

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт физики

**Направление подготовки: 10.03.05 – информационная безопасность
автоматизированных систем**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
Цифровая обработка изображений:
Геометрические преобразования**

**Студент 3 курса
группы 06-952**

Глазков А.Ю.

Научный руководитель

Корчагин П.А.

Казань – 2022

Лабораторная работа №1

Базовые операции ввода-вывода изображений

Цель: Освоение операций загрузки цифровых изображения, работы с графическими форматами файлов и вывода изображений на экран.

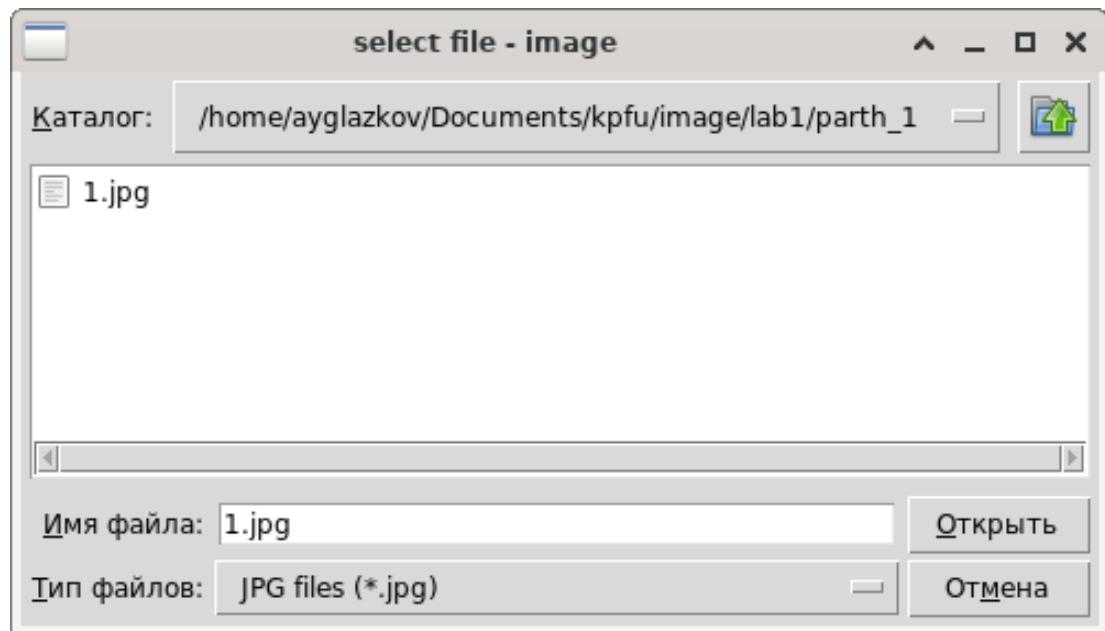
Использованный язык программирования: python.

Ход работы:

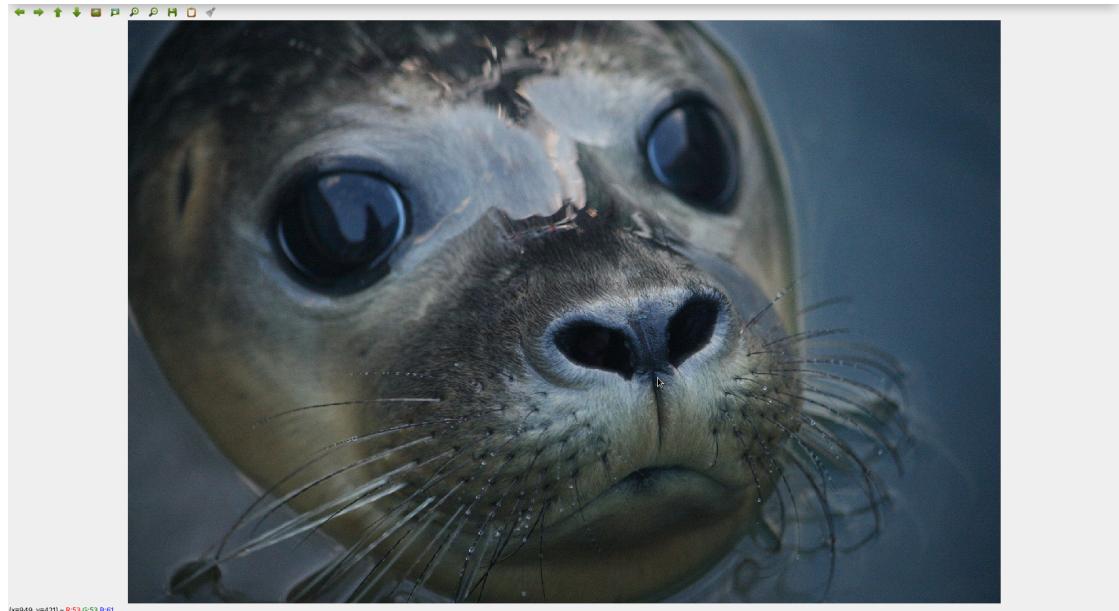
- 1) Выводим содержимое папки до исполнения программы:

```
→ part_1 ls -l
итого 420
-rw-r--r-- 1 ayglazkov users 425980 фев 28 14:05 1.jpg
-rw-r--r-- 1 ayglazkov users    1271 фев 28 14:10 lab1.py
→ part_1
```

