

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. 3342

Романов Е.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Освоение функций модулей Pillow и numpy языка python, получение практических навыков их использования.

Задание

Вариант 2

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.

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`.

Функция должна вернуть обработанное изображение.

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`)

Функция должна вернуть обработанное изображение.

Выполнение работы

Программа состоит из 3 функций, каждая из которых решает определённую задачу варианта.

Первая функция `pentagram()` принимает объект изображения, в котором необходимо разместить пентаграмму в круге, вычисляет центр окружности, используя подаваемые в качестве аргументов координаты левого верхнего и правого нижнего углов квадрата, в который она вписана. После чего вычисляется радиус, используемый в последствии для вычисления координат вершин пентаграммы. В цикле `for` вычисляются координаты вершин пентаграммы, которые после записываются в список для дальнейшего использования. Прорисовка изображений осуществляется методом `Draw` модуля `ImageDraw`: сначала рисуется эллипс, а после, соединяя посчитанные координаты линиями изображается пентаграмма. Полученное изображение возвращается функцией.

Вторая функция `invert` принимает в качестве аргументов изображение, целое число `N`, и значение условного типа данных, определяющее режим работы функции. Сначала устанавливается ширина и высота исходного изображения, затем определяется режим работы функции, после чего изображение в цикле `for` разбивается либо на горизонтальные, либо на вертикальные полосы. Цикл работает с шагом два, находя сразу те части изображения, цвет которых необходимо инвертировать, найденные части вырезаются методом `crop()`. При помощи метода `invert()` цвет полученной части изображения меняется на противоположный, после чего, при помощи метода `paste()` вырезанная часть устанавливается на своё исходное место. Функция возвращает полученное изображение.

Третья функция принимает в качестве аргументов изображение и словарь, содержащий инструкцию по изменению изображения. Внутри функции создаётся два списка: `parts` и `coords`. При помощи цикла `for`, в первый список записываются полученные методом `crop()` части исходного изображения, таким образом, что порядок частей соответствует их нумерации на исходном

изображении. В том же цикле во второй список записываются координаты верхнего левого угла каждой вырезанной части изображения, их порядок в списке также соответствует порядку частей на изображении. Во втором цикле `for` в исходное изображение методом `paste()` вставляются части картинки в порядке, соответствующему правилам, переданным как аргумент функции. Полученное изображение возвращается.

Переменные, используемые в программе:

-`radius` хранит радиус окружности, в которую вписывается пентаграмма.

Локальная переменная функции `pentagram`

-`centerX` хранит координату `X` центра окружности, в которую вписывается пентаграмма. Локальная переменная функции `pentagram`

-`centerY` хранит координату `Y` центра окружности, в которую вписывается пентаграмма. Локальная переменная функции `pentagram`

-`coords` список, хранящий координаты вершин пентаграммы. Локальная переменная функции `pentagram`

-`phi` угол между первой вершиной пентаграммы и текущей, определённой порядком итерации в цикле `for`. Локальная переменная функции `pentagram`

-`node_i` координаты текущей вершины пентаграммы, определённые порядком итерации в цикле `for`. Локальная переменная функции `pentagram`

-`width` хранит ширину исходного изображения. Локальная переменная функции `invert`

-`height` хранит высоту исходного изображения. Локальная переменная функции `invert`

-`block` кортеж координат, определяющих вырезанную часть изображения. Локальная переменная функции `invert`

-part вырезанная по координатам переменной block часть изображения.

Локальная переменная функции invert

-parts список частей, из которых состоит исходное изображение. Локальная переменная функции mix

-coords список координат верхнего левого угла частей изображения.

Локальная переменная функции mix

-w ширина исходного изображения. Локальная переменная функции mix

-h высота исходного изображения. Локальная переменная функции mix

-drawing хранит объект ImageDraw для рисования на изображении

-pi хранит число π

Функции, используемые в программе:

-cos функция модуля numpy для вычисления косинуса угла

-sin функция модуля numpy для вычисления синуса угла

-Draw метод объекта ImageDraw, используемый для создания объекта для рисования внутри подаваемого в качестве аргумента изображения

-ellipse метод объекта ImageDraw, изображающий на исходном изображении эллипс

-line метод объекта ImageDraw, изображающий на исходном изображении линию

-invert метод объекта ImageChops, инвертирующий цвета изображения

-crop метод объекта Image, возвращающий прямоугольную область из исходного изображения

-paste метод объекта Image, вставляет другое изображение в исходное

-append вставляет новый элемент в конец списка

Разработанный код смотри в приложении А

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Функция	Входные данные	Выходные данные
1.	pentagram	img.size = (300,300); x0 = 50;y0 = 50; x1 = 200; y1 = 200;thickness = 5, color = [255,0,0]	корректные
2.	invert	img; img.size = (300,300); N = 70; vertical = True	корректные
3.	mix	img; img.size = (300,300); rules = {0:1,1:2,2:4,3:4,4:5,5:3,6:8,7:8,8:8}	корректные

Выводы

Были изучены основные инструменты для работы с изображениями и рисованием при помощи модуля Pillow. Была разработана программа, состоящая из трёх функций: рисование пентаграммы, инвертирование цветов изображения и смена частей изображения местами.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from numpy import *
from PIL import ImageDraw, ImageChops

def pentagram(img, x0, y0, x1, y1, thickness, color):
    radius = int((x1-x0) / 2)
    centerX = int((x0+x1)/2)
    centerY = int((y0+y1)/2)
    coords=[]
    for i in range(0,10,2):
        phi = (pi/5)*(2*i+3/2)
        node_i =
(int(centerX+radius*cos(phi)),int(centerY+radius*sin(phi)))
        coords.append(node_i)

    drawing = ImageDraw.Draw(img)
    drawing.ellipse(((x0,y0),(x1,y1)), None, tuple(color), thickness)

    for i in range(10*len(coords)):
        drawing.line((coords[i%5], coords[(i+1)%5]), tuple(color),
thickness)
    return img

def invert(img, N,vertical):
    width = img.size[0]
    height =img.size[1]
    if vertical:
        for i in range(1,int(ceil(width / N)),2):
            block = (i*N,0,(i+1)*N,height);
            part = ImageChops.invert(img.crop(block))
            img.paste(part, block)

    else:
        for i in range(1,int(ceil(height / N)),2):
            block = (0,i*N,width,(i+1)*N);
            part = ImageChops.invert(img.crop(block))
            img.paste(part, block)

    return img

def mix(img,rules):
    parts = []
    coords=[]
    w = img.width
    h = img.height
    for y in range(3):
        for x in range(3):
            path = img.crop((x*(w//3), y*(h//3), (x+1)*(w//3),
(y+1)*(h//3)))
            coords.append((x*(w//3), y*(h//3)))
            parts.append(path)
```

```
for i in range(9):  
    img.paste(parts[rules[i]], coords[i])  
  
return img
```