

**Федеральное государственное бюджетное образовательное
учреждение высшего образования «Московский
государственный университет имени М. В. Ломоносова»**

Кафедра вычислительной механики

**ОТЧЁТ ПО ЗАДАЧЕ НА РАБОТУ С
ИЗОБРАЖЕНИЯМИ ПО ТЕМЕ:
ИЗМЕНЕНИЕ ЯРКОСТИ ВМР-ИЗОБРАЖЕНИЯ**

**Преподаватель: Почеревин Роман Владимирович
Студент 223 группы: Скворцов Андрей Сергеевич**

Москва
2024

Отчёт по работе с BMP-изображениями в Python-3

Задание. Реализовать алгоритмы фрактального сжатия и восстановления изображения.

Решение. Используя библиотеки `numpy` и `scipy`, можно решить эту задачу проще:

```
from scipy import ndimage import numpy as np
```

Пусть изображения заданы внутри программы. Выгрузим, посчитаем длину и ширину:

```
imin = Image.open("input.bmp")  
x1, y1 = imin.size  
px1 = imin.load()
```

Далее создадим выходное изображение и возможность изменять в нем пиксели, а также введем необходимые переменные:

```
imout = Image.new("RGB (x1, y1), (0, 0, 0)) draw = ImageDraw.Draw(imout)
```

$$size_{square} = 15 fractal = 0 \quad (1)$$

Теперь - логика программы. Найдем среднюю яркость квадратного блока со стороной $size_{square}$ и запишем ее в переменную $factor$. Затем найдем среднюю яркость 3 соседних блоков того же размера и сохраним их значения в список $factora$.

Теперь посмотрим какой из этих 3 блок больше похож на первый сравнивая их по средней яркости и заменяя все 4 блока на самый похожий:

Теперь нарисуем оставшиеся блоки также как в исходном изображении:

Сохраним изображение и очистим память от элемента $draw$:

```
imout.save("out.bmp "BMP")  
del draw
```