

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)  
Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра 806: «Вычислительная математика и программирование»

# ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №1

По курсу: «Цифровая обработка изображений»  
Тема: «Основы работы с цифровыми изображениями»

*Студент:* Чернышев Д.В.

*Группа:* М8О-107М-22

*Вариант:* 8

*Преподаватель:* Гаврилов К.Ю.

Москва

2023

## СОДЕРЖАНИЕ

1	Задание к лабораторной работе №1.....	3
2	Выполнение лабораторной работы №1 .....	4

# **1 Задание к лабораторной работе №1**

## **Часть 1**

Создайте полутоновое изображение размером 1000 x 1000 пикселей. На изображении создайте вертикальные полосы с увеличивающейся шириной: каждая следующая полоса шире предыдущей в 2 раза. Ширина первой полосы 5 пикселей.

Яркость каждой полосы должна увеличиваться по линейному закону от самой темной (черный цвет) до самой светлой (белый цвет). Сохраните изображение в виде файла формата jpg.

## **Часть 2**

Создайте цветную картину размером 500 x 500 пикселей с изображением интерференции двух кольцевых волн красного цвета длиной 20 пикселей, исходящих из левого верхнего и левого нижнего углов. Сохраните изображение в виде файла формата jpg.

## 2 Выполнение лабораторной работы №1

### Часть 1

```
1 # Загрузка необходимых библиотек
2 import math
3 from PIL import Image, ImageDraw, ImageStat
4 import halftone as ht
5 import random
6 import numpy as np
7 import os
8 import imageio
9
10 w,h = 1000,1000
11 mu, sigma = 255/2, 255
12
13
14 pixels = np.random.uniform(255,255,size=(w,h,3))
15 image = Image.fromarray(pixels.astype('uint8'), 'RGB')
16 image_file = image.convert('1') # convert image
17 image_file.save('your_file.png')
18 image.save("your_file_RGB.png")
19
20 im = Image.open("your_file.png")
21
22 data = np.array(im.getdata())
23 pixels_from_image = data.reshape(*im.size, -1)
24
25 # функция линейного изменения яркости от 255 до 0
26 def linear_brightness(w,start_i):
27     b0=0
28     b1 = w
29     x0 = 255
```

```

30     x1 = 0
31
32     return (b1- b0)/(x1-x0) * start_i
33 # запуск основного цикла
34 width_i = 5 # начальная ширина
35 start_x = 0 # начальной позиция прохождения по ширине
36 while True:
37     brightness = linear_brightness(w,start_x)
38     pixels_from_image[:,start_x : start_x + width_i ] = brightness
39     print(f"w = {w}; start_x = {start_x}")
40     start_x += width_i
41     width_i *=2
42     if start_x > w:
43         break
44 # конвертация матрицы и сохранение изображения
45 pixels_from_image=pixels_from_image.reshape((w, h))
46 pixels_from_image = pixels_from_image.astype('uint8')*255
47 image = Image.fromarray(pixels_from_image.astype('uint8'))
48 image.save("your_file_fixed.png")
49

```



Рис. 1: Результат выполнения части 1 лабораторной работы №1.

## Часть 2

```
1 # частоты волн видимого спектра
2 frequencies = {
3     "red":430*1e4,
4     "orange":492*1e4,
5     "yellow":520*1e4,
6     "green":575*1e4,
7     "cyan":610*1e4,
8     "blue":645*1e4,
9     "violet":730*1e4
10 }
11 # волновая функция
12 def phases_waves( x,y,t,
13                 x0:float=0,y0:float=0,
14                 frequency_wave:str="red",
```

```

15         Amplitude:float=20,
16         phi_0=0):
17     C = 299792458.0 # "Speed of light in vacuum
18     freq = frequencies[frequency_wave]
19
20     lambda_wave = C/freq
21     k = 2 * np.pi / lambda_wave
22     # при раскоментировании оставшейся части код
23     return Amplitude * np.cos(
24         (((x-x0)**2 + (y-y0)**2)*k)**0.5
25         # - 2 * np.pi * freq * t + phi_0
26     )
27 # создание папок для сохранения
28 import os
29 WAVES_IMAGES = "./WAVES_jpgs"
30 if not os.path.exists(WAVES_IMAGES):
31     os.makedirs(WAVES_IMAGES)
32
33 w,h = 500,500
34
35 xc0,yc0 = 0,0
36 xc1,yc1 = w,0
37
38
39 wave = 'red'
40 Amplitude = 50
41 t_i = 0
42 X = np.tile(np.linspace(0,w,w), (w, 1))
43 Y = np.tile(np.linspace(0,h,h).reshape((-1,1)), (1, h))
44
45 pixels = np.full((w,h,3), 255)
46 pixels[:, :, 0] = pixels[:, :, 1] = pixels[:, :, 2] = phases_waves(
47     t=t_i,

