**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3342 |  | Епонишникова А.И |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Целью работы является на практике изучить основные функции библиотеки Pillow, а также научиться с ней работать.

## Задание

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.

1. 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).

1. 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

Для начала подключим библиотеку Pillow.

1. Функция triangle(img, x0, y0, x1, y1, x2, y2, thickness, color, fill\_color).  
   Нам надо нарисовать треугольник, используя дающиеся нам координаты. Для того, чтобы появилась возможность рисовать фигуры на изображении надо перейти к специальному объекту, вызвав ImageDraw.Draw(img). Далее мы проверяем fill\_color == None или нет, а затем используем метод ImageDraw.Draw.polygon. Функция возвращает измененное изображение.
2. Функция change\_color(img, color). Мы записываем в img\_new img. Далее с помощью метода getdata(), который возвращает содержимое этого изображения в виде объекта последовательности, которое содержит значения пикселей. Создаем библиотеку colors, в которой будут записаны цвета, а также как часто они встречаются в изображении. Далее проходимся по всем пикселям, а результаты записываем в библиотеку colors. Находим цвет, который чаще всего встречается в изображении. Затем проходимся по ширине и длине изображения, с помощью getpixel() находим пиксели, которые соответствуют цвету, который чаще всего встречается, а затем заменяем его на заданный цвет, используя putpixel(). Функция возвращает новое изображение(img\_new).
3. Функция collage(img, N, M). Записываем ширину и высоту исходного изображения. Далее создаем новое изображение(img\_new), его ширина и высота равна предыдущем, только умноженным соответственно на M и N, а также цвет изображения делаем белый. Записываем новую ширину и высоту. Далее проходим по новой ширине от нуля с шагом исходной ширины. То же самое с высотой. Используем специальный метод paste, чтобы отрисовать исходное изображение на img\_new. Функция возвращает новое изображение.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
| triangle | img = (‘RGB’, (300,300), ‘white’); x0 = 100, y0 = 100, x1 = 200, y1 = 100, x2 = 150, y2 = 200; thickness = 5; color = [100, 230, 140]; fill\_color = [120, 45, 98] | Корректные |
| change\_color | img = (‘RGB’, (300,300), ‘green’); color = [155, 28, 149] | Корректные |
| collage | img = (‘RGB’, (300,300), ‘red’); M = 5; N = 4 | Корректные |

## Выводы

На практике научились работать с основными функциями встроенной библиотеки Pillow, также была разработана программа, состоящая из трех функций.

# Приложение А Исходный код программы

Название файла: lab\_2\_cs.py

import PIL

import numpy as np

import math

from PIL import Image, ImageDraw

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

drawing= ImageDraw.Draw(img)

if fill\_color:

drawing.polygon(((x0,y0),(x1,y1),(x2,y2)), fill=tuple(fill\_color), outline=tuple(color), width = thickness)

else:

drawing.polygon(((x0,y0),(x1,y1),(x2,y2)), fill=None, outline=tuple(color), width = thickness)

return img

def change\_color(img, color):

img\_new = img

pixels = img\_new.getdata()

colors = dict()

for x in pixels:

if x in colors:

colors[x] +=1

else:

colors[x] = 1

max\_color = max(colors, key=colors.get)

width,height = img\_new.size

for x in range(width):

for y in range(height):

if img\_new.getpixel((x,y)) == max\_color:

img\_new.putpixel((x,y), tuple(color))

return img\_new

def collage(img, N, M):

width, height = img.size

img\_new = Image.new('RGB', (width\*M, height\*N), 'white')

new\_width, new\_height = img\_new.size

for x in range(0, new\_width, width):

for y in range(0, new\_height, height):

img\_new.paste(img, (x,y))

return img\_new