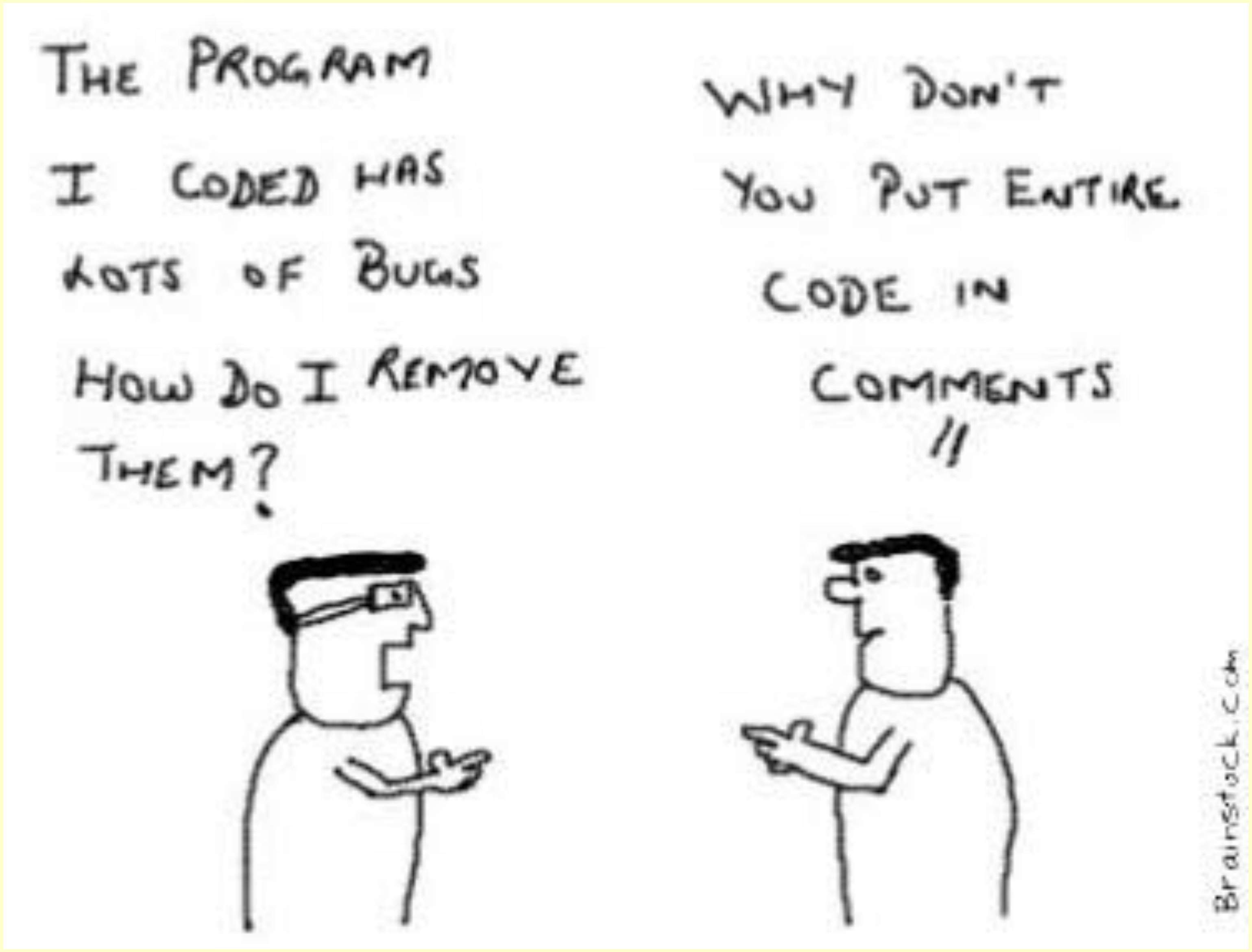
J Paul Gibson, D311

paul.gibson@telecom-sudparis.eu

http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC7003/

Code Documentation

…/~gibson/Teaching/CSC7003/L8-CodeDocumentation.pdf


THE PROGRAM I CODED HAS LOTS OF BUGS HOW DO I REMOVE THEM?
WHY DON'T YOU PUT ENTIRE CODE IN COMMENTS //
Brainstock.com

Types Of Documentation In Software Development

It is, traditionally, written text that complements software/models in order to clarify or explain:

- **Requirements**
- **Design**
- **Implementation**
- **Tests**
- **End User**
- **Marketing**

The implementation/code should be documented, and the most important aspect is that the documentation should clarify the consistency between the code and the other software/models in the system. It should highlight implementation decisions – *why not what*

Documenting Code: some reading material

•*Information Distribution Aspects of Design Methodology*, **David Lorge Parnas**, 1971

•*Literate Programming*, **Donald E. Knuth**, 1984

•*How To Comment Code***, Steve Drevok,** 1996

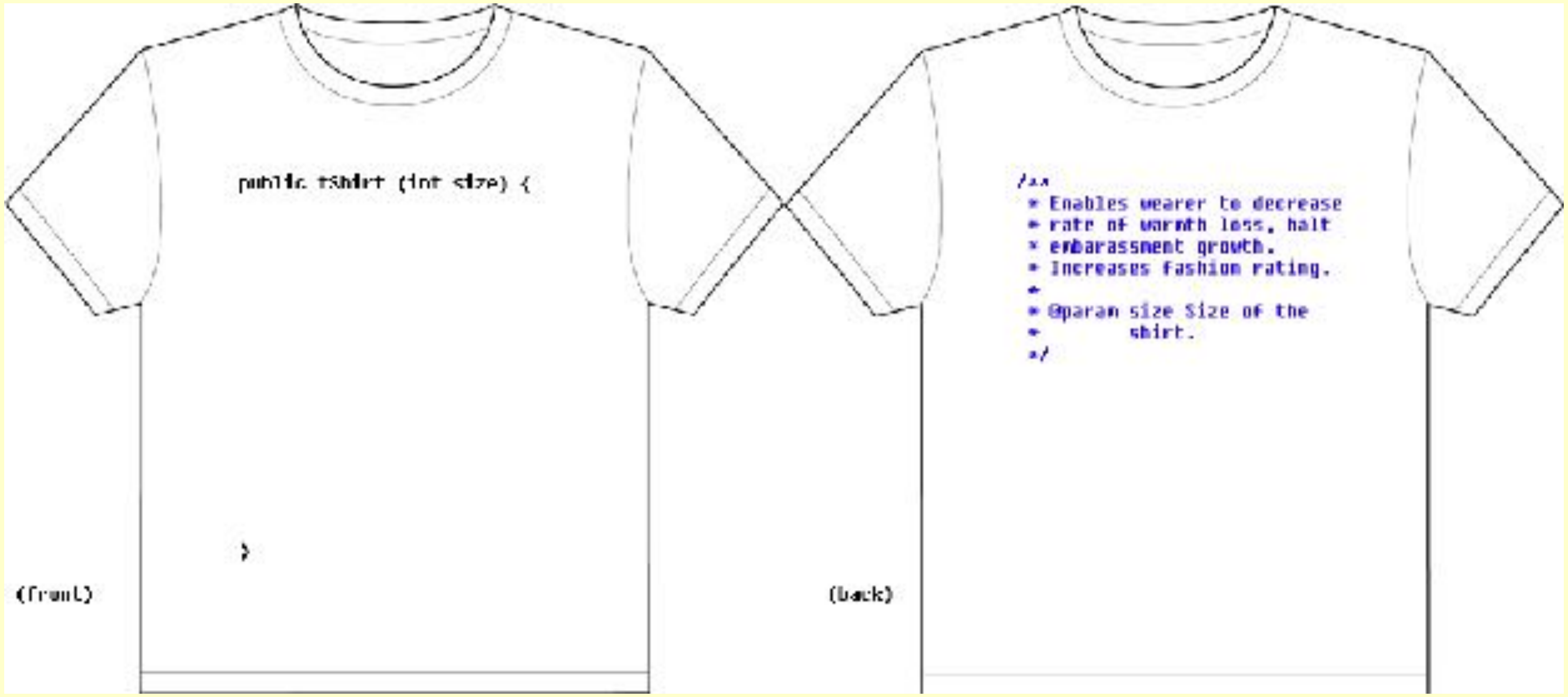
• *Comments Are More Important Than Code*, **Jef Raskin**, 2005

