МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Информатика»

ТЕМА: ВВЕДЕНИЕ В АРХИТЕКТУРУ КОМПЬЮТЕРА

Студент гр. 3344 _____ Пачев Д.К.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Написать программу на языке Python, выделив каждую задачу в отдельную функцию, а также освоить на практических задачах модуль Pillow.

Задание

Вариант 2. Вариант Предстоит решить 3 подзадачи, используя библиотеку **Pillow (PIL)**. Для реализации требуемых функций студент должен использовать **numpy** и **PIL**. Аргумент image в функциях подразумевает объект типа <class 'PIL.Image.Image'>

1) Рисование пентаграммы в круге

Необходимо написать функцию pentagram(), которая рисует на изображении пентаграмму в круге.

Функция pentagram() принимает на вход:

- Изображение (img)
- координаты левого верхнего и правого нижнего угла квадрата, в который вписана окружность (x0,y0,x1,y1)
- Толщину линий и окружности (thickness)
- Цвет линий и окружности (color) - представляет собой список (list) из 3-х целых чисел

Функция должна вернуть обработанное изображение.

Примечание:
Вершины пентаграммы высчитывать по формуле:

phi = (pi/5)*(2*i+3/2)
node_i = (int(x0+r*cos(phi)),int(y0+r*sin(phi)))
x0,y0 - координаты центра окружности, в который вписана пентаграмма
r - радиус окружности
i - номер вершины от 0 до 4
Подсказка: Округляйте все вычисляемые вами значения (кроме значений углов) до целых чисел.

2) Инвертирование полос

Необходимо реализовать функцию invert, которая делит изображение на "полосы" и инвертирует цвет нечетных полос (счёт с нуля).

Функция invert() принимает на вход:

- Изображение (img)
- Ширину полос в пикселах (N)
- Признак того, вертикальные или горизонтальные полосы(vertical - если True, то вертикальные)

Функция должна разделить изображение на вертикальные или горизонтальные полосы шириной N пикселей. И инвертировать цвет в нечетных полосах (счет с нуля). **Последняя полоса может быть меньшей ширины, чем N.**

3) Поменять местами 9 частей изображения

Необходимо реализовать функцию mix, которая делит квадратное изображение на 9 равных частей **(сторона изображения делится на 3),** и по правилам, записанным в словаре, меняет их местами.

Функция mix() принимает на вход:

- Изображение (img)
- Словарь с описанием того, какие части на какие менять (rules)

Пример словаря rules:

{0:1,1:2,2:4,3:4,4:5,5:3,6:8,7:8,8:8}

Элементы нумеруются слева-направо, сверху-вниз.

В данном случае нулевой элемент заменяется на первый, первый на второй, второй на четвертый, третий на четвертый и так далее.

Функция должна вернуть обработанное изображение.

Выполнение работы

- Функция `pentagram(img, x0, y0, x1, y1, thickness, color)`:

В переменной `img_draw` создается объект `ImageDraw` и вызывается метод `draw()` для рисования. Радиус окружности `r` высчитывается по формуле `(x1 - x0) // 2`, также определяются координаты центра окружности `circle_x = (x0 + x1) // 2` и `circle_y = (y1 + y0) // 2`. Далее с помощью метода `ellipse()` рисуется окружность, затем с помощью цикла в список `pent_coords` добавляются координаты для построения пентаграммы. С использованием цикла `for` и метода `line ()` рисуется сама пентаграмма. Возвращает функция обработанное изображение.

- Функция `invert(img, N, vertical):`

В переменной `num_of_strip` хранится число полос, на которые необходимо разделить фото. Циклом `for` проходимся по нечетным полосам и с помощью метода `invert` инвертируем их цвет, а методом `paste()` вставляем измененные полосы в изображение. Возвращает функция измененное изображение.

- Функция `mix(img, rules):`

С помощью цикла `for` заполняется данными словарь `paste_coords`, в котором хранятся координаты квадратов, на которые делится изображение, а также в список `img_paste_list` добавляются копии этих квадратов, которые были получены методом `crop()`. Далее опять же циклом `for` и методом `paste()` изменяется исходное изображение в соответствии с указанными правилами `rules`. Возвращает функция обработанное изображение.

Тестирование

Результаты тестирования представлены в Таблице 1

Таблица 1 - Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	pentagram(Image.new('RGB',(300,300),'red'),9,28,129,148,3,[150,178,182])	img	
2.	invert(Image.new('RGB',(300,300),'red'), 100, False)	img	
3.	mix(Image.open('krab1.jpeg'), {0:2,1:2,2:2,3:5,4:5,5:5,6:8,7:8,8:8})	img	

Выводы

В ходе выполнения лабораторной работы была разработана программа на языке Python, в которой каждая подзадача была выделена в отдельную функцию, а также изучены основные методы работы с Pillow.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from PIL import Image, ImageDraw, ImageChops
from math import pi, sin, cos


def pentagram(img, x0, y0, x1, y1, thickness, color):
    r = (x1 - x0) // 2
    img_draw = ImageDraw.Draw(img)
    color = tuple(color)
    circle_x = (x0 + x1) // 2
    circle_y = (y0 + y1) // 2
    img_draw.ellipse(((x0, y0), (x1, y1)), outline=color,
width=thickness)
    pent_coords = []
    for i in range(5):
        phi = (pi / 5) * (2 * i + 3 / 2)
        node_i = (int(circle_x + r * cos(phi)), int(circle_y + r *
sin(phi)))
        pent_coords.append(node_i)
    for i in range(2):
        img_draw.line([pent_coords[i - 2], pent_coords[i],
pent_coords[i + 2]], fill=color, width=thickness)
    img_draw.line([pent_coords[2], pent_coords[4]], fill=color,
width=thickness)
    return img


def invert(img, N, vertical):
    width, height = img.size
    if vertical == True:
        num_of_strip = width // N
        for i in range(1, num_of_strip + 1, 2):
            box = (N * i, 0, N * (i + 1), height)
            invert_img = ImageChops.invert(img.crop(box))
            img.paste(invert_img, box)
    else:
        num_of_strip = height // N
        for i in range(1, num_of_strip + 1, 2):
            box = (0, N * i, width, N * (i + 1))
            invert_img = ImageChops.invert(img.crop(box))
            img.paste(invert_img, box)
    return img


def mix(img, rules):
    width, height = img.size
    rect_width = width // 3
    rect_height = height // 3
    row_count = 1;
    paste_coords = {}
    img_paste_list = []
    for i in range(9):
        box = (rect_width * (i % 3), rect_width * (row_count - 1),
```

```
rect_width * ((i % 3) + 1), rect_height * row_count)
        if i != 0 and (i + 1) % 3 == 0:
            row_count += 1
        paste_coords[i] = box
        img_paste_list.append(img.crop(box))
    for i in range(9):
        opt = rules[i]
        img.paste(img_paste_list[opt], paste_coords[i])
    return img
```