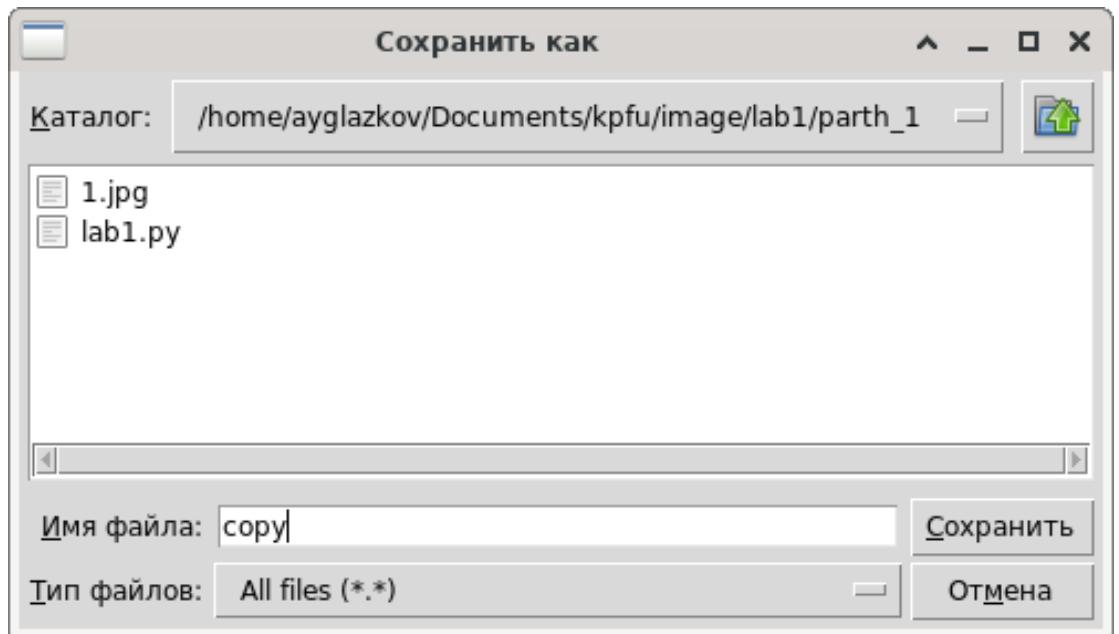
- 2) При помощи графического интерфейса выбираем изображение, которое будем обрабатывать:



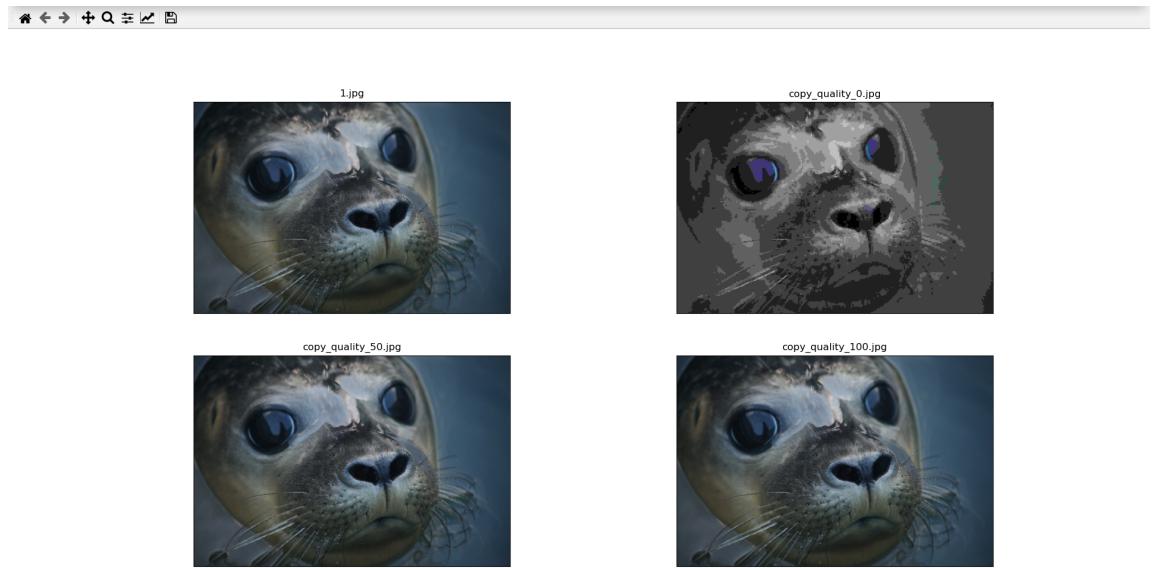
3) Выбрав изображение, выполняется вывод на экран до нажатия любой клавиши:



4) В диалоговом окне выбираем директорию и имя для сохранения изображений с различными степенями сжатия:



5) Для наглядности выводятся полученные изображения рядом друг с другом:



6) После завершения работы оценим содержимое исходной директории и объем полученных файлов:

```
→ parth_1 ls -lh
итого 1,9M
-rw-r--r-- 1 ayglazkov users 416K фев 28 14:05 1.jpg
-rw-r--r-- 1 ayglazkov users 41K мар 8 22:23 copy_quality_0.jpg
-rw-r--r-- 1 ayglazkov users 1,3M мар 8 22:23 copy_quality_100.jpg
-rw-r--r-- 1 ayglazkov users 141K мар 8 22:23 copy_quality_50.jpg
-rw-r--r-- 1 ayglazkov users 1,3K фев 28 14:10 lab1.py
→ parth_1
```

Листинг написанной программы:

```
import cv2
import matplotlib.pyplot as plt
from easygui import fileopenbox, filesavebox

def ft_imgComparison(name_file):
    for i in range(4):
        img = cv2.imread(name_file[i], cv2.IMREAD_ANYCOLOR)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        plt.subplot(2, 2 , i + 1),plt.imshow(img)
        plt.title(name_file[i].rpartition('/')[2])
        plt.xticks([]),plt.yticks([])
    plt.show()

def main():
    flt = ["*.jpg", "*.png", "*.gif", "*.bmp"]
    name_file = []
    path = fileopenbox("image",
                       "select file",
                       default = flt[0],
                       filetypes = flt)
    if path:
        cv2.namedWindow("image", cv2.WINDOW_NORMAL)
        img = cv2.imread(path)
        name_file.append(path)
        cv2.imshow("image", img)
        cv2.waitKey()
        cv2.destroyAllWindows()
        path = filesavebox(default = "copy")
        if path == None:
            path = "img"
        for i in range(0, 101, 50):
            cv2.imwrite(path + "_quality_" + str(i) + ".jpg",
                        img,
                        [int(cv2.IMWRITE_JPEG_QUALITY), i])
            name_file.append(path + "_quality_" + str(i) + ".jpg")
    ft_imgComparison(name_file)

main()
```

Лабораторная работа №2

Функции геометрического преобразования изображений

Цель: Изучение и использование библиотечных функций python выполняющих геометрическое преобразование изображений.