Clean code: a handbook of agile software craftsmanship, **Robert Martin**, 2008

•*Coding Guidelines: Finding the Art in the Science***, Robert Green and Henry Ledgard,** 2011

Documenting Code : 2 useful tools

Doxygen and JavaDoc: links on web site



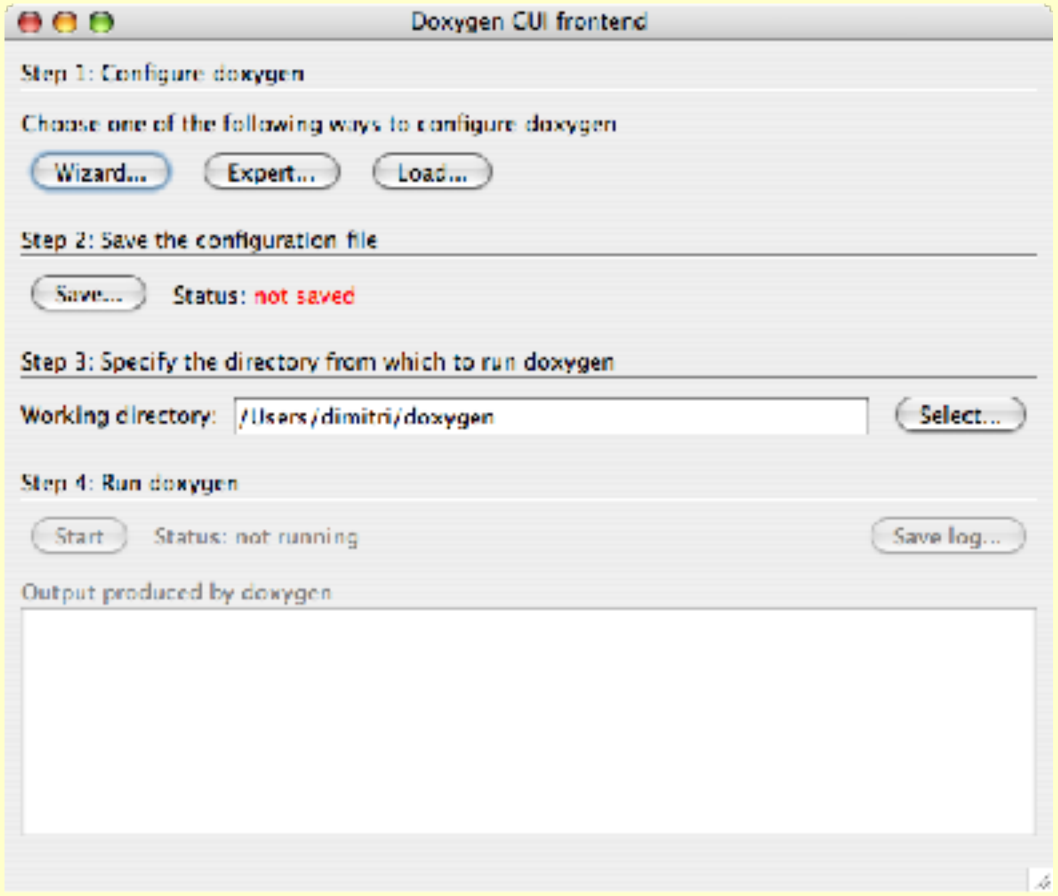
```
public tShirt (int size) {

}
```

```
/**
 * Enables wearer to decrease
 * rate of warmth loss, halt
 * embarassment growth.
 * Increases fashion rating.
 *
 * @param size Size of the
 *        shirt.
 */
```

Doxygen http://www.stack.nl/~dimitri/doxygen/

Doxywizard is a GUI front-end for configuring and running doxygen. When you start doxywizard it will display the main window (the actual look depends on the OS used).

