Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления

**Отчет №2**

по дисциплине «Функциональное программирование»

**Параллельные вычисления**

Автор работы: Дацык Р.В.

Группа: 22.Б15-пу

Преподаватель: Киямов Ж. У.

Санкт-Петербург, 2023

**Оглавление**

[**1.** **Цель работы** 3](#_Toc151866055)

[**2.** **Задачи** 3](#_Toc151866056)

[**3.** **Теоретическая часть** 3](#_Toc151866057)

[**4.** **Алгоритм метода** 3](#_Toc151866058)

[**5.** **Описание программы** 4](#_Toc151866059)

[**6.** **Рекомендации пользователя** 5](#_Toc151866060)

[**7.** **Рекомендации программиста** 5](#_Toc151866061)

[**8.** **Контрольный пример** 5](#_Toc151866062)

[**9.** **Вывод** 6](#_Toc151866063)

[**10.** **Список использованной литературы** 6](#_Toc151866064)

# **Цель работы**

Разработать эффективную программу для параллельной обработки изображений, используя несколько фильтров, таких как фильтр увеличения резкости, фильтр сепии и фильтр уменьшения размера.

# **Задачи**

Для достижения цели поставлены следующие задачи:

1. Разработать архитектуру программы для эффективной многозадачной обработки изображений.
2. Обработать ошибки и исключения, которые могут возникнуть в процессе обработки изображений.
3. Эффективно распределить ресурсы, такие как процессорное время, между потоками или процессами

# **Теоретическая часть**

В данной задаче решается проблема параллельной обработки изображений с использованием многозадачности. Ключевые концепции включают в себя многозадачность, обработку изображений и эффективное управление ресурсами. Для параллельной обработки данных используются потоки или процессы.

В контексте обработки изображений, используется библиотека Pillow (PIL), предоставляющая инструменты для работы с изображениями, такие как фильтры увеличения резкости, сепии и изменения размера. Применение этих фильтров требует грамотного управления данными и обеспечения безопасности при параллельной обработке.

# **Алгоритм метода**

1) **Обработка изображений:**

1. Функция apply\_filters\_to\_image(image\_path, out\_folder, filters, lock): Реализует обработку отдельного изображения. Изображение открывается, к нему применяются выбранные фильтры (например, увеличение резкости, сепия) и результат сохраняется в указанной папке. Обработка выполняется в многозадачной среде с использованием блокировки для обеспечения безопасности доступа к данным.Функция sharpen(inp, out\_path): Открывает изображение, увеличивает резкость и сохраняет результат.

2) **Обработка в многозадачной среде:**

1. Функция apply\_filters(image\_folder, out\_folder, filters): Организует обработку изображений в многозадачной среде. Для каждого изображения в указанной папке создаются задачи с выбранными фильтрами. Обработанные изображения сохраняются в выходной папке..

3) **Обработка ошибок и исключений:**

1. В случае возникновения ошибок при открытии или сохранении изображения, выводится сообщение об ошибке, и программа продолжает обработку следующего изображения.

4) **Распределение ресурсов:**

1. Задачи обработки изображений распределяются между потоками с использованием ThreadPoolExecutor для оптимального использования процессорного времени.

5) **Безопасное сохранение:**

1. Обеспечивается безопасное сохранение обработанных изображений в отдельную папку, чтобы избежать конфликтов доступа к данным.

# **Описание программы**

Программа реализована на языке Python 3.10 с использованием библиотеки Pillow для обработки изображений. Основные компоненты программы включают в себя функции обработки изображений (sepia, sharpen, resize, process).

Архитектура программы предусматривает безопасное сохранение обработанных изображений в отдельную папку, а обработка ошибок и исключений осуществляется с целью обеспечения стабильной работы программы. Ресурсы, такие как процессорное время, распределяются эффективно между потоками или процессами для минимизации временных затрат.

Таблица 5.1. Описание методов

|  |  |
| --- | --- |
| Название функции | Назначение функции |
| sepia | Изменение цветового фильтра изображения |
| sharpen | Увеличение резкости изображения |
| resize | Изменение разрешения изображения |
| process | Объединение функций |

# **Рекомендации пользователя**

Для запуска программы достаточно указать путь к папке с изображениями, которые нужно обработать, а также путь к папке с результатом. Среди возможных фильтров выбрать нужные и нажать кнопку «Применить фильтры».