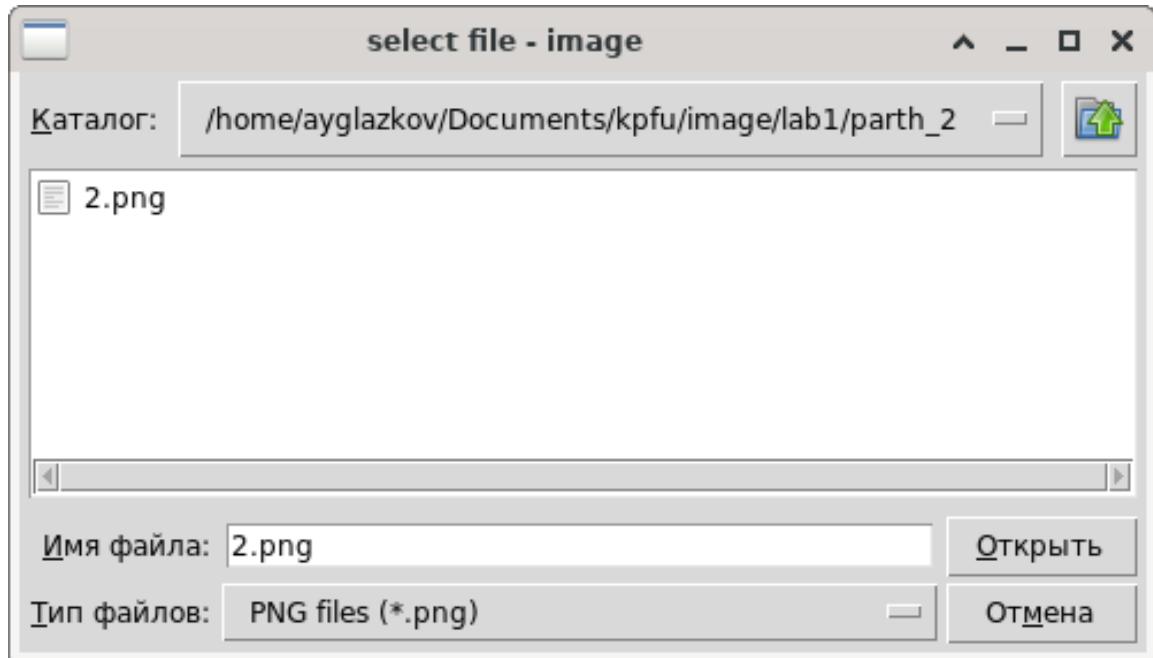
Использованный язык программирования: python.

Ход работы:

- 1) Выводим содержимое папки до исполнения программы:

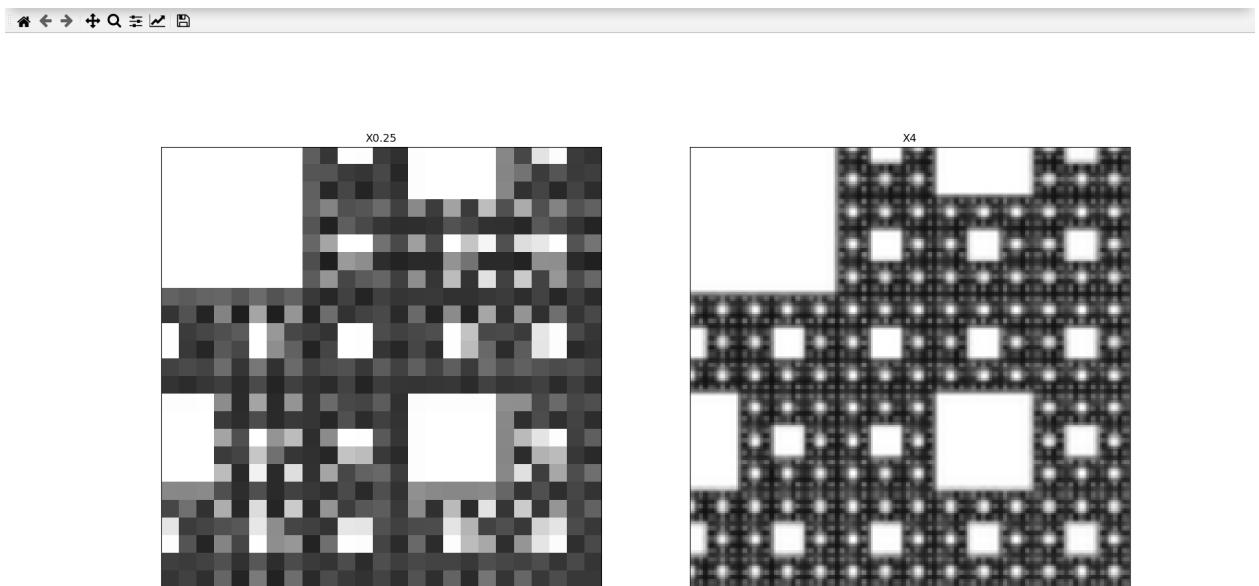
```
→ parth_2 ls -lh
итого 84К
-rw-r--r-- 1 ayglazkov users 78K фев 28 14:13 2.png
-rw-r--r-- 1 ayglazkov users 1,5K фев 28 14:37 lab2.py
→ parth_2
```

