

Zadania 1, 2, 3 oraz 4 rozwiązałem w odpowiednich plikach Python (.py). Pliki będą załączone razem ze sprawozdaniem oraz użytymi zdjęciami, domyślnym kotem i korytarzem.

Zad.1

```
## Autor: Michał Krystecki 342906

import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0, 7, 0.05)
x_param = 1.5 * np.sin(t)**3
y_param = 3 * np.cos(t) - 1.2 * np.cos(2*t) - 0.6 * np.cos(3*t)

x_sig = np.arange(-5, 5, 0.1)
k = 2
x0 = 0
y_sig = 1 / (1 + np.exp(-k * (x_sig - x0)))

img_cat = plt.imread('cat.png')
img_corridor = plt.imread('corridor.png')

fig, axs = plt.subplots(2, 2, figsize=(10, 8))

axs[0, 0].imshow(img_cat)
axs[0, 0].set_title("Lewy Górny: Kot")
axs[0, 0].axis('off')

axs[0, 1].plot(x_param, y_param)
axs[0, 1].grid(True)
axs[0, 1].set_title("Prawy Górny: Krzywa parametryczna")

axs[1, 0].plot(x_sig, y_sig)
axs[1, 0].grid(True)
axs[1, 0].set_title("Lewy Dolny: Funkcja sigmoidalna")

axs[1, 1].imshow(img_corridor)
axs[1, 1].set_title("Prawy Dolny: Korytarz")
axs[1, 1].axis('off')

fig.suptitle('Zadanie 1: Obrazy i Funkcje')
plt.tight_layout()
plt.show()
```

Komentarz 1:

Skrypt tworzy figurę składającą się z 4 różnych obrazów według instrukcji. W lewym górnym rogu znajduje się obraz kota, w lewym dolnym funkcja sigmoidalna wyrażona zadany wzorem, w prawym górnym znajduje się krzywa parametryczna, która przyjmuje kształt serca, a w prawym dolnym rogu znajduje się zdjęcie korytarza.

Zad.2

Autor: Michał Krystecki 342906

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0, 10 * np.pi, 500)
A = 1.0
lambda_coeff = 0.05
y_sin = A * np.exp(-lambda_coeff * x) * np.sin(x)
y_env_upper = A * np.exp(-lambda_coeff * x)
y_env_lower = -A * np.exp(-lambda_coeff * x)
```

```
plt.figure(figsize=(10, 6))
```

```
plt.plot(x, y_sin, label='Tłumiona sinusoida', linestyle='-')
plt.plot(x, y_env_upper, label=r'Gorna obwiednia $A e^{\{-\lambda x\}}$', linestyle='-.', color='red')
plt.plot(x, y_env_lower, label=r'Dolna obwiednia $-A e^{\{-\lambda x\}}$', linestyle='--', color='orange')
```

```
plt.title("Wykres tłumionej sinusoidy")
plt.xlabel("Czas (t)")
plt.ylabel("Amplituda")
plt.legend()
plt.grid(True)
plt.show()
```

Komentarz 2:

Skrypt generuje wykres funkcji tłumionej sinusoidy w przedziale od 0 do 10 pi oraz jej obwiednie używając różnych typów linii oraz kolorów.

Zad.3

Autor: Michał Krystecki 342906

```
from PIL import Image, ImageEnhance, ImageFilter
import matplotlib.pyplot as plt
cat = Image.open("cat.png").convert("RGB")
corridor = Image.open("corridor.png").convert("RGB")
img_paste = corridor.copy()
img_paste.paste(cat.resize((200, 150)), (850, 650))
enhancer = ImageEnhance.Sharpness(corridor)
img_sharp = enhancer.enhance(2.0)
noise = Image.effect_noise(cat.size, 50).convert("RGB")
img_noisy_cat = Image.blend(cat, noise, 0.5)
img_filter_cat = cat.filter(ImageFilter.DETAIL)
img_filter_corridor = corridor.filter(ImageFilter.MedianFilter(size=9))
fig, axs = plt.subplots(3, 4, figsize=(15, 10))
fig.suptitle('Zadanie 3: Przetwarzanie obrazów PIL')
axs[0, 0].imshow(corridor)
axs[0, 0].set_title("Original: Korytarz")
axs[0, 0].axis('off')
axs[0, 1].imshow(img_paste)
axs[0, 1].set_title("Paste: Kot w korytarzu")
axs[0, 1].axis('off')
axs[0, 2].imshow(img_sharp)
axs[0, 2].set_title("Sharpness: Wyostrzony korytarz")
axs[0, 2].axis('off')
axs[0, 3].imshow(img_filter_corridor)
axs[0, 3].set_title("Filter: Korytarz Median(9)")
axs[0, 3].axis('off')
axs[1, 0].imshow(cat)
axs[1, 0].axis('off')
axs[1, 0].set_title("Original: Kot")
axs[1, 1].imshow(img_filter_cat)
axs[1, 1].set_title("3. Filter: Kot DETAIL")
axs[1, 1].axis('off')
axs[1, 2].imshow(img_noisy_cat)
axs[1, 2].set_title("2. Blend: Zaszumiony kot")
axs[1, 2].axis('off')
# Wyłączenie pustych osi
axs[1, 3].axis('off')
axs[2, 0].axis('off')
axs[2, 1].axis('off')
axs[2, 2].axis('off')
axs[2, 3].axis('off')
plt.tight_layout()
plt.show()
```

Komentarz 3:

Kod wykorzystuje bibliotekę PIL (Pillow) do przetwarzania obrazów. Wykonuje następujące operacje:

1. Wklejenie przeskalowanego obrazu kota na obraz korytarza.
2. Wyostrenie obrazu korytarza (ImageEnhance.Sharpness).
3. Dodanie szumu do obrazu kota poprzez mieszanie (blend) z wygenerowanym szumem.
4. Zastosowanie filtrów: filtru detali (DETAIL) na kocie oraz filtru medianowego na korytarzu
5. Wyświetla je w dwóch rzędach razem z oryginalnym zdjęciem dla porównania

Zad.4

Autor: Michał Krystecki 342906

```
def gnome_sort(arr):
    n = len(arr)
    i = 1
    j = 2

    while i < n:
        if arr[i-1] <= arr[i]:
            i = j
            j = j + 1
        else:
            arr[i-1], arr[i] = arr[i], arr[i-1]
            i = i - 1
            if i == 0:
                i = 1
                j = 2
    return arr

if __name__ == '__main__':
    data = [34, 2, 10, -5, 8, 100, 1]
    print("Nieposortowana lista:", data)
    print("Posortowana lista:", gnome_sort(data))
```

Komentarz 4:

Skrypt implementuje algorytm jako skrypt a następnie wywołuje funkcję dla testowej tablicy sortując ją.