

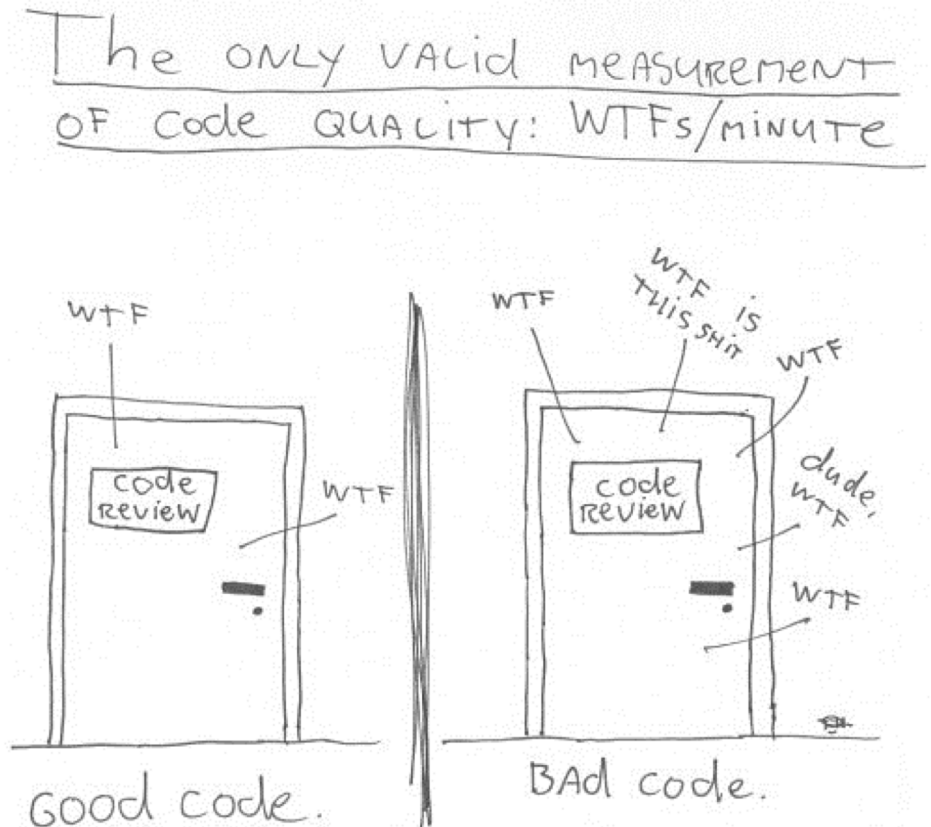
IMD0030

LINGUAGEM DE PROGRAMAÇÃO I

Aula 10 – Documentação com Doxygen

A importância de boas práticas em programação

- Um código fonte, ainda que esteja ruim, pode funcionar
- Perdem-se horas incontáveis e recursos importantes por conta de um código mal escrito (mas que muitas vezes funciona)
- Um dos mecanismos para aumento na qualidade, compreensão, manutenção e evolução de código fonte por meio de **documentação**
 - A principal forma de documentação é por meio de comentários



Comentários

Bons comentários

- Comentários sucintos e que expliquem o porquê de alguma coisa no código
- Pendências a serem futuramente implementadas (TODOs) ou corrigidas (FIXMEs)
- Ênfase a alguma coisa que não pode passar despercebida
- Melhor explanação do funcionamento de alguma instrução, função, etc. cuja compreensão não seja imediata por um ser humano
 - Explicação de decisões de implementação
- Explicação de precondições, restrições e limitações



Comentários

Maus comentários

- Comentários com erros de escrita (ortografia e/ou gramática)
 - Comentários são lidos apenas por seres humanos e completamente ignorados por computadores
- Comentar tudo, incluindo aquilo que é óbvio demais (redundância)
 - Mais comentários que código em si é altamente prejudicial
- Comentários muito extensos, verbosos
- Comentários que não têm a ver com o que o código faz
 - Atentar para o caso de comentários tornarem-se obsoletos com relação ao código