```

```
48     x=X,
49     y=Y,
50     x0=xc0,
51     y0=yc0,
52     frequency_wave=wave,
53     Amplitude=Amplitude
54 )
55 + phases_waves(
56     t=t_i,
57     x=X,
58     y=Y,
59     x0=xc1,
60     y0=yc1,
61     frequency_wave=wave,
62     Amplitude=Amplitude
63 )
64
65 image = Image.fromarray(pixels.astype('uint8'), 'RGB')
66 images.append(image)
67 image.save(f"{WAVES_IMAGES}/{t_i}.jpeg")
```



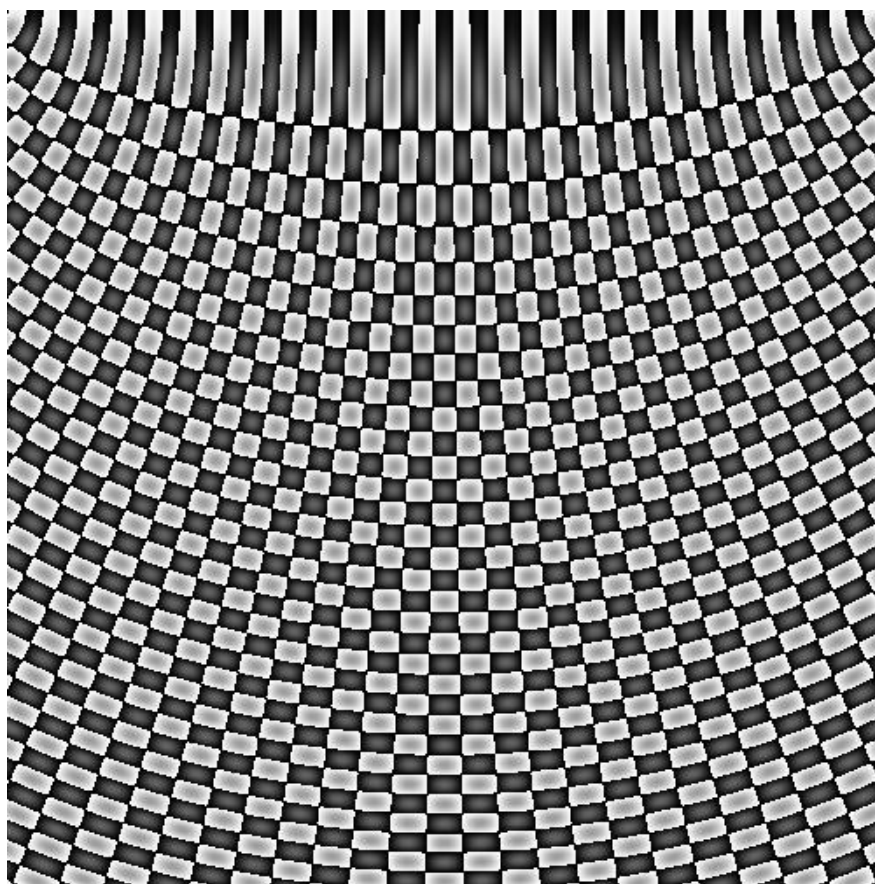


Рис. 2: Результат выполнения части 2 лабораторной работы №1.