**Санкт-Петербургский государственный УНИВЕРСИТЕТ**

**Факультет прикладной математики и процессов управления**

**отчет**

**по практической работе №1**

**по дисциплине «Функциональное программирование»**

**на тему «Параллельная обработка изображений в многозадачной среде»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22Б15 |  | Мифтеев Р.Р. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

**[1. Цель работы 3](#_Toc24110)**

**[2. Задача 3](#_Toc18333)**

**[3. Алгоритм метода 3](#_Toc28774)**

**[4. Описание программы 3](#_Toc13884)**

**[4.1. Описание функций 4](#_Toc5594)**

**[5. Рекомендации пользователю 4](#_Toc20118)**

**[6. Рекомендации программисту 5](#_Toc7375)**

**[7. Контрольный пример 5](#_Toc2889)**

**[8. Вывод 7](#_Toc11543)**

**[9. Литература 7](#_Toc21992)**

# **Цель работы**

Разработка программы для параллельной обработки изображений с использованием многозадачности с обеспечением безопасного доступа к данным и эффективным использованием ресурсов.

# **Задача**

Разработать архитектуру программы для параллельной загрузки и обработки изображений с помощью нескольких фильтров: увеличение резкости, сепия, уменьшение размера.

# **Алгоритм метода**

Для каждого изображения создаётся новый поток, в котором происходит открытие изображения, применение каждого требуемого фильтра: увеличение резкости, сепия, уменьшение размера, и сохранение изменённого изображения. Заметим, что уменьшение размера будет происходить в первую очередь для оптимизации последующих фильтров.

После создания всех потоков происходит их параллельный запуск.

После завершения всех потоков программа завершает свою работу.

# **Описание программы**

Программа, разработанная в рамках лабораторной работы, представляет собой систему, основанную на концепциях объектно-ориентированного программирования. Python был выбран в качестве языка программирования, поскольку он предоставляет эффективные средства для реализации ООП-подхода, что способствует созданию чистого, модульного и легко поддерживаемого кода. Программа состоит из нескольких ключевых классов, каждый из которых отвечает за определённую функциональность. Взаимодействие между классами организовано с учётом принципов ООП, что обеспечивает чёткую структуру и управляемость кода.

# **Описание функций**

В программе используются 2 функции: 1 связанный с загрузкой изображений и 1 непосредственной обработкой. В таблице 4.1 представлено описание классов программы.

Таблица 4.1. Описание классов

|  |  |  |
| --- | --- | --- |
| Имя класса | Наследование | Описание класса |
| App | QWidget | Управление элементами форм.  Навигация по программе |
| Converter | - | Выполнение алгоритма генерации |

В таблице 4.2 представлено описание функций класса Converter

Таблица 4.2. Описание функций класса Converter

|  |  |  |
| --- | --- | --- |
| Имя | Входные данные | Описание |
| load\_images | Количество изображений,  папка, в которую будут сохранены изображения | Параллельное скачивание случайных изображений с помощью стороннего API |
| convert\_images | Фактор резкости, фактор уменьшения размера,  папка с исходными изображениями, папка, в которую будут сохранены обработанные изображения | Выполнение алгоритма генерации |

# 

# **Рекомендации пользователю**

Перед началом работы убедитесь, что в корневом каталоге проекта имеются все необходимые компоненты программы и установлены необходимые библиотеки, перечисленные в файле requirements.txt. В корневом каталоге проекта откройте config.ini файл, в котором укажите API ключ сервиса случайных изображений.

Запустите main.py файл и укажите необходимые конфигурации фильтров и директорию исходных изображений.

# **Рекомендации программисту**

Для запуска программы необходима 64-битная операционная система Windows и версия Python, совместимая с версией 3.10. Необходимые для работы программы библиотеки перечислены в файле requirements.txt. Основные из них: Pillow, requests, PyQt5.

# **Контрольный пример**

В данном разделе представлен контрольный пример, демонстрирующий способность программы.

 

Рисунок 7.1 – 7.2. Пример обработанного изображения

На рисунке 7.1 приведено исходное изображение. На рисунке 7.2 приведено обработанное с помощью данной программы изображение. Параметры фильтров: фактор резкости = 2, фактор уменьшения размера = 0.5.

# **Вывод**

В результате практической работы была разработана программа, способная эффективно обрабатывать изображения в многозадачной среде. Эта программа позволяет обрабатывать каждое изображение с использованием фильтров увеличения резкости, сепии и уменьшения размера. Кроме того, были учтены механизмы обработки ошибок и исключений, обеспечена безопасность при сохранении обработанных изображений и оптимизировано использование ресурсов.

# **Литература**

1. <https://random.responsiveimages.io/>
2. <https://pillow.readthedocs.io/en/stable/>
3. <https://api-ninjas.com/>