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. 3344 _____ Ханнанов А.Ф.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы.

Научиться работать с библиотеками в языке программирования Python.

Задание.

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 отрезка. Отрезок определяется:

- координатами начала
- координатами конца
- цветом
- толщиной.

Необходимо реализовать функцию user_func(), рисующую на картинке отрезок

Функция user_func() принимает на вход:

- изображение;
- координаты начала (x0, y0);
- координаты конца (x1, y1);
- цвет;
- толщину.

Функция должна вернуть обработанное изображение.

2) Преобразовать в Ч/Б изображение (любым простым способом).

Функционал определяется:

- Координатами левого верхнего угла области;
- Координатами правого нижнего угла области;
- Алгоритмом, если реализовано несколько алгоритмов преобразования изображения (по желанию студента).

Нужно реализовать 2 функции:

- check_coords(image, x0, y0, x1, y1) - проверяет координаты области (x0, y0, x1, y1) на корректность (они должны быть неотрицательными, не превышать размеров изображения, поскольку x0, y0 - координаты левого верхнего угла, x1, y1 - координаты правого нижнего угла, то x1 должен быть больше x0, а y1 должен быть больше y0);
- set_black_white(image, x0, y0, x1, y1) - преобразовывает заданную область изображения в черно-белый (используйте для конвертации параметр '1'). В этой функции должна вызываться функция проверки, и, если область некорректна, то должно быть возвращено исходное изображение без изменений. Примечание: поскольку черно-белый формат изображения (greyscale) является самостоятельным форматом, а не вариацией RGB-формата, для его получения необходимо использовать метод Image.convert.

3) Найти самый большой прямоугольник заданного цвета и перекрасить его в другой цвет. Функционал определяется:

- Цветом, прямоугольник которого надо найти
- Цветом, в который надо его перекрасить.

Написать функцию find_rect_and_recolor(image, old_color, new_color), принимающую на вход изображение и кортежи rgb-компонент старого и нового цветов. Она выполняет задачу и возвращает изображение. При необходимости можно писать дополнительные функции.

Выполнение работы.

Задача 1: Создана функция user_func, в которой с помощью ImageDraw.Draw.line() рисуется линия.

Задача 2: Создана функция check_coords, она возвращает булевое значение — входит ли область в рисунок. Создана функция set_black_while, в ней вызывается функция check_coords, для проверки области, при возвращении

истины она изменяет заданную область при помощи Image.crop().convert() и Image.paste().

Задача 3: Создана функция find_rect_and_recolor, вначале ищется максимальный прямоугольник. Поиск происходит через два цикла, которыми проходятся все точки, переменные циклов определяют левую верхнюю точку. После этого, если точка соответсвует цвету, внутри цикла запускаются по очереди два цикла while, с помощью них происходит поиск правой нижней точки. После этого через координаты вычисляется площадь прямоугольника, и если его площадь превосходит ранее найденную максимальную, то точки записываются в rect_pos. После выполнения циклов сохраняются точки максимального прямоугольника и с помощью ImageDraw.Draw.rectangle() перекрашивается этот прямоугольник.

Функции:

- user_func(image, x0, y0, x1, y1, fill, width) — рисует прямой отрезок с координатами концов (x0, y0) и (x1, y1), цвета fill и ширины width
- check_coords(image, x0, y0, x1, y1) — проверяет, корректны ли координаты области
- set_black_white(image, x0, y0, x1, y1) — преобразует заданную область в чёрно-белое изображение
- find_rect_and_recolor(image, old_color, new_color) — ищет наибольший прямоугольник цвета old_color и перекрашивает его в цвет new_color

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

1.	user_func(image, 10, 10, 100, 150, „red“, 20)	image	Корректное выполнение
2.	set_black_white(image, 15, 10, 300, 300)	image	Корректное выполнение
3.	find_rect_and_recolor(image, „red“, „black“)	image	Корректное выполнение

Выводы.

В ходе выполнения лабораторной работы, были получены навыки работы с языком Pyhton, получены навыки работы с библиотекой Pillow.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from PIL import Image, ImageDraw


# З а д а ч а 1
def user_func(image, x0, y0, x1, y1, fill, width):
    drawing = ImageDraw.Draw(image)
    drawing.line((x0, y0, x1, y1), fill, width)

    return image


# З а д а ч а 2
def check_coords(image, x0, y0, x1, y1):
    x, y = image.size
    if (0 <= x0 < x1 <= x) and (0 <= y0 < y1 <= y):
        return True
    return False


def set_black_white(image, x0, y0, x1, y1):
    if check_coords(image, x0, y0, x1, y1):
        image_zone = image.crop((x0, y0, x1, y1)).convert('1')
        image.paste(image_zone, (x0, y0))

    return image


# З а д а ч а 3
def find_rect_and_recolor(image, old_color, new_color):
```

```
color_arr = image.load()
x, y = image.size
rect_pos = (0, 0, 0, 0)
max_square = 0

for x0 in range(x):
    for y0 in range(y):
        if color_arr[x0, y0] == old_color:
            x1, y1 = x0, y0
            while x1 < x and color_arr[x1, y0] == old_color:
                x1 += 1
            while y1 < y and color_arr[x0, y1] == old_color:
                y1 += 1
            square = (x1 - x0) * (y1 - y0)
            if square > max_square:
                max_square = square
                rect_pos = (x0, y0, x1 - 1, y1 - 1)

drawing = ImageDraw.Draw(image)
drawing.rectangle(rect_pos, new_color)

return image
```