- 2) При помощи графического интерфейса выбираем изображение, которое будем обрабатывать:



- 3) После выбора файла открывается окно с кадрированными изображениями 0.25 от исходного изображения (нижняя четверть) и выполненным масштабированием с коэффициентами 4 и 0.25 с билинейным методом

интерполяции:

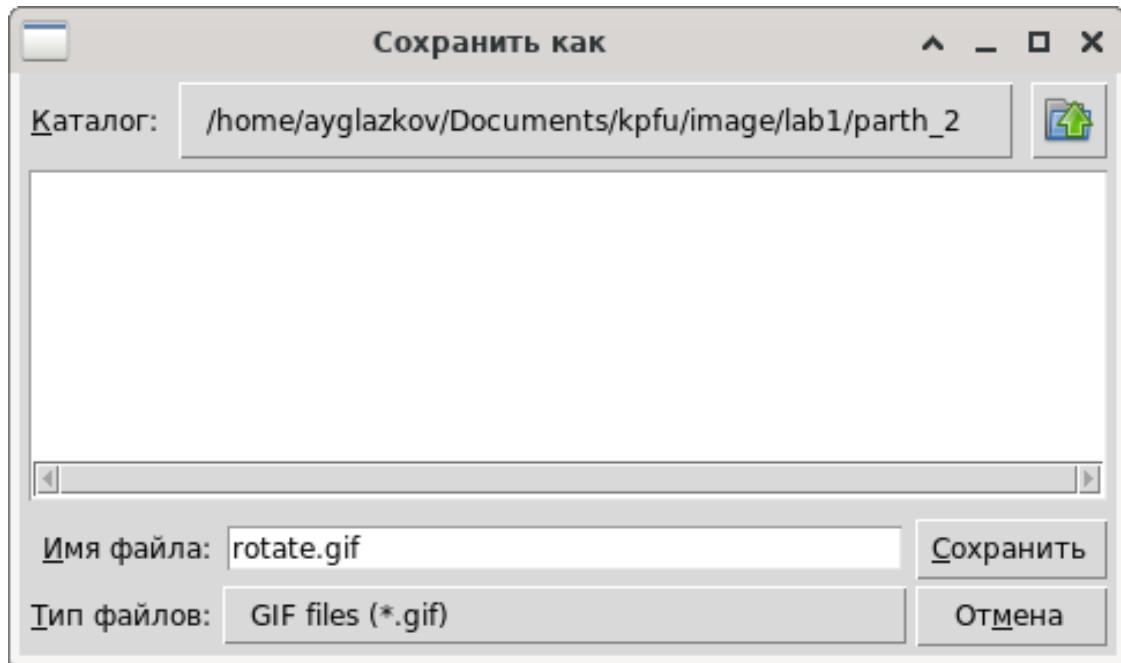


- 4) В это же время в терминале будет выведена информация о размерах матрицы исходного изображения:

```
→ parth_2 ls -lh
итого 84K
-rw-r--r-- 1 ayglazkov users  78K фев 28 14:13 2.png
-rw-r--r-- 1 ayglazkov users 1,5K фев 28 14:37 lab2.py
→ parth_2 python lab2.py
200 x 200
```