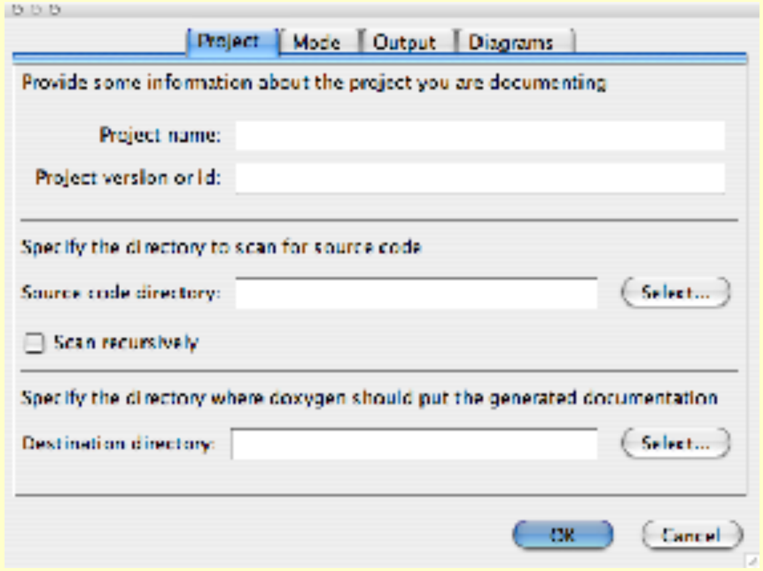

Doxygen GUI frontend
Step 1: Configure doxygen
Choose one of the following ways to configure doxygen
Wizard...
Expert...
Load...
Step 2: Save the configuration file
Save...
Step 3: Specify the directory from which to run doxygen
Select...
Step 4: Run doxygen
Start
Save log...
Output produced by doxygen

Status: not saved

Working directory: /Users/dimitri/doxygen

Status: not running

Doxywizard



Project | Mode | Output | Diagrams

Provide some information about the project you are documenting

Project name:

Project version or Id:

Specify the directory to scan for source code

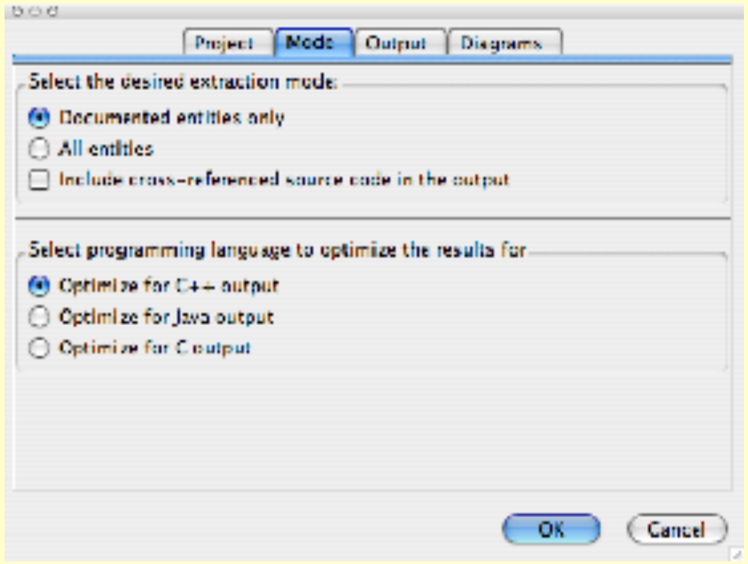
Source code directory: Select...

☐ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: Select...

OK Cancel



Project | Mode | Output | Diagrams

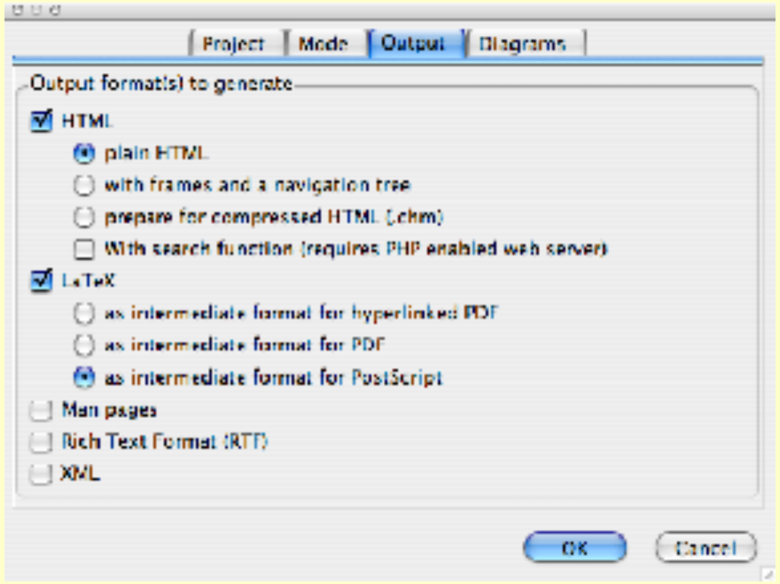
Select the desired extraction mode:

