МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет интеллектуальных систем и программирования

Кафедра «Программная инженерия» им. Л.П. Фельдмана

ОТЧЁТ

по дисциплине «Профессиональная практика программной инженерии»

Лабораторная работа № 4

ВАРИАНТ № 10

|  |  |
| --- | --- |
| Проверила:  асистент кафедры программной  инженерии им. Л.П. Фельдмана  Незамова Л. В. | Выполнил:  студент группы  ПИ-20а  Лазуренко Е. А. |

ДОНЕЦК – 2024

**Тема:** Создание самодокументирующегося кода

**Цель:** научиться добавлять в программный код специальным образом оформление докблок-комментарии, для последующей автоматической генерации API reference, а также познакомиться с форматом оформления документации DocBook.

**Вариант:**

10. Растровый редактор, с реализацией не менее 20-ти программных средств доступных в Adobe Photoshop не ниже версии CS3.

Данные созданного аккаунта GitHub: <https://github.com/MisterMLiL>

Созданный репозиторий PPPI-Raster-Editor: <https://github.com/MisterMLiL/PPPI-Raster-Editor>

Добавим докблок-комментарии во все существующие файлы, имитирующие разработку программного кода. Используем VS Code для создания докблок-комментариев для каждого файла, так как в нем можно удобно переключаться между файлами в папке.

Выберем один из файлов, который не является самым маленьким и не самый большой, и продемонстрируем его полный листинг с добавленными докблок-комментариями.

В качестве такого файла, используем «GUI\_Functionality.cpp»:

// GUI\_Functionality.cpp

#include "GUI\_Functionality.h"

/\*!

\brief Модуль: Графический интерфейс пользователя

\details В данном модуле содержатся функции для отображения главного окна программы и работы с документами.

\*/

void GUI\_Functionality::showMainWindow() {

/// \brief Метод для отображения главного окна программы

/// \details Этот метод отображает главное окно программы с панелями инструментов и меню.

}

void GUI\_Functionality::openDocument() {

/// \brief Метод для открытия документа

/// \details Этот метод позволяет пользователю открыть документ для просмотра и редактирования.

}

void GUI\_Functionality::saveDocument() {

/// \brief Метод для сохранения документа

/// \details Этот метод позволяет пользователю сохранить изменения в документе.

}

void GUI\_Functionality::exportDocument() {

/// \brief Метод для экспорта документа

/// \details Этот метод позволяет экспортировать документ в другие форматы.

}

void GUI\_Functionality::printDocument() {

/// \brief Метод для печати документа

/// \details Этот метод позволяет пользователю распечатать документ.

}

Сгенерируем справочное руководство для программистов автоматически с использованием специального инструмента Doxygen. Все комментарии были написаны в соответствии со структурой необходимой для Doxygen.

Для начала откроем doxywizard.exe. На странице Wizard заполняем информацию о проекте, что показано на рисунке 1.

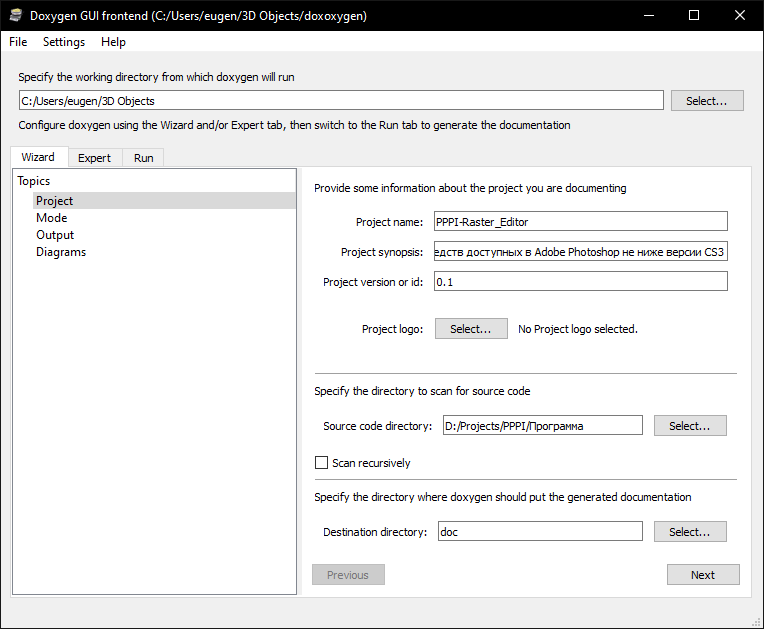


Рисунок 1 – Заполненная информация об проекте

Теперь перейдем в окно Run, и запустим создание документации (рисунок 2).

