

**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

Колесниченко М.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Освоение работы с функциями и с библиотекой Pillow в языке python.

## Задание

Вариант 1.

Задача 1.

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.

Задача 2.

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).

Задача 3.

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).

При необходимости можно писать дополнительные функции.

## **Выполнение работы**

Данная программа написана на языке Python с использованием библиотеки Pillow. Она состоит из 3-функций.

Первая функция `triangle` по заданным параметрам рисует треугольник на изображении и возвращает его.

Вторая функция `change_color`. Функция заменяет наиболее часто встречающийся цвет на заданный цвет.

Третья функция `collage` заменяет исходное изображение на изображение-коллаж. Она создает коллаж изображений на основе исходного изображения `img`, повторяя его N раз по вертикали и M раз по горизонтали.

Данная программа демонстрирует использование функций библиотеки Pillow и работу функций на языке Python.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
	img.size=(90, 51) ; x0 = 34; y0 = 49; x1 = 47; y1 = 38; x2 = 39; y2 = 5; thickness = 5; color = [28, 126, 47]; fill_color = [96, 101, 254]	корректные	Ответ корректный
	Img.size(100, 100) ; [28, 126, 47]	корректные	Ответ корректный
	Img.size(120, 120) ; 10 ; 12	корректные	Ответ корректный

## **Выводы**

Были изучены правила работы с функциями в языке python и работа с библиотекой Pillow.

Разработаны функции, возвращающие решения определенных задач.

## 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):
    draw = ImageDraw.Draw(img)
    coords = [(x0,y0),(x1,y1),(x2,y2)]
    if fill_color != None:
        draw.polygon(coords, tuple(fill_color), tuple(color),
thickness)
    else:
        draw.polygon(coords, None, tuple(color), thickness)
    return img

# Задача 2
def change_color(img, color):
    width, height = img.size
    counters = {}
    for x in range(0,width):
        for y in range(0, height):
            curr_color = img.getpixel((x,y))
            try:
                counters[curr_color] += 1
            except KeyError:
                counters[curr_color] = 1
    max_key = max(counters, key=counters.get)
    for x in range(0,width):
        for y in range(0, height):
            curr_color = img.getpixel((x,y))
            if curr_color == max_key:
                img.putpixel((x,y), tuple(color))
    return img

# Задача 3
def collage(img, N, M):
    width, height = img.size
    new_img = Image.new("RGB", (width*M, height*N), "white")
    for x in range(M):
        for y in range(N):
            new_img.paste(img, ((width*x), (height*y)))
    return new_img
```