- 5) После закрытия предыдущего окна появляется диалоговое окно для выбора директории и названия анимированного изображения (вращение

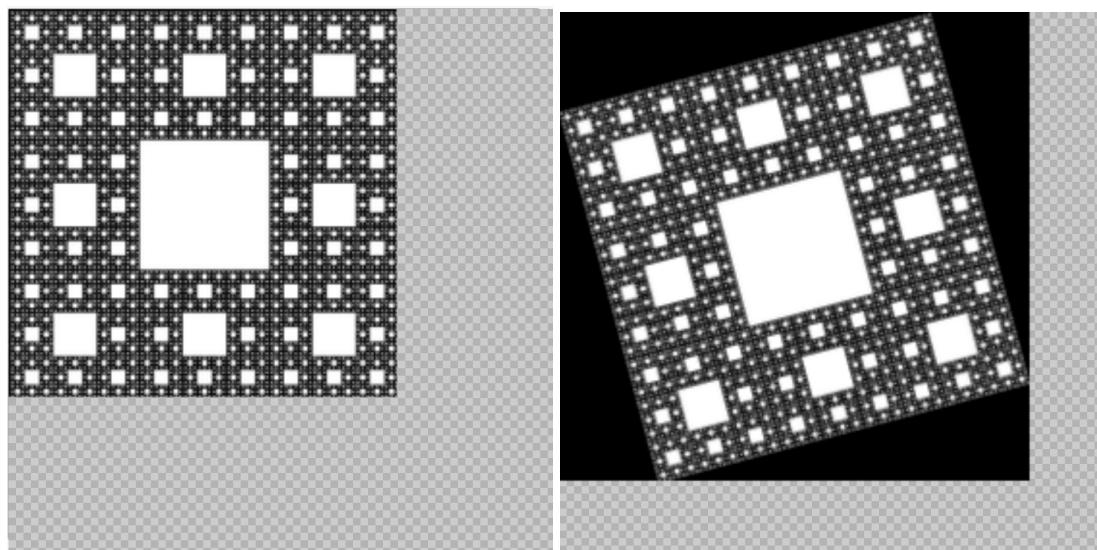
выбранного изображения с шагом 15 градусов):



6) После сохранения оценим содержимое директории:

```
→ part_2 ls -lh
итого 1,1M
-rw-r--r-- 1 ayglazkov users 78K фев 28 14:13 2.png
-rw-r--r-- 1 ayglazkov users 1,5K фев 28 14:37 lab2.py
-rw-r--r-- 1 ayglazkov users 1,1M мар 8 22:47 rotate.gif
→ part_2
```

7) Убедимся в корректности полученной анимации:



Листинг написанной программы:

```
import cv2
import imageio
import matplotlib.pyplot as plt
from imutils import rotate_bound as rb
from easygui import fileopenbox, filesavebox

def ft_size(img):
    (h, w) = img.shape[:2]
    print(h, "x", w)

def ft_crop(img):
    (y, x) = img.shape[:2]
    (y0, x0) = (y, x)
    (y, x) = (y // 2, x // 2)
    res = []
    title = ["X0.25", "X4"]
    crop = img[y: 2 * y, x: 2 * x]
    for i in range(8, 0, -7):
        res.append(cv2.resize(crop,
                              dsize = (x0 // i, y0 // i),
                              interpolation = cv2.INTER_LINEAR))
    for i in range (2):
        plt.subplot(1, 2, i + 1),plt.imshow(res[i])
        plt.title(title[i])
        plt.xticks([]), plt.yticks([])
    plt.show()

def ft_animation(img):
    frames = []
    for i in range(45, 360+45, 15):
        frames.append(rb(img, i))
    path = filesavebox(default = "rotate.gif")
    if path == None:
        path = "Yuuuuuhuu.gif"
    with imageio.get_writer(path, mode = "I") as writer:
        for frame in frames:
            writer.append_data(frame)

def main():
    flt=[ "*.jpg", "* .png"]
    path = fileopenbox("image",
                       "select file",
                       default = flt[0],
                       filetypes = flt)
    if path:
        img = cv2.imread(path, cv2.IMREAD_ANYCOLOR)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        ft_size(img)
        ft_crop(img)
```

```
ft_animation(img)  
main()
```

Лабораторная работа №3

Геометрическое преобразование векторных изображений

Цель: Изучение и программная реализация алгоритмов геометрических преобразований на плоскости.

