Рабочее название: **Photos from Windows spotlight**

**Релиз 0**

Общее:

Программа находит фото, которые появляются на экране блокировки Windows 10 и сохраняет их указанном месте.

Описание:

Фото, которые появляются на экране блокировки называется Windows: Интересное (Windows spotlight). Находятся фото по адресу: C:\Users\leoon\AppData\Local\Packages\Microsoft.Windows.ContentDeliveryManager\_cw5n1h2txyewy\LocalState\Assets

И находятся они в виде рабочих файлов типа:

3b2144242ed014c8229d80c84ff0d8e09b74627ced31e4619e0f518ba701a230

Без расширения.

Задача программы:

Взять эти файлы, скопировать в нужную папку, изменить название и добавить расширение .jpg

07.02.18

1) Разобраться, как в цикле можно перебирать фото, вернее проверять их размер не перегружая память. Т.е. если сейчас просто взять через Image в цикле, то, как я понимаю, просто каждая фото загружается в память, но ее не успевает гарбич коллектор выгрузить из памяти и память переполняется. (решил с помощью Bitmap)

10.02.18

# Task 0001

- Git

Изменяю задачу, хотел перезаписывать файлы. Но нужно сделать информацию о файлах, скопированных в папку с картинками. Я это сделаю через файл xml, т.к мне кажется «дорогим» использовать здесь SQL, не требуется много данных.

При каждом запуске программы, будет читаться этот файл. Файл XMLData.xml. Если файла нет, он будет создаваться сразу с нужной структурой, если файл есть, то будут читаться данные из файла (в отдельном потоке) и эти данные будут находится в коллекции. (не исключено, что далее в место файла сделаю локальную SQL.

Формат файла xml:

<Фото скопированное>

<Имя> </Имя>

<Размер> </Размер>

<Дата файла> </Дата файла>

<Предыдущее имя> </Предыдущее имя>

</Фото скопированное>

<>

<>

<>

<>

(+ 15.02.2018)

* Делаю проверку, есть ли в директории файл с данными, т.е. «data.xml». Если файл существует, то мы пишем в него, если не существует, то создаем его.
* Копируем все данные из файла в коллекцию типа List<> параметризированную классом с данными.
* Записываем данные фотографий в файл.
* При не первом запуске сверяем обнаруженные файлы и файлы из файла коллекции (нужно еще добавить определение имен файлов в папке назначения)

- резервная копия заменённых файлов.

\* полностью меняю все, обдумав задачу

03.03.2018

- сравнение фото, имеющихся в папке назначения и найденных.

Делаю список файлов, имеющихся в папке назначения и Или создаю «снимки фото», сравниваю их с новыми, Или каждую новую по очереди с каждой имеющейся сравниваю. Сравниваю только фото с одинаковой датой фото.

11.03.2018

- Нужно добавить выведение найденных картинок и отобранных, чтоб понятно было, что найдено и что добавлено, на случай сбоя сравнения.

**Task 0001 End**

# Task 0002

1. Добавляем графический интерфейс WPF.
2. Хранить данные о всех найденных фото до этого в виде их уменьшенного хэша изображения и названия, которым его сохранили, дате обнаружения.
   1. При запуске проверяем есть ли файл конфигурации, если есть читаем его. Это скорей всего нужно делать в отдельном потоке (или асинхронно, но не точно).
   2. Если нашли файл конфигурации и в нем есть адрес, куда до этого сохраняли файлы картинок, то в (?)Следующем потоке сверяем перцептивный хэш с полученным хэшем текущих картинок.
   3. (?)Вторым потоком сканируем папку с картинками в системной директории.
   4. Оптимизировать, т.к. 2гб памяти на 250 картинок – это жесть!
3. НОРМАЛЬНОЕ ИМЕНОВАНИЕ ПЕРЕМЕННЫХ И МЕТОДОВ
4. Добавить файл со строкой загрузки
5. Сделать конфиг файл к программе

- Определить, что файл – это картинка

- Нужно изменить метод переименования фото.

- Добавить вывод информации с какими картинками совпадает и какое совпадение.

- портретный ландшафтные

- Проверить, есть ли обновления файлов.

- Изменить формирование имени. Год-месяц-день.

- Увеличить точность совпадений

- с увеличением точности увеличится и время перебора совпадений, значит нужно добавить многопоточность.

- отображение имени картинок, которые совпали.