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 _____ Лучкин М.А.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург
2023

Цель работы

Целью работы является освоение работы с функциями в языке python и с библиотеками numpy и Pillow.

Задание

Вариант 2.

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 библиотек numpy и Pillow. Она состоит из 3-функций, которые вызываются сразу на сайте https://e.moevm.info.

Первая функция triangle возвращает исходное, обработанное изображение. Функция по заданным параметрам рисует треугольник на изображении и возвращает его.

Вторая функция change_color. Функция заменяет исходное изображение на изображение, в котором наиболее часто встречающийся цвет заменяется 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.

Функции, используемые в этой программе:
-polygon рисует фигуру по заданным параметрам.
-numpy.unique возвращает отсортированные уникальные элементы массива.
-image.new создает новое изображение
-image.size возвращает размер изображения

Данная программа демонстрирует использование функций библиотек numpy и 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 numpy.

Разработаны функции, возвращающие решения определенных математических заданий.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
import numpy as np
from PIL import Image, ImageDraw
# З а д а ч а 1
def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill_color=None):
    draw = ImageDraw.Draw(img)
    points = [(x0, y0), (x1, y1), (x2, y2)]
    draw.polygon(points,outline=tuple(color), width=thickness,fill=tuple(fill_color) if fill_color else None)

    return img

# З а д а ч а 2
def change_color(img, color):
    img_array = np.array(img)
    unique_colors, counts = np.unique(img_array.reshape(-1, 3), return_counts=True, axis=0)
    most_frequent_color = unique_colors[np.argmax(counts)]
    img_array[np.all(img_array == most_frequent_color, axis=-1)] = color
    new_img = Image.fromarray(img_array)

    return new_img

# З а д а ч а 3
def collage(img, N, M):
    width, height = img.size
    collage_img = Image.new('RGB', (M * width, N * height))

    for y in range(N):
        for x in range(M):
            collage_img.paste(img, (x * width, y * height))

    return collage_img
```