Использованный язык программирования: python.

Ход работы:

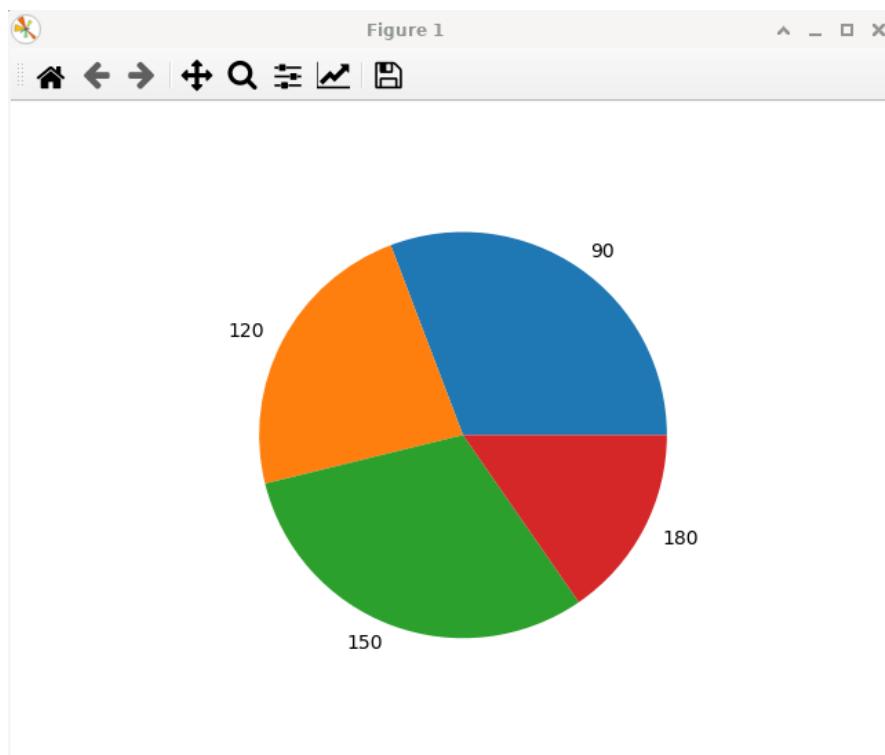
- 1) Выводим содержимое папки до исполнения программы:

```
→ part_3 ls -lh
итого 4,0К
-rw-r--r-- 1 ayglazkov users 2,3К мар  8 21:12 lab3.py
→ part_3
```

- 2) После запуска в терминале будет предложено ввести параметры a и b:

```
→ part_3 python lab3.py
a=5
b=9
```

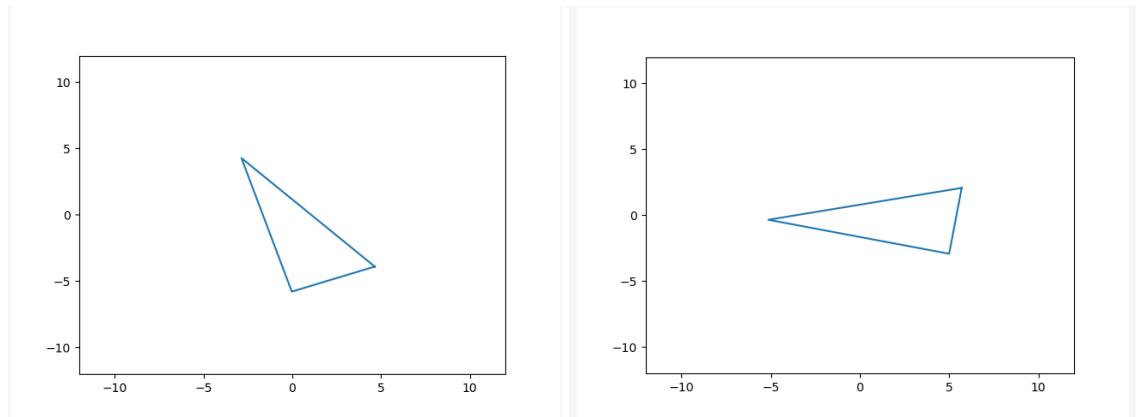
- 3) Затем будет выполнено построение процентного соотношения количества вершин многоугольника от различным углов приращения (90, 120, 150 и 180):



