**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Информатика»**

Тема: Введение в архитектуру компьютера

|  |  |  |
| --- | --- | --- |
| Студент гр. 3341 |  | Мокров И.О. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Целью лабораторной работы является изучение модуля PIL языка python и его практическое применение для решения трех подзадач лабораторной работы.

## 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).

При необходимости можно писать дополнительные функции.

## Выполнение работы

Импортируем Image, ImageDraw, ImageOps из модуля PIL, импортируем модель numpy.

Задача 1.

Создадим переменную draw с помощью которой будем добавлять треугольник на картинку, после смотрим, если переменная fill\_color имеет значение None, то рисуем прозрачный треугольник при помощи метода polygon по заданным координатам с заданными значениями обводки (ее цветом и шириной), иначе создаем такой же треугольник, но с уже с заливкой определенного цвета. В конце функции возвращаем полученное изображение.

Задача 2.

Создаем переменную colors в которой будет храниться кортеж значений количество цветов – сам цвет. При помощи цикла for находим максимальное количество раз, когда цвет встречался в изображении и присваиваем переменной color\_of\_max этот значение этого цвета, далее при помощи метода load проходимся по каждому пикселю и если цвет текущего пикселя совпал с цветом color\_of\_max, меняем его цвет на тот, который давался в аргументе функции (на color) после завершения циклов возвращаем получившееся изображение.

Задача 3.

В переменные x y записываем ширину и высоту изображения, которое подавалось в аргументе функции, далее создаем новое изображение back ширина которого равна x\*M, а высота y\*N. Создаем вложенный цикл, в котором изображение img заполняет всю строку back M раз при помощи метода past, когда img заполнит всю строку первый цикл сместит back\_y на y пикселей, тем самым перейдя на новую строку и так N раз. В конце возвращаем полученное изображение back.

## Тестирование

Результаты тестирования представлены в Таблице 1.

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
| 1 | triangle(img, 125, 125, 0, 150, 0, 0, 1, (125, 125, 125), (0, 125, 0)) |  | Задача 1 |
| 2 | change\_color(Image.new("RGB", (350, 350), (255, 0, 0)), (0, 255,0)) |  | Задача 2 |
| 3 | collage(img, 3, 3) |  | Задача 3 |

## Выводы

В результате выполнения лабораторной работы были изучены и использованный на практике функции библиотеки PIL языка python, написанна программа, выполняющая три подзадачи.

# Приложение А Исходный код программы

Исходный файл: main.py

import PIL

from PIL import Image, ImageDraw

def triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill\_color):

draw = ImageDraw.Draw(img)

if fill\_color == None:

draw.polygon([(x0,y0), (x1, y1), (x2,y2)], None, (color[0], color[1], color[2]), thickness)

return img

else:

draw.polygon([(x0,y0), (x1, y1), (x2,y2)], (fill\_color[0], fill\_color[1], fill\_color[2]), (color[0], color[1], color[2]), thickness)

return img

# Задача 2

def change\_color(img, color):

colors = img.getcolors(100000)

maxcolor = 0

color\_of\_max = (0,0,0)

for i in colors:

if maxcolor < i[0]:

maxcolor = i[0]

color\_of\_max = i[1]

pixdata = img.load()

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y] == color\_of\_max:

pixdata[x, y] = (color[0], color[1], color[2])

return img

# Задача 3

def collage(img, N, M):

x, y = img.size[0], img.size[1]

back = Image.new("RGB", (x\*M, y\*N), (0, 0, 0))

for back\_y in range(0, back.size[1], y):

for back\_x in range(0, back.size[0], x):

back.paste(img, (back\_x, back\_y))

return back