Geração automática de documentação

- Produzir documentação de qualidade, ao mesmo tempo que é fundamental no (re)uso de *software*, pode ser **uma tarefa relativamente difícil e demandar certo tempo**
- **Ferramentas para geração automática de documentação** podem auxiliar o desenvolvedor na tarefa de produzir uma boa documentação de *software*
 - Permitem que o desenvolvedor se concentre na elaboração da documentação do código (conteúdo), ao invés da aparência final de tal documentação
 - Mecanismos de **marcações** (ou **anotações**) no código fonte são utilizados pela ferramenta para gerar e organizar automaticamente a documentação final

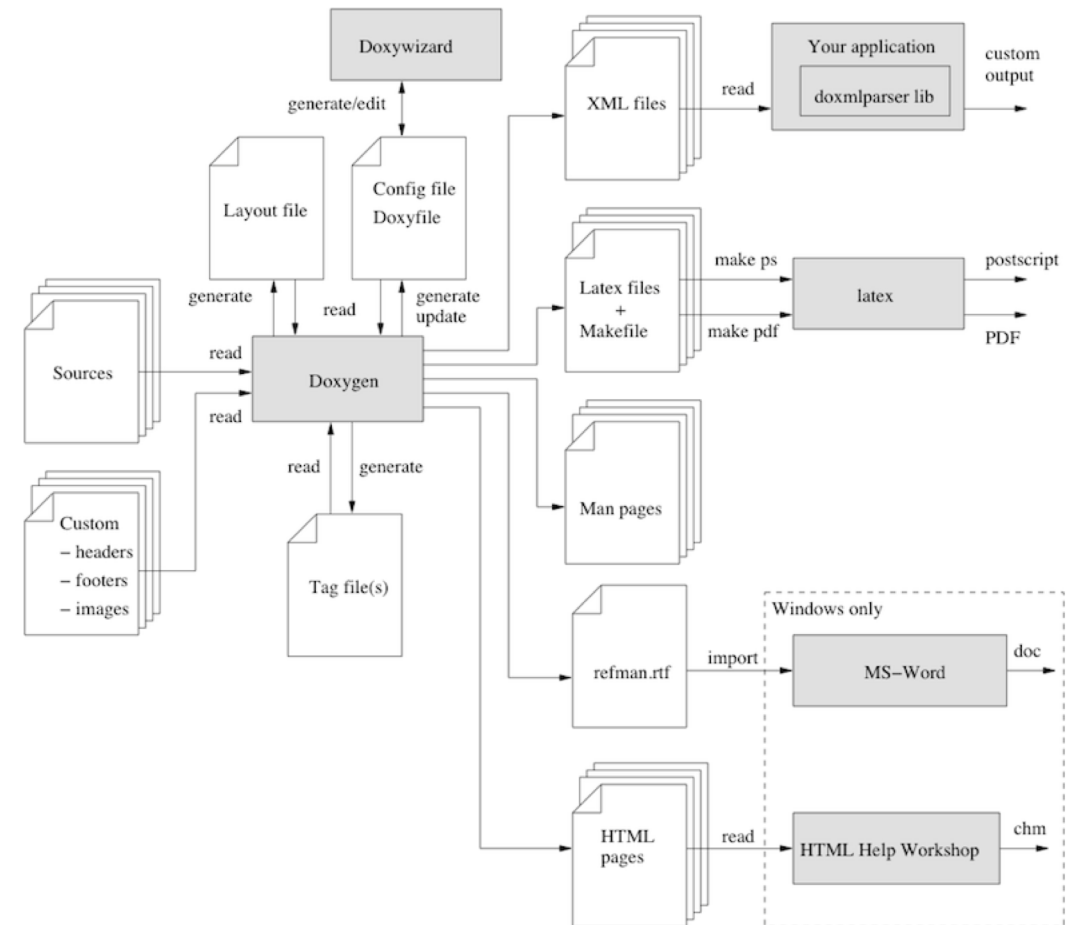
Doxygen



- Ferramenta para geração automática de documentação de programas implementados nas linguagens de programação C, C++, Java, Objective-C, Python, PHP, etc.
- Disponível para os sistemas operacionais Windows, Linux e Mac OS
- Disponível em: <http://www.doxygen.org/>