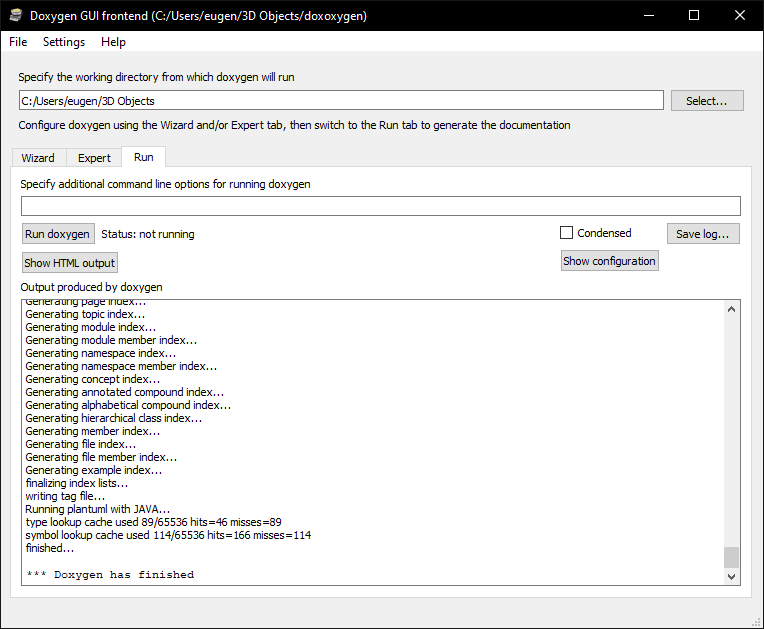


Рисунок 2 – Окно запуска создания документации

Теперь откроем файл документации нажав на «Show HTML output» (рисунок 3).

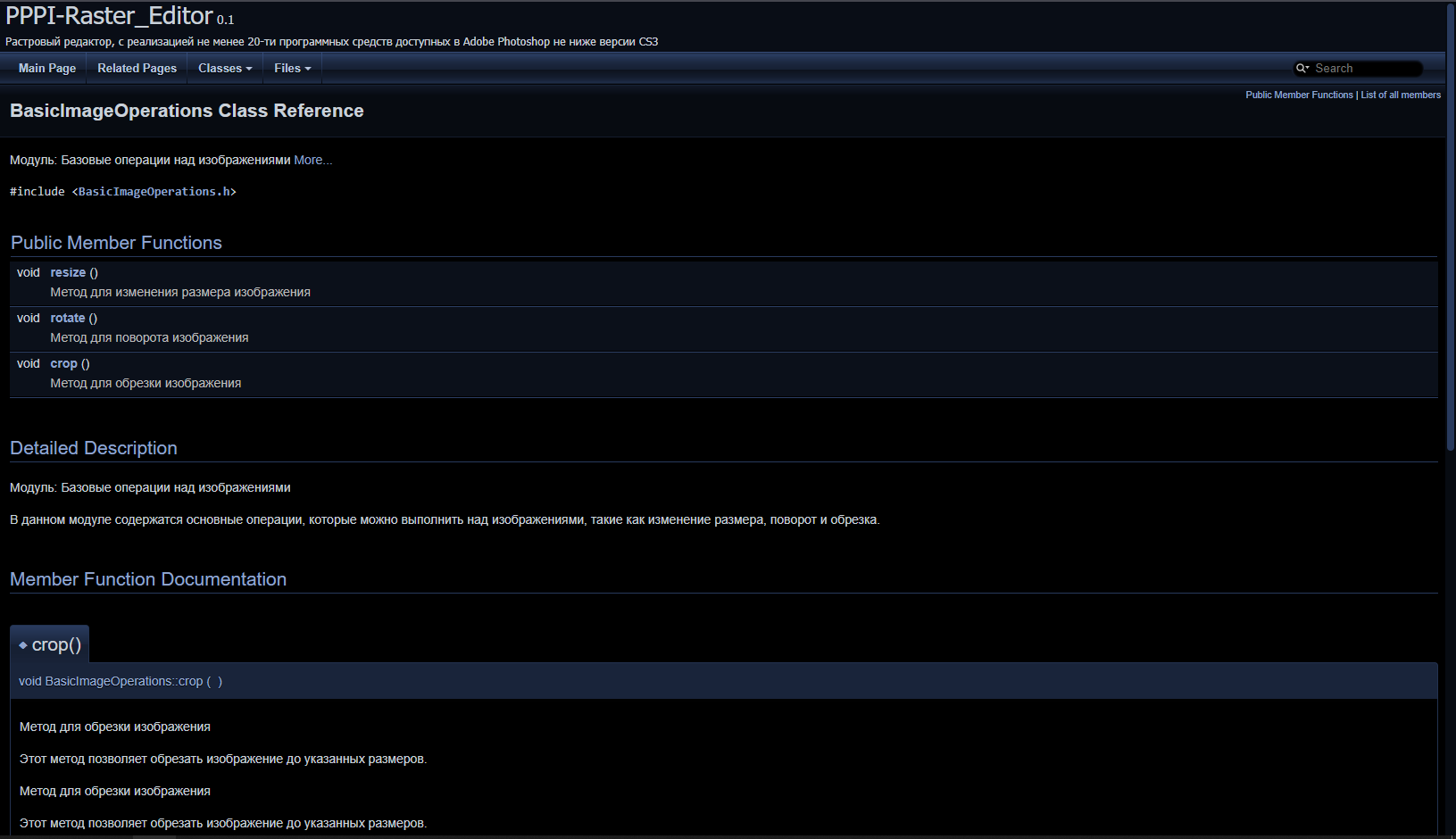


Рисунок 3 – Пример страницы со сгенерированной документацией

Теперь у нас есть готовая документация с описанием классов и методов этих классов.