Documentation (Presentation #5)

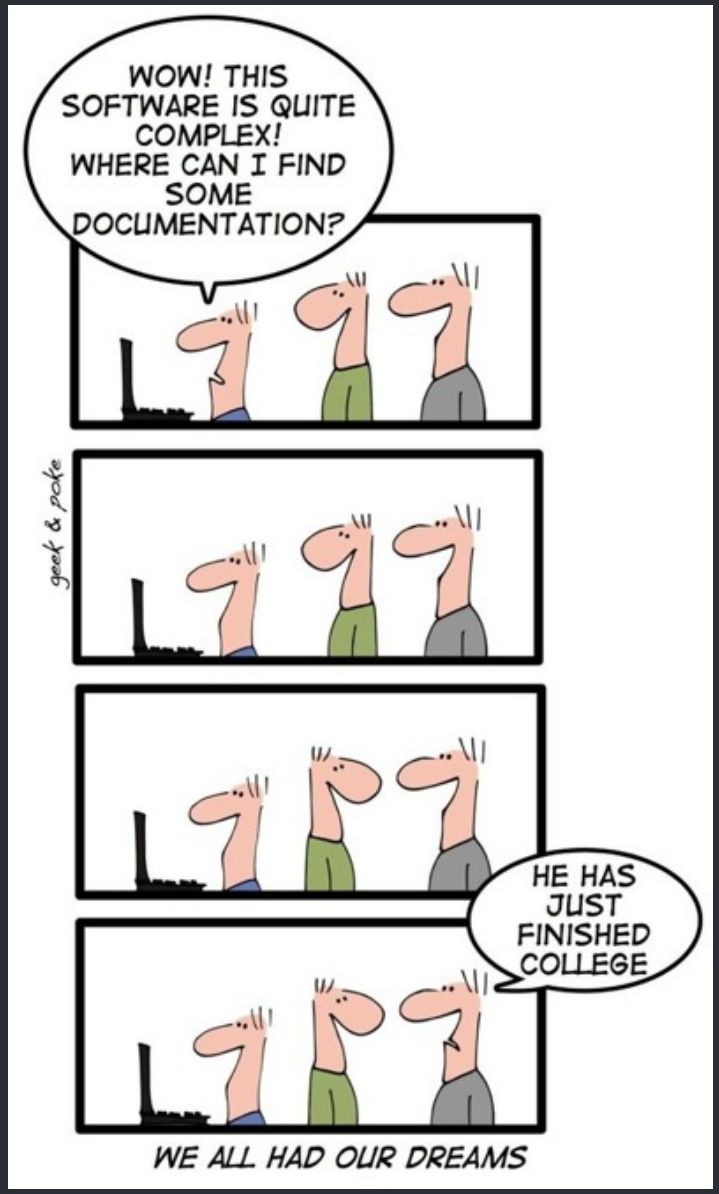
Special Topics (CS3001)

Illya Starikov

A Brief Introduction

One of the most critical aspects of computer science is documentation.

A Brief Introduction



A Brief Introduction

- Documentation was one of those things discussed once briefly in Introduction to Programming. You were expected to do it, but never really taught how to do it.
- One of my classes this semester (Object Oriented Numerical Modeling) require *a lot* of documentation.
- Throughout the semester, I wanted to write good, thorough documentation.

Prior Knowledge

- Mediocre documentation is easy to write.
  - This is the kind of documentation I wrote my freshmen year.
- Great documentation is much harder to write.
  - This is documentation I strive to write now.

Tools

- Often, it's hard to read documentation when it'd directly inside code.
- This is why I've picked up Doxygen, a tool to scrap all documentation outside of a codebase.

Tools (Example)

file:///Users/starboy/Desktop/ninjas/2018-sp-a-hw6-isgx2/doc/html/class_matrix.html

Eigen

A Linear Algebruh Library

Main Page | Classes | Files | Search

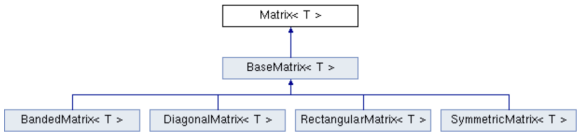
Public Member Functions | List of all members

Matrix< T > Class Template Reference abstract

An interface for a Matrix. More...

`#include <matrix.h>`

Inheritance diagram for Matrix< T >:



Public Member Functions

virtual void	**resize** (const size_t rows, const size_t columns)=0
virtual T &	**operator()** (const int row, const int column)=0
virtual const T &	**operator()** (const int row, const int column) const =0
virtual T &	**zero** () const noexcept=0
virtual size_t	**rows** () const noexcept=0
virtual size_t	**columns** () const noexcept=0

Detailed Description

template<class T>
class Matrix< T >

An interface for a Matrix.

Template Parameters

T The type of the elements the matrix stores.

The documentation for this class was generated from the following file:

- lib/matrix.h

Generated by doxygen 1.8.14

Goals

- To be able to use Doxygen efficiently.
- To write documentation that's easy to read, but also informative.

Before — Intro To Programming

```
// Description:     To get user input to dictate what menu option
     is selected.
// Pre-Condition:   A valid character must be entered.
// Post-Condition:  A value for charMenuOption is received,
// dictating what functions are used form there.
```

After — Object Oriented Numerical Modeling

```
/**
 *  @brief Read the specified input file and ensure proper
 *         data format.
 *
 *  @param filename The filename where file is to be read from.
 *         Can be relative or absolute.
 *
 *  @return An `std::string` with the entirety of the file
 *          contents in them (`n` and all).
 *
 *  @warning This does terminate prematurely if the file
 *          contents is not in the proper format.
 **/
```

Resources

- A great starting point was a Fermilab introduction article.
- From there, I would reference:
  - The doxygen website.
  - The special doxygen command guide.
- I would also read articles about better documentation.
  - HackerNoon.
  - Documentation Handbook.
  - Hacker News.

Goal Accomplishment

- So far, this semester, I have only missed two points on documentation for all of my classes.
  - Some of my graders even commented on my documentation ability (see attached).
- I have been more conscious about my technical writing.
- Overall, I saw an improvement in my documentation skills. This is reflected in my school work and my work life.

In Closing

All question, comments, and insults can be directed towards me:

- starikov@mst.com
- Illya Starikov
- Illya Starikov
- FreneticArray.com