# **Рекомендации программиста**

Для запуска программы необходима 64-битная операционная система Windows, Linux или macOS. Для работы с кодом необходима среда разработки, совместимая с python 3.10.

Исходный код программы доступен по ссылке:

https://github.com/CapTopGrade/Functional\_Programming/blob/main/%D0%A4%D0%9F/%D0%A4%D0%9F%20%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0/%D0%A4%D0%9F%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9/%D0%A4%D0%9F%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9/PP%20lab%201.py

# **Контрольный пример**

В данном разделе представлен контрольный пример, демонстрирующий работу программы.

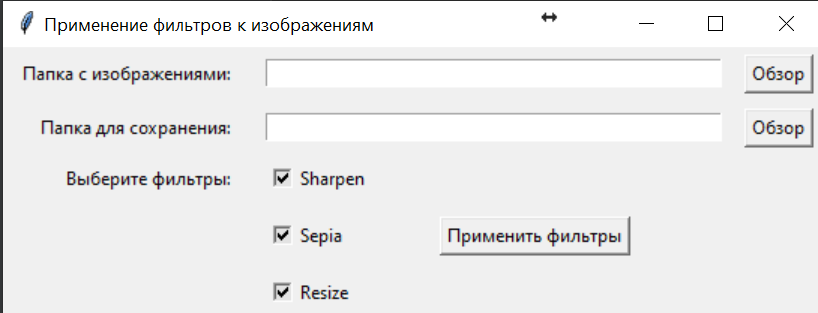


Рисунок 6.1, интерфейс программы

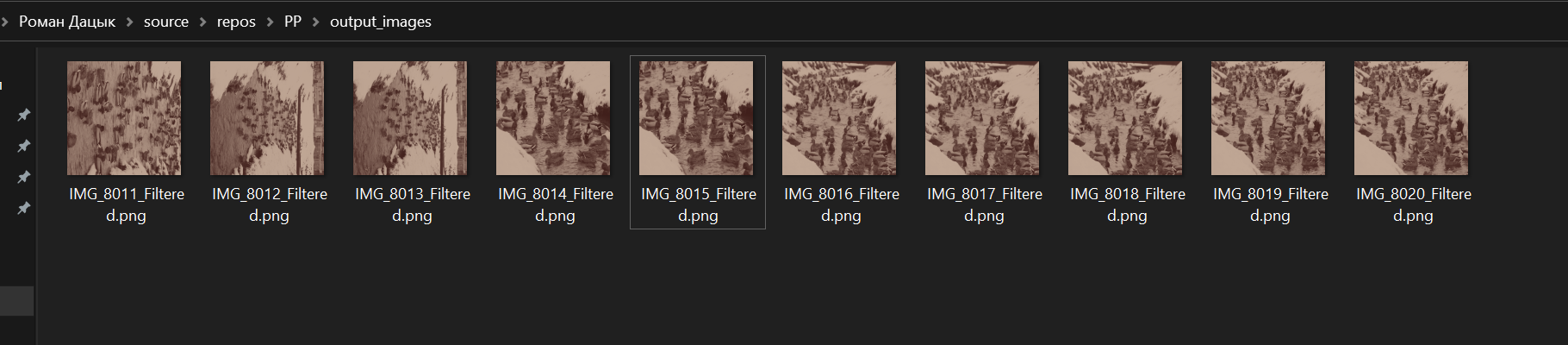


Рисунок 6.2, Пример результата работы программы

# **Вывод**

Разработана эффективная программа для параллельной обработки изображений с использованием многозадачности. Учтены особенности работы с разными фильтрами, обработка ошибок и распределение ресурсов для оптимальной производительности.

# **Список использованной литературы**

[1] Документация библиотеки Pillow (PIL): <https://pillow.readthedocs.io/en/stable/>

[2] Документация Python по многозадачности: <https://docs.python.org/3/library/concurrent.futures.html>

[3] Документация Python по обработке ошибок и исключений: <https://docs.python.org/3/tutorial/errors.html>