- 4) После нажатия на любую клавишу гистограмма будет закрыта и начнется процесс создания анимированного изображения,

выполняющий поворот над случайным многоугольником. По окончанию которого в папке с проектом появится файл “rotate.gif”:

```
→ part_3 ls -lh
итого 328K
-rw-r--r-- 1 ayglazkov users 2,3K мар 8 21:12 lab3.py
-rw-r--r-- 1 ayglazkov users 322K мар 8 23:05 rotate.gif
→ part_3
```



Листинг написанной программы:

```
import os
import random
import imageio
import numpy as np
import matplotlib.pyplot as plt

def ft_createAnimation(a, b):
    filenames = []
    s_axis=max(a,b)*1.2
    polygon = ft_createPolygon(np.pi, a, b);
    for i in range(0, 360, 10):
        x = []
        y = []
        angle = np.radians(i)
        M = np.float32([ [np.cos(angle), -(np.sin(angle)), 0],
                         [np.sin(angle), np.cos(angle), 0],
                         [0, 0, 1] ])
        ret = polygon.dot(M)
        for j in range(np.size(polygon, 0)):
            x.append(ret[j][0])
            y.append(ret[j][1])
        plt.plot(x,y);
        plt.axis([-s_axis, s_axis, -s_axis, s_axis]);
        filenames.append("tmp_fig" + str(i) + ".png")
        plt.savefig("tmp_fig" + str(i) + ".png")
        plt.close()
        x.clear()
        y.clear()
    with imageio.get_writer('rotate.gif', mode='I') as writer:
        for filename in filenames:
            image = imageio.imread(filename)
            writer.append_data(image)
        for filename in set(filenames):
            os.remove(filename)

def ft_createPolygon(B, a, b):
    n = 0
    phi = []
    phi.append(0)
    while (phi[n] <= np.pi * 2):
        dphi = B * random.random()
        phi.append(phi[n] + dphi)
        n+=1
    mat_1 = np.zeros((1, n))
```

```

for i in range(0, n):
    mat_1[0][i] = a + (b - a) * random.random();
    mat_2 = np.ones((n + 1, 3))
    for i in range(n):
        mat_2[i][0] = mat_1[0][i] * np.cos(phi[i])
        mat_2[i][1] = mat_1[0][i] * np.sin(phi[i])
        mat_2[n] = mat_2[0]
    return (mat_2)

def ft_drawDiagram(a,b):
    vals = []
    pol_1 = ft_createPolygon(np.pi, a, b)
    vals.append(np.size(pol_1, 0) - 1)
    pol_2= ft_createPolygon(np.pi + np.pi / 3, a, b)
    vals.append(np.size(pol_2, 0) - 1)
    pol_3= ft_createPolygon(np.pi + 2 * np.pi / 3, a, b)
    vals.append(np.size(pol_3, 0) - 1)
    pol_4= ft_createPolygon(2 * np.pi, a, b)
    vals.append(np.size(pol_4, 0) - 1)
    labels = ["90", "120", "150", "180"]
    fig, ax = plt.subplots()
    ax.pie(vals, labels=labels)
    plt.draw()
    plt.waitforbuttonpress()
    plt.close(fig)

def main():
    a = int(input("a="))
    b = int(input("b="))
    ft_drawDiagram(a, b)
    ft_createAnimation(a, b)

main()

```

Вывод

В ходе выполненных лабораторных работ ознакомился с основами цифровой обработки изображений на языке python. Изучил основные графические функции и алгоритмы геометрических преобразований на плоскости.