Como o Doxygen funciona

- Entrada:
 - **Arquivo de configuração (*Doxyfile*)** com um conjunto de opções que especificam alguns parâmetros referentes ao projeto em questão e como o Doxygen irá se comportar
 - **Arquivos de código fonte** comentados seguindo o estilo reconhecido pelo Doxygen
 - **Arquivos adicionais** (opcionais) para maior customização
- Saída: documentação na forma de páginas Web (HTML), LaTeX, RTF, XML e *man pages* para Linux



Anotando o código fonte (1)

- Um bloco de documentação no estilo Doxygen **difere ligeiramente** do padrão de comentário existente nas linguagens de programação C e C++, pela adição de **marcadores**
- Os marcadores (anotações) permitem ao Doxygen reconhecer que aquela parte do arquivo deve ser utilizada no momento em que a documentação é gerada
- Relação completa de marcadores:
<https://www.stack.nl/~dimitri/doxygen/manual/commands.html>
- O estilo de documentação mais utilizado para Doxygen é o **JavaDoc**:

```
/**  
 * ... texto ...  
 */
```

Anotando o código fonte (2)

Possibilidades mais usadas para documentação com Doxygen

- **Descrição breve:** comentário de única linha (anotação @brief)
- **Descrição detalhada:** comentário que pode se estender por várias linhas (anotação @details)

```
/**  
 * @brief Descricao breve  
 * @details      Descricao detalhada.  
 */
```

Anotando o código fonte (3)

Possibilidades mais usadas para documentação com Doxygen

- **Documentação de membros** de estruturas, uniões, classes ou enumerações:
 - Uso do símbolo < em um bloco de comentário logo após a declaração do membro, **ou**
 - Inclusão de uma descrição breve (anotação @brief) antes da declaração do membro

```
int qtAlunos;          /**< Define a quantidade de alunos. O valor maximo e... */
```

```
/** @brief Define a quantidade de alunos. O valor maximo e... */  
int qtAlunos;
```

Anotando o código fonte (4)

Possibilidades mais usadas para documentação com Doxygen

- **Documentação de funções** em termos de descrições breves e/ou detalhadas, além da de seus parâmetros de entrada e retorno
 - Documentação de parâmetro de entrada: anotação @param seguida do nome do parâmetro
 - Documentação de retorno: anotação @return

```
/**  
 * @brief Funcao que calcula o fatorial de um numero  
 * @param n Numero cujo fatorial sera calculado  
 * @return Fatorial do numero  
 */  
long double fatorial(long double n);
```

Anotando o código fonte (5)

Documentação do arquivo fonte propriamente dito

Marcador (anotação)	Descrição
@file	Documentação do arquivo fonte em questão
@author	Inserção do(s) nome(s) do(s) autor(es) do código fonte em questão. É possível separar os nomes de múltiplos autores por meio de vírgulas ou múltiplas anotações @author contendo o nome de um autor.
@since	Inserção da data de início da implementação
@date	Inserção de data (por exemplo, a data da última modificação do arquivo)
@version	Indicação da versão atual do arquivo
@sa	Inserção de referências cruzadas para classes, funções, métodos, variáveis, arquivos ou mesmo endereços da Internet

Anotando o código fonte (6)

Documentação do arquivo fonte propriamente dito

```
/**  
 * @file hello.cpp  
 * @brief Primeiro programa na linguagem de programacao C++  
 * @author Joao dos Anzois  
 * @since 01/01/2017  
 * @date 01/03/2017  
 * @sa http://www.google.com/  
 */
```

