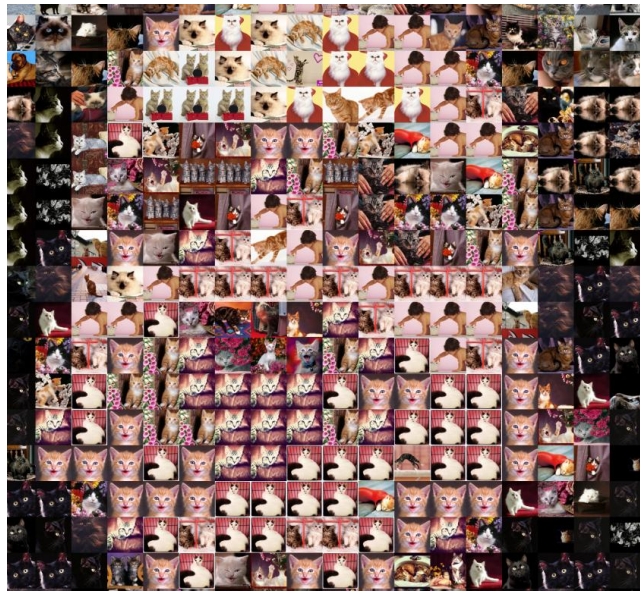
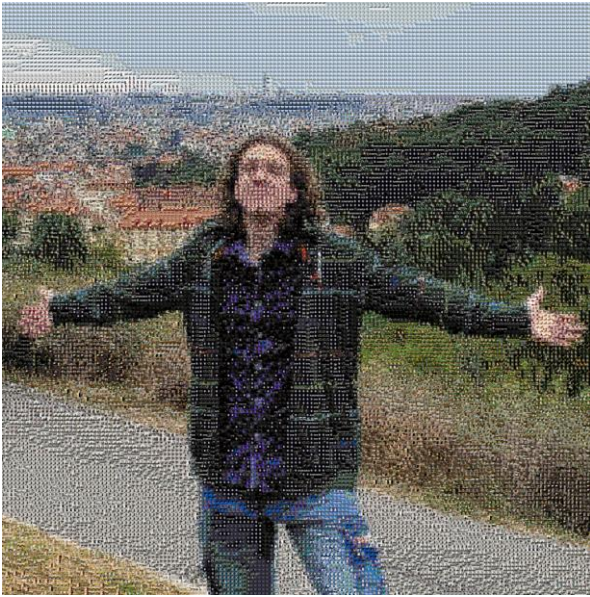


SRS



Общее описание функциональности

Необходимо разработать консольное приложение, которое будет создавать изображения, у которых каждый пиксель является другим изображением. Пример выше.

Функциональные требования

- 1) Приложение должно запускаться из консоли
- 2) Все входные параметры указываются также через консоль и при помощи флагов. Для работы с приложением изменения в файлы не должны вноситься
- 3) Пользователь должен указывать
 - 1) путь до оригинального изображения(обязательно) – без флага
 - 2) путь до директории, в которой находятся изображения, которые будут заменять пиксели(обязательно) – через флаг
 - 3) путь до итогового файла(обязательно) – через флаг
 - 4) размер изображения, которое будет заменять пиксель(необязательно, по умолчанию 50 пикселей) – через флаг
 - 5) разрешение изображения в ЗАМЕНЕННЫХ пикселях – через 2 флага: один для горизонтали, другой для вертикали(необязательно, по умолчанию итоговое разрешение по каждой стороне должно быть x10 от оригинального. При указании разрешения всего 1 из сторон вторая должна высчитываться, исходя из пропорций первой)

Обязательно/необязательно не в смысле обязательно/необязательно реализовывать, а в смысле обязательно/необязательно указывать этот флаг при работе с приложением

4) Приложение должно иметь возможность открыть справку через аргумент "?"

5) Соотношение сторон итогового изображения может отличаться от соотношения сторон оригинального изображения

6) При запуске генерации изображения пользователю должна выводиться информация о времени, которое займет генерация

7) Все константы, используемые в исходном коде приложения, должны быть вынесены в переменные окружения

8) Приложение должно иметь красивый и внятный README.md файл с описанием работы приложения, его функциями и скриншотами

9) Приложение должно быть написано с применением принципом ООП, SOLID

10) Приложение должно быть написано с применением паттерном проектирования. Для такого маленького проекта они могут быть излишни, однако в рамках обучения подумайте где их можно обоснованно применить. Исходите из того, что в дальнейшем это приложение будет сильно разрастаться. Будут добавляться новые способы взаимодействия(не только через консоль, а также, например, через веб страницу, через тг бота), будет расширяться функционал, будет меняться старый. Будет добавляться работа с базами данных, интеграция с инструментами мониторинга и визуализации.

Учитывайте все когда будете думать над архитектурой(по крайней мере на том уровне, на котором можете)

Нюансы реализации

1) Обработка аргументов командной строки должна производиться библиотекой(ищем и выбираем сами)

2) Переменные окружения должны задаваться в файле .env и получается в коде при помощи библиотеки dotenv

Демонстрация результата

В результате у вас должен быть репозиторий на github со всем исходным кодом. Ветка main должна быть пустой. Проект должен писаться в ветке dev. Проверяться будет PULL REQUEST из ветки dev в ветку main.

В репозитории должен быть README.md файл, описанный выше. Также в репозитории должен быть отчет в формате Word/PowerPoint/MD, в котором вы со скриншотами/ссылками на строки на гитхабе расписываете ГДЕ и ПОЧЕМУ вы применили те или иные принципы ООП, SOLID и паттерны проектирования. При описании реализованных паттернов

обязательно указать почему вы применили здесь именно этот паттерн, как он может в будущем.

Задание должно быть выполнено ЗА СУТКИ до момента сдачи, т.к. мне необходимо заранее просмотреть репозитории и оставить внятные правки.