● Documented entities only
○ All entities
☐ Include cross-referenced source code in the output

Select programming language to optimize the results for

● Optimize for C++ output
○ Optimize for Java output
○ Optimize for C output

OK Cancel

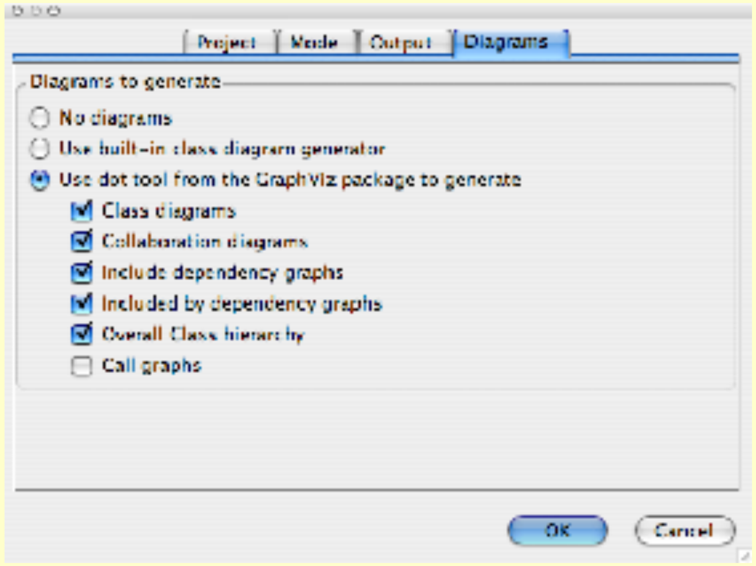


Project | Mode | Output | Diagrams

Output format(s) to generate

☑ HTML
  ● plain HTML
  ○ with frames and a navigation tree
  ○ prepare for compressed HTML (.chm)
  ☐ With search function (requires PHP enabled web server)
☑ LaTeX
  ○ as intermediate format for hyperlinked PDF
  ○ as intermediate format for PDF
  ● as intermediate format for PostScript
☐ Man pages
☐ Rich Text Format (RTF)
☐ XML

