**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 |  | Тукалкин. В.А. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Ознакомиться с основными управляющими конструкциями языка Python и библиотеки Pillow.

## Задание.

Вариант 2.

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.

1. 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.

1. 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).

## Выполнение работы

Выполнение работы будет расписано по шагам:

1. Подключить модуль numpy и сократить до np, PIL, Image ImageDraw, ImageOps из PIL.
2. Написать функцию pentagram, на вход которой поступают: img(изображение), x0(координаты левого верхнего угла по x), y0(координаты левого верхнего угла по y), x1(координаты правого нижнего угла по x), y1(координаты правого нижнего угла по y), thickness(толщина линий), color(список и трёх чисел). Преобразовать color из списка в кортеж, нарисовать круг поверх img, с помощью двух циклов высчитать координаты для 5 точек и записать их в список coordinates, вторым циклом нарисовать линии. Вернуть готовое изображение img.
3. Написать функцию invert, на вход которой поступают: img(изображение), N(количество пикселей в линии), vertical(определитель вертикальные или горизонтальные линии). Написать 2 условия для определения ориентации линий, для вертикальных: написать цикл, идущий от 1 до округления в большую сторону ширины делённой на количество пикселей, с шагом 2. В цикле вырезается полоса и инвертируется цвет, после вставляет в img. Для горизонтальный линий аналогично, только вместо x и осей x используется y и ось y.
4. Написать функцию mix, на вход которой поступают: img(изображение), rules(словарь с перестановками). Перенести значения из словаря в список, вложеными циклами разрезать изображение img на 9 равных частей и добавить в список pieces. С помощью цикла вставить все кусочки согласно правилам из словаря rules, для этого берём число из списка a, которое будет индексом для списка pieces.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | pentagram(img,143,87,182,126,2,[18,59,70]) |  | Верный ответ |
|  | invert(img,30,True) |  | Верный ответ |
|  | mix(img,{0:1,1:2,2:4,3:4,4:5,5:3,6:8,7:8,8:8}) [ 1 -3 -1] [-2 7 2] [ 3 2 -4]  check\_rotation input:  [ 5 -6 1] 0.87 |  | Верный ответ |

## Выводы

Были изучены основные управляющие конструкции языка Python и библиотеки Pillow на примере использующей их программы.

Разработана программа, выполняющая обработку изображения, при помощи библиотеки Pillow.

# Приложение А Исходный код программы

Название файла: lb2.py

import math

import numpy as np

from PIL import Image, ImageDraw, ImageOps

def pentagram(img, x0, y0, x1, y1, thickness, color):

color=(color[0],color[1],color[2])

ImageDraw.Draw(img).ellipse([(x0,y0),(x1,y1)],None,color,thickness)

r=abs(x0-x1)/2

coordinates=[]

for i in range(5):

phi=(np.pi/5)\*(2\*i+3/2)

coordinates.append((int(x0+r+r\*np.cos(phi)),int(y0+r+r\*np.sin(phi))))

for i in range(3):

ImageDraw.Draw(img).line((coordinates[i-2],coordinates[i],coordinates[i+2]),color,thickness)

return img

def invert(img,N,vertical):

x,y=img.size

if vertical==True:

for i in range(1,math.ceil(x/N)+1,2):

img1=img.crop((i\*N,0,(i + 1)\*N,y))

img1=ImageOps.invert(img1)

img.paste(img1,(i\*N,0))

if vertical==False:

for i in range(1,math.ceil(y/N)+1,2):

img1=img.crop((0,i\*N,x,(i+1)\*N))

img1=ImageOps.invert(img1)

img.paste(img1,(0,i\*N))

return img

def mix(img, rules):

a=[]

for i in range(len(rules)):

a.append(rules[i])

pieces=[]

x,y=img.size

for j in range(3):

for i in range(3):

img2=img.crop((i\*x//3,j\*y//3,(i+1)\*x//3,(j+1)\*y//3))

pieces.append(img2)

for i in range(9):

img.paste(pieces[a[i]],((x//3)\*(i%3),(y//3)\*(i//3)))

return img