

Міністерство освіти і науки, молоді та спорту України
Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

з домашнє завдання №24.2 з дисципліни:

“Алгоритми та моделі обчислень”

Варіант: № 24.

Виконав:

ст. групи КІ-203

Ширий Богдан Ігорович

Перевірів:

ст. викладач кафедри ЕОМ

Козак Назар Богданович

ЗАВДАННЯ:

УМОВА:

Виконати домашнє завдання №24.1 повторно за допомогою Python.

ВИБІР ВАРІАНТУ:

$$(N_{\text{ж}} + N_{\text{г}} + 1) \% 10 + 1 = (24 + 3 + 1) \% 10 + 1 = 28 \% 10 + 1 = 9,$$

де: $N_{\text{ж}}$ – порядковий номер студента в групі, а $N_{\text{г}}$ – номер групи.

Отож, мій шуканий варіант – це максимум 9 облич, які можуть бути виявлені.

ВИКОНАННЯ:

Склав Python десктопну програму та її роботу зобразив на рисунках від 1 до 7.

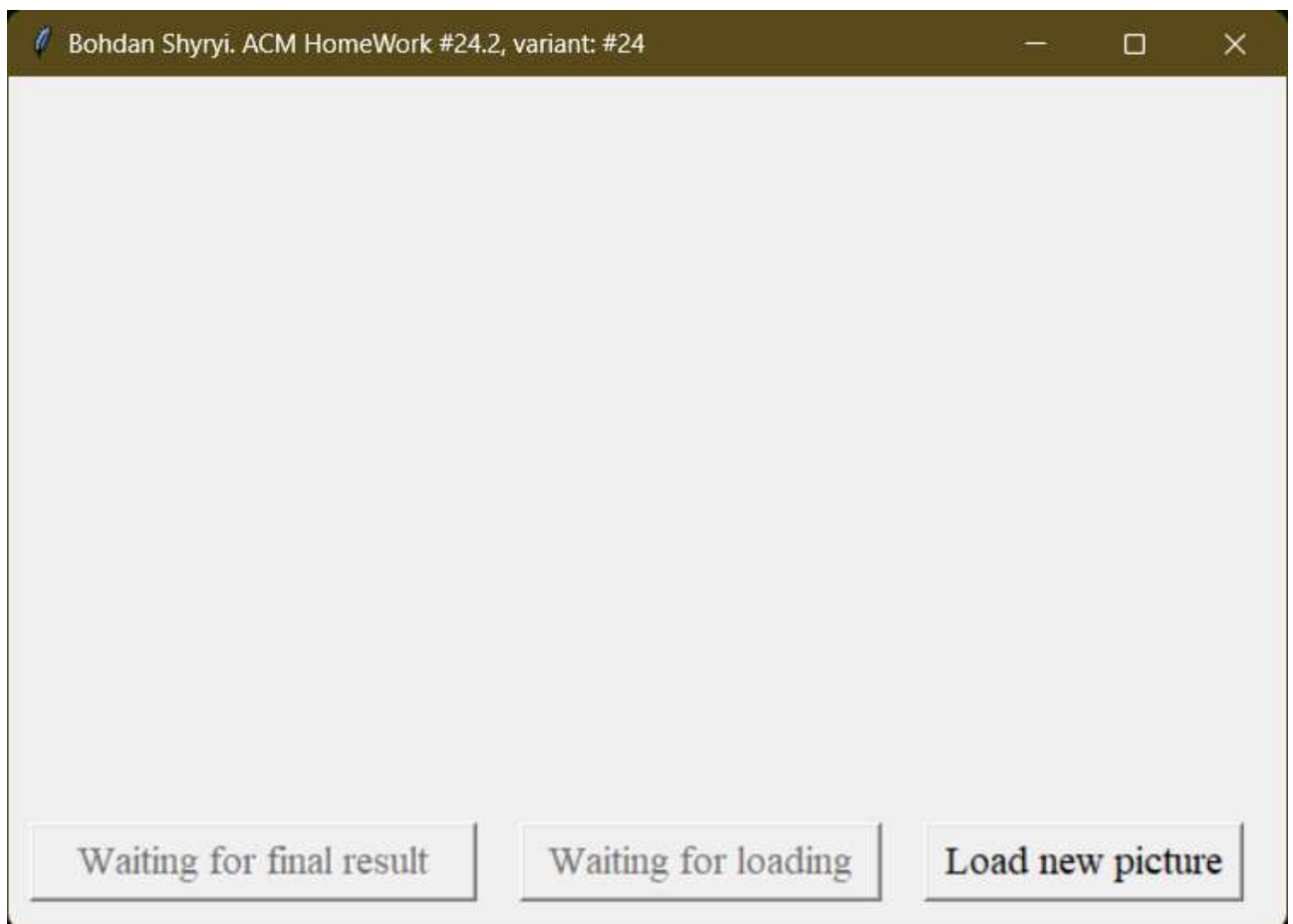


Рис. 1. Початковий вигляд програми.