OK Cancel



Project | Mode | Output | Diagrams

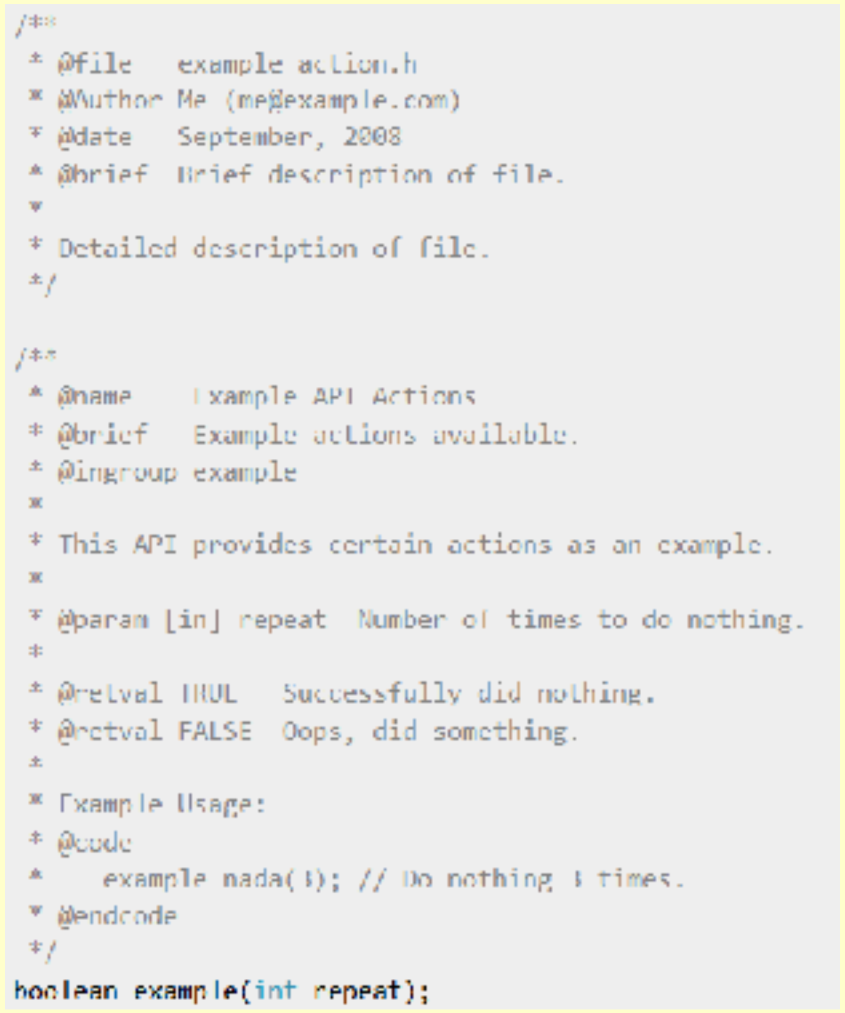
Diagrams to generate

○ No diagrams
○ Use built-in class diagram generator
● Use dot tool from the GraphViz package to generate
  ☑ Class diagrams
  ☑ Collaboration diagrams
  ☑ Include dependency graphs
  ☑ Included by dependency graphs
  ☑ Overall Class hierarchy
  ☐ Call graphs

OK Cancel

Doxygen example C code

http://stackoverflow.com/questions/51667/best-tips-for-documenting-code-using-doxygen



```
/**
 * @file    example_action.h
 * @Author  Me (me@example.com)
 * @date    September, 2008
 * @brief   Brief description of file.
 *
 * Detailed description of file.
 */

/**
 * @name     Example API Actions
 * @brief    Example actions available.
 * @ingroup example
 *
 * This API provides certain actions as an example.
 *
 * @param [in] repeat  Number of times to do nothing.
 *
 * @retval TRUE   Successfully did nothing.
 * @retval FALSE  Oops, did something.
 *
 * Example Usage:
 * @code
 *    example_nada(3); // Do nothing 3 times.
 * @endcode
 */
boolean example(int repeat);
```

