МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Донецкий национальный технический университет»

Факультет ИСП

Кафедра ПИ им Л. П. Фельдмана

Лабораторная работа № 4

по курсу: «Профессиональная практика программной инженерии»

на тему: «Создание самодокументирующегося кода»

Проверил:

асс. каф. ПИ Филипишин Д. А.

Выполнил:

ст. гр. ПИ-20б

Кочетуров В. В.

Донецк – 2024

**Задание к лабораторной работе**

Внедрить в ранее разработанный программный код докблоки. Разработать DocBook и проверить корректность созданного DocBook файла с помощью средств редактора или одного из онлайн валидаторов.

**Тема для разрабатываемого проекта**

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

Добавим докблок-комментарии во все существующие файлы, имитирующие разработку программного кода.

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 «plugins.py»:

class PluginManager:

"""

Класс для управления плагинами, включая загрузку, получение и запуск.

"""

def \_\_init\_\_(self):

"""

Инициализирует экземпляр PluginManager.

"""

# ...

def load\_plugins(self):

"""

Загружает доступные плагины.

"""

# Загрузка плагинов

# ...

def get\_plugin(self, plugin\_name):

"""

Возвращает плагин по его имени.

Аргументы:

plugin\_name (str): Имя плагина.

Возвращает:

Plugin: Объект плагина, если он найден, иначе None.

"""

# Получение плагина по имени

# ...

def run\_plugin(self, plugin\_name, image):

"""

Запускает плагин с заданным именем и применяет его к

изображению.

Аргументы:

plugin\_name (str): Имя плагина.

image (Image): Изображение, к которому нужно применить плагин.

"""

# Запуск плагина

# ...

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.

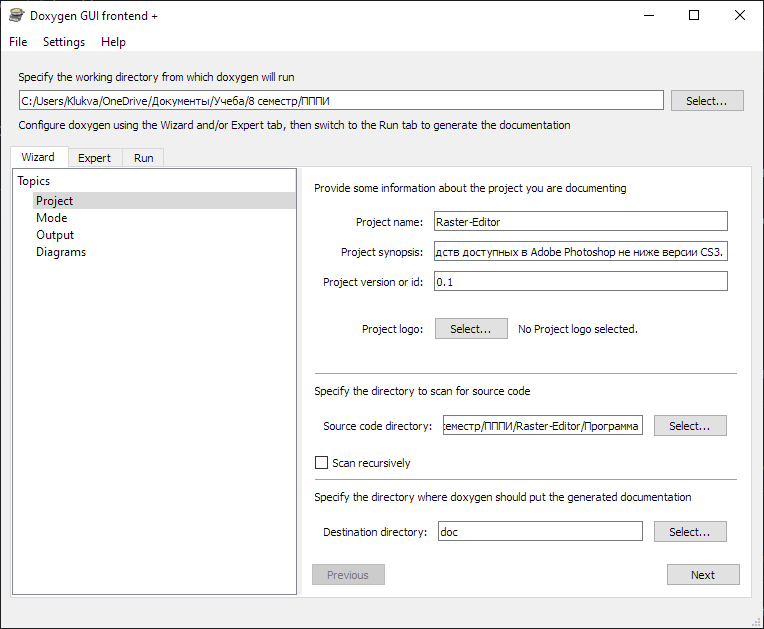


Рисунок 1 – Информация для генерации

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

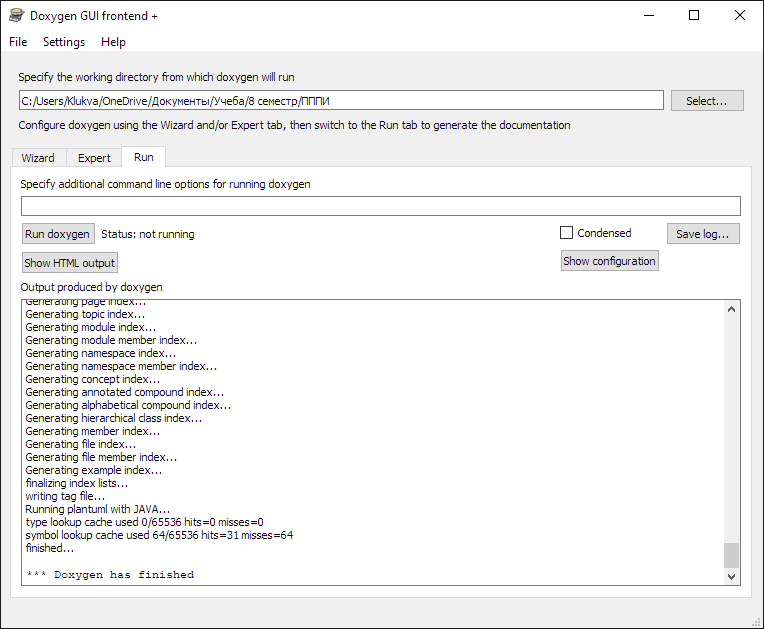


Рисунок 2 – Создание документации

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



Рисунок 3 – Сгенерированная документация