

# Literate Programming

Using Doxygen For TPF C & C++

Edwin W. Van de Grift



# Rationale

- C has been around for 30+ years
- C++ for 15+ years
- There's a world of experience that are readily usable for us (the TPF community)
  - ▶ There is no need to experience these 30+ years ourselves



# What is Literate Programming?

- I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: "Literate Programming."

Donald Knuth



# The Goal of a Literate Program

- Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

Donald Knuth



# Literate Programmers

- The practitioner of literate programming can be regarded as an essayist, whose main concern is with exposition and excellence of style. Such an author, with thesaurus in hand, chooses the names of variables carefully and explains what each variable means.

Donald Knuth



# Traditional Programs

- A traditional computer program consists of a text file containing program code. Scattered in amongst the program code are comments which describe the various parts of the code.

Ross Williams



# Literate Programs

- In literate programming the emphasis is reversed. Instead of writing code containing documentation, the literate programmer writes documentation containing code.

Ross Williams



# Sounds Good, but Now What?

- A well coded program does only need comments in places where non-trivial logic gets executed
- A function or class method implementation should read like a story
- Don't all programmers hate writing comments and documentation?
  - ▶ Program such that comments are superfluous!



# Variable and Function Naming

- Names of variables and functions tend to be longer when making them meaningful and descriptive, however:
- Rule of thumb:
  - ▶ Any line of code gets read far more often than it gets written or modified
  - ▶ The source code read/write ratio usually exceeds 100



# What Does This Buy Us?

- Quality
- Maintainability
- Readability
- Less discrepancies between code and documentation
- Motivation



# Examples

```
const int maxNumberOfPassengers(42);
```

```
Customer* getCustomer(const char* const customerName);  
Customer* getCustomer(const CustomerId id);
```

```
static int i;
```

```
static int l_st_uliI;
```



# No Comments?

- One very essential piece of code documentation remains: the interface
  - ▶ Documented in the header file
  - ▶ Any developer that needs to call a function should not have to look anywhere else but in the function's header file
    - Encapsulation



# Doxygen

- Open source documentation system for software written in C++, Java, IDL, C and more

<http://www.doxygen.org/>



# Generated Documents

- On-line documentation browser (in HTML)
  - ▶ Search engine
- Off-line reference manual (RTF, PostScript or LaTeX)
- Unix man pages
- And more



# From Your Source Code

```
#include <iostream.h>

int main(int argc, char* argv[]) {
    cout << "Hello world";
}
```



<http://yourIntranet.com/yourProject>



# JavaDoc

- Tool for generating API documentation in HTML format from doc comments in source code

```
/**  
 * This is the description part of a doc comment  
 *  
 * @tag Comment for the tag  
 */
```

<http://java.sun.com/j2se/javadoc/index.html>



# JavaDoc Comment Blocks

- Comment blocks
  - ▶ Brief description
  - ▶ Detailed description

```
/**  
 * Brief description. Details after the first dot...  
 * Bla bla bla  
 */  
void doSomething(const int i);
```



# JavaDoc Function Comments

- Function comments
- Parameters & return values

```
/**  
 * Does something.  
 * @param i The value that blabla...  
 * @param s The tex string that blabla...  
 * @see doSomethingRelated()  
 * @return An integer value containing the result.  
 */  
int doSomething(const int i, const char* const s);
```



# Graphs and Diagrams

- Included in doxygen output
  - ▶ For example
    - Class relationship diagrams
    - Include file relationships
  - ▶ Open source graph drawing software
    - Optionally used by/with Doxygen
    - Graphviz

<http://www.research.att.com/sw/tools/graphviz/>



# Doxygen Demo

- Installed Doxygen
- Installed Graphviz
- Created sample files
- Ran Doxygen (doxywizard)
- We'll now look at the resulting web site
  - ▶ "Palm Springs Sample Project"



# Thanks!

Edwin W. Van de Grift  
IBM TPF Services and Education  
edwin\_vandegrift@nl.ibm.com



Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. Other company, product and service names may be trademarks or service marks of others.