

**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

Хайруллов Д.Л.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Целью работы является изучение работы с библиотекой Pillow в языке программирования python.

## Задание

### Вариант 1

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 треугольника

Необходимо написать функцию `triangle()`, которая рисует на изображении треугольник

Функция `triangle()` принимает на вход:

Изображение (`img`)

Координаты вершин (`x0,y0,x1,y1,x2,y2`)

Толщину линий (`thickness`)

Цвет линий (`color`) - представляет собой список (`list`) из 3-х целых чисел

Цвет, которым залит (`fill_color` - если значение `None`, значит треугольник не залит) - представляет собой список (`list`) из 3-х целых чисел

Функция должна вернуть исходное обработанное изображение.

#### 2) Замена наиболее часто встречаемого цвета.

Необходимо написать функцию `change_color()`, которая заменяет наиболее часто встречаемый цвет на переданный.

Функция `change_color()` принимает на вход:

Изображение (`img`)

Цвет (`color` - представляет собой список из трех целых чисел)

Функция должна найти в изображении самый частый цвет и заменить его на переданный, затем вернуть новое изображение (исходное изображение не должно меняться).

### 3)Коллаж

Необходимо написать функцию collage().

Функция collage() принимает на вход:

Изображение (img)

Количество изображений по "оси" Y (N - натуральное)

Количество изображений по "оси" X (M - натуральное)

Функция должна создать коллаж изображений (это же изображение, повторяющееся NxM раз. (N раз по высоте, M раз по ширине) и вернуть его (новое изображение).

## **Выполнение работы**

В функции `triangle()` по заданным параметрам с помощью метода `polygon()` рисует треугольник на данном изображении. В конце работы функция возвращает измененное изображение.

В функции `change_color()` создается копия заданного изображения. Затем с помощью методов `getpixel()` и `putpixel()` заменяются цвета пикселей того цвета, который чаще встречается в изображении. В конце работы функция возвращает измененную копию изображения.

В функции `collage()` создается новое изображение, на которое с помощью метода `paste()` помещается заданное изображение. Тем самым создается коллаж, возвращаемый в конце работы функции.

Разработанный программный код см. в приложении А.

## **Выводы**

Было изучено использование библиотеки Pillow в языке программирования python.

Для решения каждой задачи была написана отдельная функция.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from PIL import Image, ImageDraw
import numpy

# Задача 1
def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color,
fill_color):
    drawing = ImageDraw.Draw(img)
    if fill_color != None:
        drawing.polygon(((x0,y0), (x1,y1), (x2,y2)),
tuple(fill_color), tuple(color), thickness)
    else:
        drawing.polygon(((x0,y0), (x1,y1), (x2,y2)), None,
tuple(color), thickness)
    return img

# Задача 2
def change_color(img, color):
    col_dict = {}
    res_col = []
    col_counter = 0
    res = img.copy()

    for width in range (0,res.size[0]):
        for length in range (0,res.size[1]):
            if res.getpixel((width, length)) not in col_dict:
                col_dict[res.getpixel((width, length))] = 1
            else:
                col_dict[res.getpixel((width, length))] +=1

    for col in col_dict.keys():
        if col_dict[col] > col_counter:
            col_counter = col_dict[col]
            res_col.append(col)

    col = res_col[::-1][0]

    for width in range (0,res.size[0]):
        for length in range (0,res.size[1]):
            if res.getpixel((width, length)) == col:
                res.putpixel((width, length), tuple(color))
    return res

# Задача 3
def collage(img, N, M):
    res = Image.new("RGB", (img.size[0]*M, img.size[1]*N), 'white')
    for x in range(0, img.size[0]*(M-1)+1, img.size[0]):
        for y in range(0, img.size[1]*(N-1)+1, img.size[1]):
            res.paste(img, (x,y))
```

```
return res
```