Спецификация архитектуры

1. Проблема:

Нужен способ составления мозаики из картинок-составляющих по картинке-шаблону, по принципу, замены группы пикселей картинки-шаблона на наиболее похожую картинку-составляющую

2. Функциональные требования:

2.1 Создавать мозаику из *проблемы* из ***картинок-составляющих*** по ***картинке-шаблону***.

2.2. Предоставлять способы составления мозаики:

2.2.1. Способ обрезки всех картинок-составляющих до оптимального размера (min(высота картинок-составляющих), min(ширина картинок-составляющих))

3. Организация программы:

представлена в файле arc.drawio проекта.

4. Основные классы:

представлены в файле arc.drawio проекта.

5. Организация данных:

Все картинки будут представлены в виде массива NumPy.

4. Пользовательский интерфейс: CLI.

5. Управление ресурсами:

картинки предварительно загружаются в оперативную память

6. Ввод-Вывод:

Вызов из терминала:

* argv[1] – путь до оригинальной картинки формата png.
* argv[2] – путь до папки с картинками-составляющими мозаики формата png.

7. Обработка ошибок:

* Простое обнаружение ошибок без корректировки данных.
* Активное обнаружение ошибок.
* Уведомлять пользователя об ошибке.

Список ошибок

\_…добавить…\_

* Передавать ошибку по цепи вызовов.

8. Отказоустойчивость:

Просто завершать выполнение программы при обнаружении ошибки, с выводом ошибки пользователю.

Основной алгоритм работы программы:

1. Считать все картинки-составляющие.
2. Передать обработку данных MosaicHarvesting:
3. mosaic\_by\_optimal\_size():
   1. Найти оптимальный размер картинки-составляющей (минимальная ширина и длинна из всех картинок).
   2. Обрезать все картинки-составляющие до оптимального размера.
   3. Составлять массив средних цветов каждой картинки-составляющей.
   4. Разбить основную картинку на части по оптимальному размеру.
   5. Брать средний цвет от каждой части и вставлять наиболее подходящую картинку-составляющую.
4. Вывести получившуюся картинку.