Gerando automaticamente a documentação

- Tendo-se o Doxygen devidamente instalado, o primeiro passo para a geração automática da documentação é criar o arquivo de configuração
 - No Linux, o Doxygen permite criar automaticamente o Doxyfile executando-se o comando `doxygen -g <nome do arquivo>`
 - Caso o nome do arquivo não seja fornecido, será criado um arquivo com o nome Doxyfile
 - O Doxyfile criado já contém uma série de opções (*tags*) estabelecidas por padrão, fazendo com que sejam necessárias pouquíssimas modificações
 - Relação completa de *tags*: <https://www.stack.nl/~dimitri/doxygen/manual/config.html>
- No Linux, a documentação é gerada automaticamente executando-se o comando `doxygen <arquivo de configuração>`
 - Caso o nome do arquivo tenha sido mantido com o padrão Doxyfile, basta executar `doxygen`

Gerando automaticamente a documentação

Exemplo de documentação gerada automaticamente com o Doxygen

IMC

Cálculo do índice de massa corporal (IMC) de uma pessoa e classificação do seu grau de obesidade

Página Principal	Arquivos	Busca
------------------	----------	-------

IMC Documentação

Gerado por  1.8.11

IMC


Cálculo do índice de massa corporal (IMC) de uma pessoa e classificação do seu grau de obesidade

Página Principal	Arquivos	Busca
------------------	----------	-------

Lista de Arquivos	Membros dos Arquivos
-------------------	----------------------

Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

 imc.cpp	Programa que calcula o índice de massa corporea (IMC) de uma pessoa
----------------------------------------------------------------------------------------------------	---------------------------------------------------------------------

Gerado por  1.8.11

Gerando automaticamente a documentação

Exemplo de documentação gerada automaticamente com o Doxygen

IMC

Cálculo do índice de massa corporal (IMC) de uma pessoa e classificação do seu grau de obesidade

Página Principal	Arquivos	Busca
Lista de Arquivos	Membros dos Arquivos	

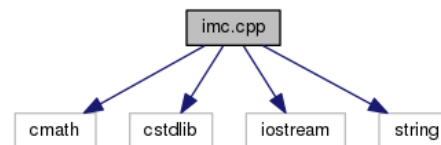
Referência do Arquivo imc.cpp

Funções

Programa que calcula o índice de massa corporea (IMC) de uma pessoa. Mais...

```
#include <cmath>
#include <cstdlib>
#include <iostream>
#include <string>
```

Gráfico de dependência de inclusões para imc.cpp:



Gerando automaticamente a documentação

Exemplo de documentação gerada automaticamente com o Doxygen

Funções

float	imc (float peso, float altura)	Funcao que calcula o IMC de um individuo adulto a partir do seu peso e de sua altura. Mais...
string	grau_obesidade (float imc)	Funcao que determina o grau de obesidade de um individuo com base em seu IMC. Mais...
int	main (int argc, char *argv[])	Funcao principal. Mais...

Descrição Detalhada

Programa que calcula o indice de massa corporea (IMC) de uma pessoa.

O IMC e uma medida internacional usada para calcular se uma pessoa esta no peso ideal. Essa medida e determinada pela divisão do peso da pessoa (em quilogramas) pelo quadrado de sua altura (em metros)

Autor

Pedro Paulo Pereira

Desde

01/01/2016

Data

01/02/2016

Gerando automaticamente a documentação

Exemplo de documentação gerada automaticamente com o Doxygen

```
float imc ( float peso,  
           float altura  
           )
```

Funcao que calcula o IMC de um individuo adulto a partir do seu peso e de sua altura.

Parâmetros

Peso em quilogramas

Altura em metros

Retorna

IMC do individuo

Gerado por [doxygen](#) 1.8.11

Take away message

“Any fool can write code that a computer can understand.
Good programmers write code that
humans can understand.”



Martin Fowler

Alguma Questão?