Для завантаження зображення у програму, використовується діалогове вікно вибору файлу, яке дозволяє вибрати бажане зображення з файлової системи. Далі потрібно здійснити пошук облич, після чого можна буде змінювати показ між початковим або кінцевим станом обробки, або ж вибрати нове зображення і здійснювати ці операції вже з ним.

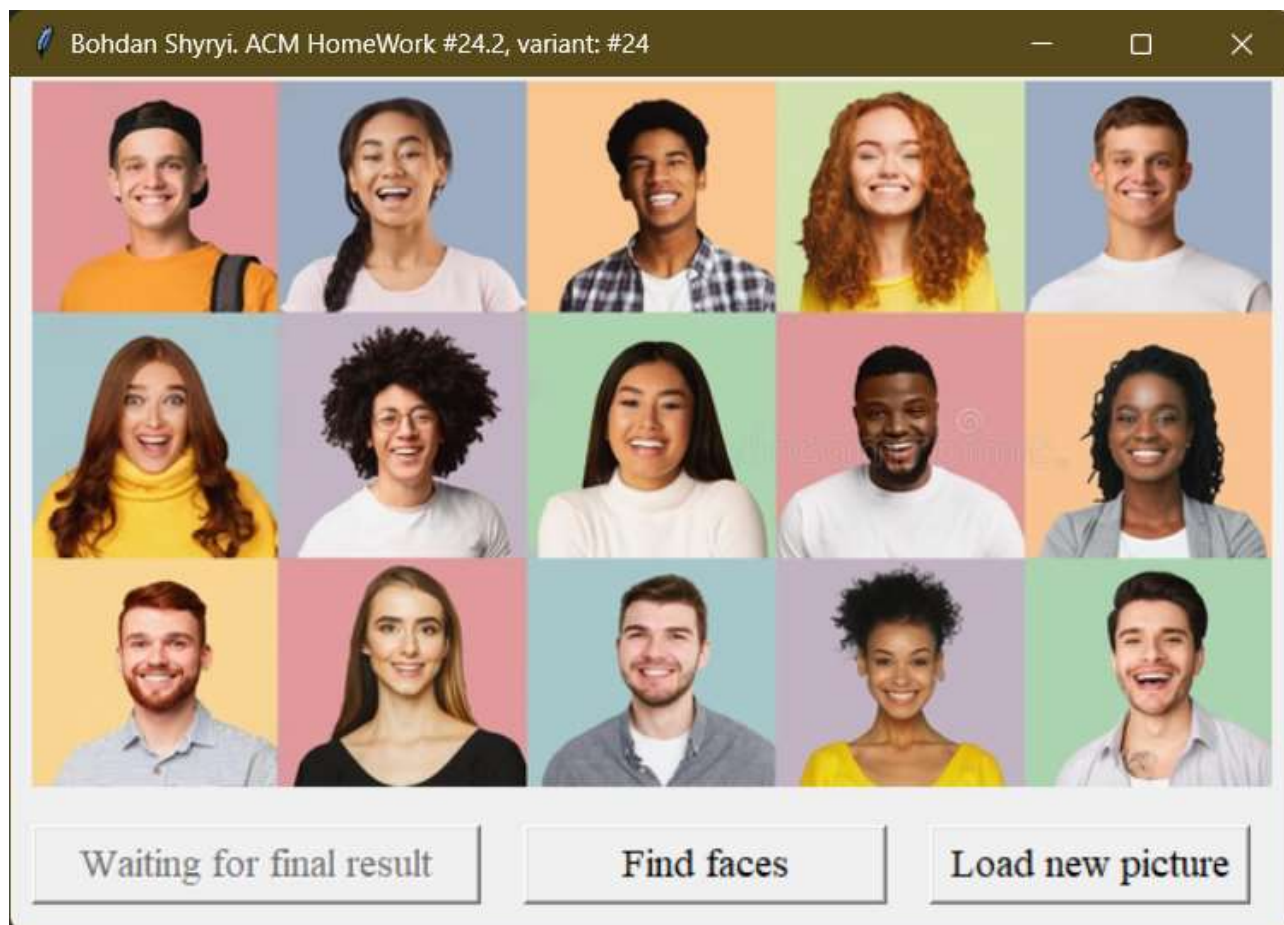


Рис. 2. Програма, після завантаження нового зображення.

