МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 6

з дисципліни Об'єктно-орієнтоване програмування СУ

Тема: «Розробка графічного інтерфейсу для розрахункових завдань і побудови графіків »

ХАІ.301 . <спец.>. 322. 6 ЛР

Виконав студент гр. \_\_\_\_\_\_322\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кулагін Олексій\_\_\_

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ас.  В. О. Білозерський

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

# Отримати досвід роботи з навчальними матеріалами та документацією до

# бібліотек Pillow і OpenCV, і навчитися розробляти віконні додатки для

# завантаження з файлу, обробки різними способами, збереження і відображення

# у вікні фото-зображень..

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вивчити документацію до бібліотеки Pillow і написати скрипт

з визначенням класу, що реалізує користувальницький інтерфейс для виконання

наступних функцій:

1) відкриття файлу із зображенням будь-якого допустимого графічного

формату;

2) відображення зображення та інформації про формат;

3) \* Установка значень для виконання функцій 4-5;

4) створення зменшеної копії вихідного зображення;

5) геометричні перетворення мініатюри, фільтрація, перетворення

формату і вставка в вихідне зображення відповідно до варіанту (див. табл.1);

6) збереження зміненого зображення в фай і реалізацією роботи з об'єктом

цього класу для запуску віконного програми.

Завдання 2. Вивчити документацію до бібліотеки OpenCV і написати

скрипт з визначенням і роботою об'єктів класу, що реалізує

користувальницький інтерфейс для виконання наступних функцій:

1) відкриття файлу із зображенням будь-якого допустимого графічного

формату;

2) \* Установка значень для виконання функцій 3-4;

3) зміна розмірів зображення;

4) геометричні перетворення зображення, зміна колірного простору,

фільтрація і виконання операцій із зображенням відповідно до варіанту (див.

табл.2);

5) відображення вихідного зображення і після кожної зміни;

6) збереження змінених зображень у файли

і реалізацією роботи з об'єктом цього класу для запуску віконного програми.

# ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі process\_image\_pil

Вхідні дані (ім’я, опис, тип, обмеження):

f:

Опис: Об'єкт файлу зображення, відкритий у бінарному режимі.

Тип: file

Обмеження: Має бути зображенням у форматі JPEG або іншому підтримуваному форматі.

img:

Опис: Об'єкт зображення, завантажений з файлу.

Тип: PIL.Image.Image

Обмеження: Формат зображення має бути підтримуваним бібліотекою Pillow.

img\_width, img\_height:

Опис: Ширина та висота зображення відповідно.

Тип: int

Обмеження: Додатні цілі числа.

bg\_width, bg\_height:

Опис: Ширина та висота фону, що трохи більші за зображення.

Тип: int

Обмеження: Додатні цілі числа, більші на 20 пікселів від відповідних розмірів зображення.

position:

Опис: Координати вставки зображення по центру фону.

Тип: tuple (int, int)

Обмеження: Додатні цілі числа.

save\_path:

Опис: Шлях для збереження результуючого зображення.

Тип: file

Обмеження: Має бути валідним шляхом для збереження файлу у форматі PNG або іншому підтримуваному форматі.

Вихідні дані (ім’я, опис, тип):

bg\_img:

Опис: Фінальне зображення з фоном та вставленим по центру обробленим зображенням.

Тип: PIL.Image.Image

Текстові сповіщення:

"Error file open":

Опис: Виводиться у разі, якщо файл зображення не був обраний.

Тип: str

"Original image size: ...":

Опис: Виводиться розмір оригінального зображення.

Тип: str

"Original image format: ...":

Опис: Виводиться формат оригінального зображення.

Тип: str

"Image saved to: ...":

Опис: Виводиться шлях, за яким було збережено результуюче зображення.

Тип: str

Лістинг коду вирішення задачі наведено в дод. А.  
 Екран роботи програми показаний на рис. Б.

# ВИСНОВКИ

# Було вивчено роботу з бібліотеками Pillow та OpenCV для обробки зображень. Закріплено на практиці навички створення віконних додатків для завантаження, обробки, збереження та відображення фото-зображень. Відпрацьовано в коді програми різні методи обробки зображень, що дозволило отримати практичний досвід у застосуванні цих бібліотек.

# ДОДАТОК А

Лістинг коду програми до задач Func26

from tkinter.filedialog import askopenfile, asksaveasfile

from PIL import Image, ImageFilter

def process\_image\_pil():

    # Открываем файл в бинарном режиме используя диалог tkinter

    f = askopenfile(mode='rb', defaultextension=".jpg",

                    filetypes=(("Image files", "\*.jpg"), ("All files", "\*.\*")))

    if f is None:

        print("Error file open")  # Если файл не выбран

        return

    img = Image.open(f)  # Передаем объект файла

    print("Original image size:", img.size)  # Получаем размер изображения

    print("Original image format:", img.format)  # Выводим формат изображения

    # 1. Поворот на 180 градусов

    img = img.rotate(180, expand=True)

    # 2. Применение фильтра SMOOTH\_MORE

    img = img.filter(ImageFilter.SMOOTH\_MORE)

    # 3. Преобразование изображения в 32-битный формат с целыми числами

    img = img.convert("RGBA")

    # 4. Вставка изображения по центру

    img\_width, img\_height = img.size

    bg\_width = img\_width + 20  # Фон больше изображения на 20 пикселей по ширине

    bg\_height = img\_height + 20  # Фон больше изображения на 20 пикселей по высоте

    # Создаем фоновое изображение

    bg\_img = Image.new('RGBA', (bg\_width, bg\_height), (255, 255, 255, 255))

    # Вставляем изображение по центру фона

    position = ((bg\_width - img\_width) // 2, (bg\_height - img\_height) // 2)

    bg\_img.paste(img, position, img)

    # Отображаем результирующее изображение

    bg\_img.show()

    # Сохраняем результирующее изображение

    save\_path = asksaveasfile(defaultextension=".png",

                              filetypes=(("PNG files", "\*.png"), ("All files", "\*.\*")))

    if save\_path:

        bg\_img.save(save\_path)

        print("Image saved to:", save\_path.name)

    f.close()  # Закрываем файл

# Вызов функции

process\_image\_pil()

ДОДАТОК Б  
 Скрін-шоти вікна виконання програми Func26  
  
  