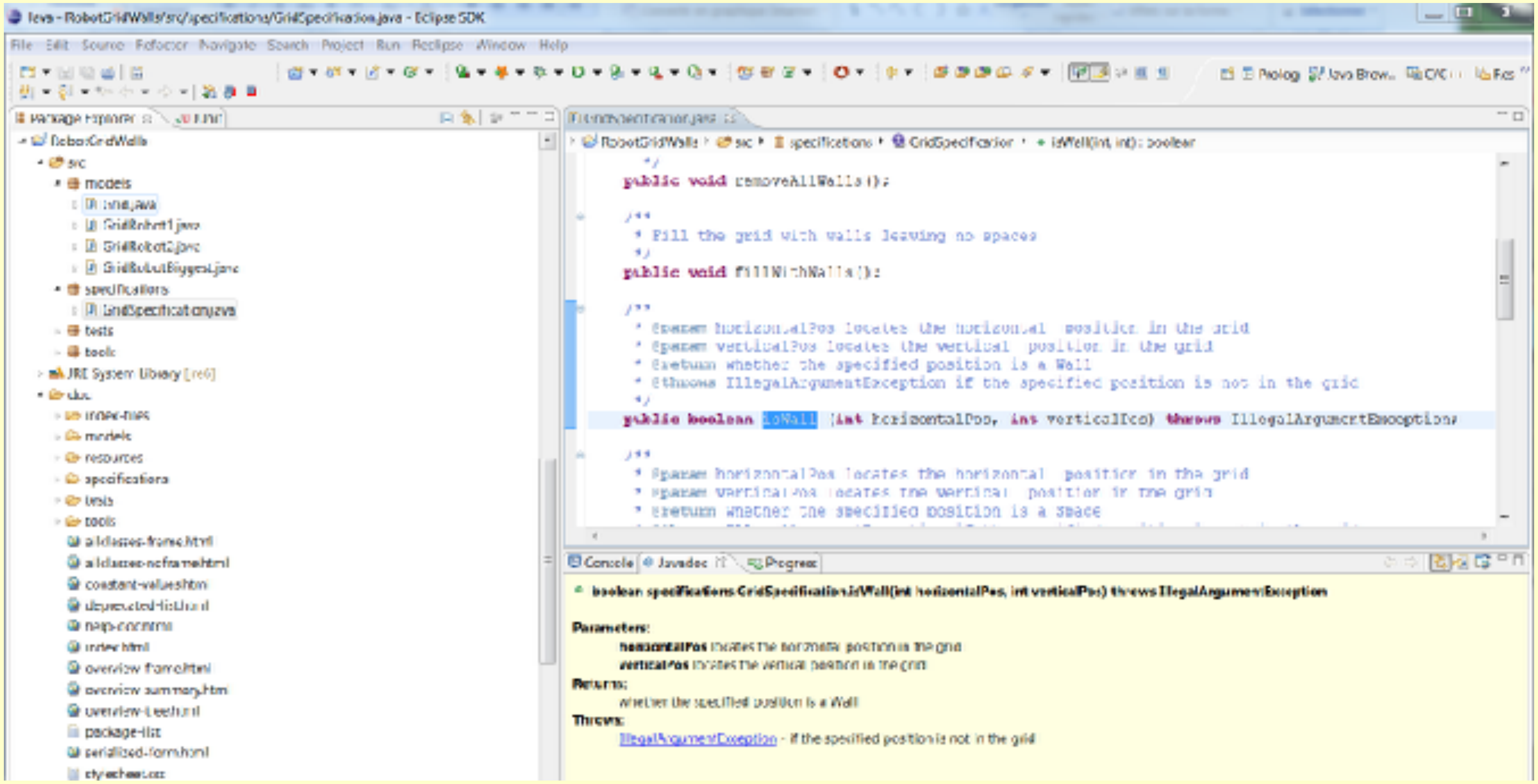
Full command list - http://www.stack.nl/~dimitri/doxygen/manual/commands.html

Javadoc

Good on-line tutorial

http://www.cs.laurentian.ca/aaron/cosc1047/eclipse-tutorials/javadoc-tutorial.html

Example in Eclipse (Robot problem in Java)



Java - RobotGridWalls/src/specifications/GridSpecification.java - Eclipse SDK

RobotGridWalls ▸ src ▸ specifications ▸ GridSpecification ▸ isWall(int, int) : boolean

```
     */
    public void removeAllWalls();

    /**
     * Fill the grid with walls leaving no spaces
     */
    public void fillWithWalls();

    /**
     * @param horizontalPos locates the horizontal  position in the grid
     * @param verticalPos locates the vertical  position in the grid
     * @return whether the specified position is a Wall
     * @throws IllegalArgumentException if the specified position is not in the grid
     */
    public boolean isWall (int horizontalPos, int verticalPos) throws IllegalArgumentException;

    /**
     * @param horizontalPos locates the horizontal  position in the grid
     * @param verticalPos locates the vertical  position in the grid
     * @return whether the specified position is a Space
```

boolean specifications.GridSpecification.isWall(int horizontalPos, int verticalPos) throws IllegalArgumentException

Parameters:
  horizontalPos locates the horizontal position in the grid
  verticalPos locates the vertical position in the grid
Returns:
  whether the specified position is a Wall
Throws:
  IllegalArgumentException - if the specified position is not in the grid

Documenting Code : Some Good Habits

- Cross reference test code with code being tested
- Link tests to requirements and design documentation
- Always keep documentation up to date as the rest of the system evolves
- Use version control on documentation
- Always use a documentation support tool/plugin
- Use 2 types of comments:
    1. External – explaining to users of the code how to (re)use it
    2. Internal – explain to maintainers of the code how it works and justify implementation decisions

Documenting Code: some PBL

Problem Specification
Implement a function (F) whose input is an integer and whose output is a single integer digit, where F is defined as follows:
F(x) = x, if x is a single digit
otherwise,
F(x) = F(x'), where x' is the sum of all the digits of x.
Examples:

F(4) = 4,
F(12) = F(1+2) = F(3) = 3,
F(79234) = F(25) = F(7) = 7.

Document your code and tests using **doxygen** or **Javadoc**