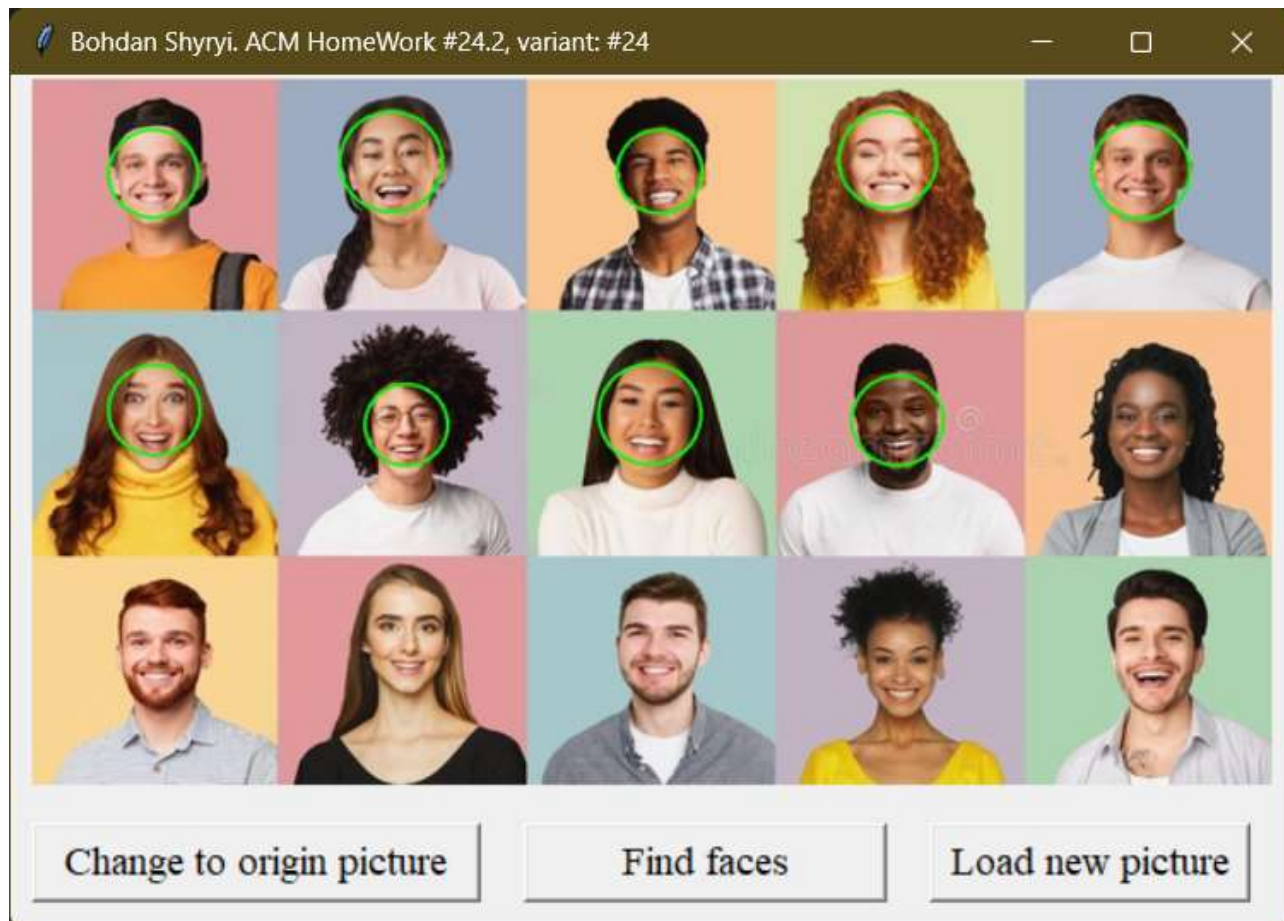


Рис. 3. Програма, після здійснення пошуку облич.

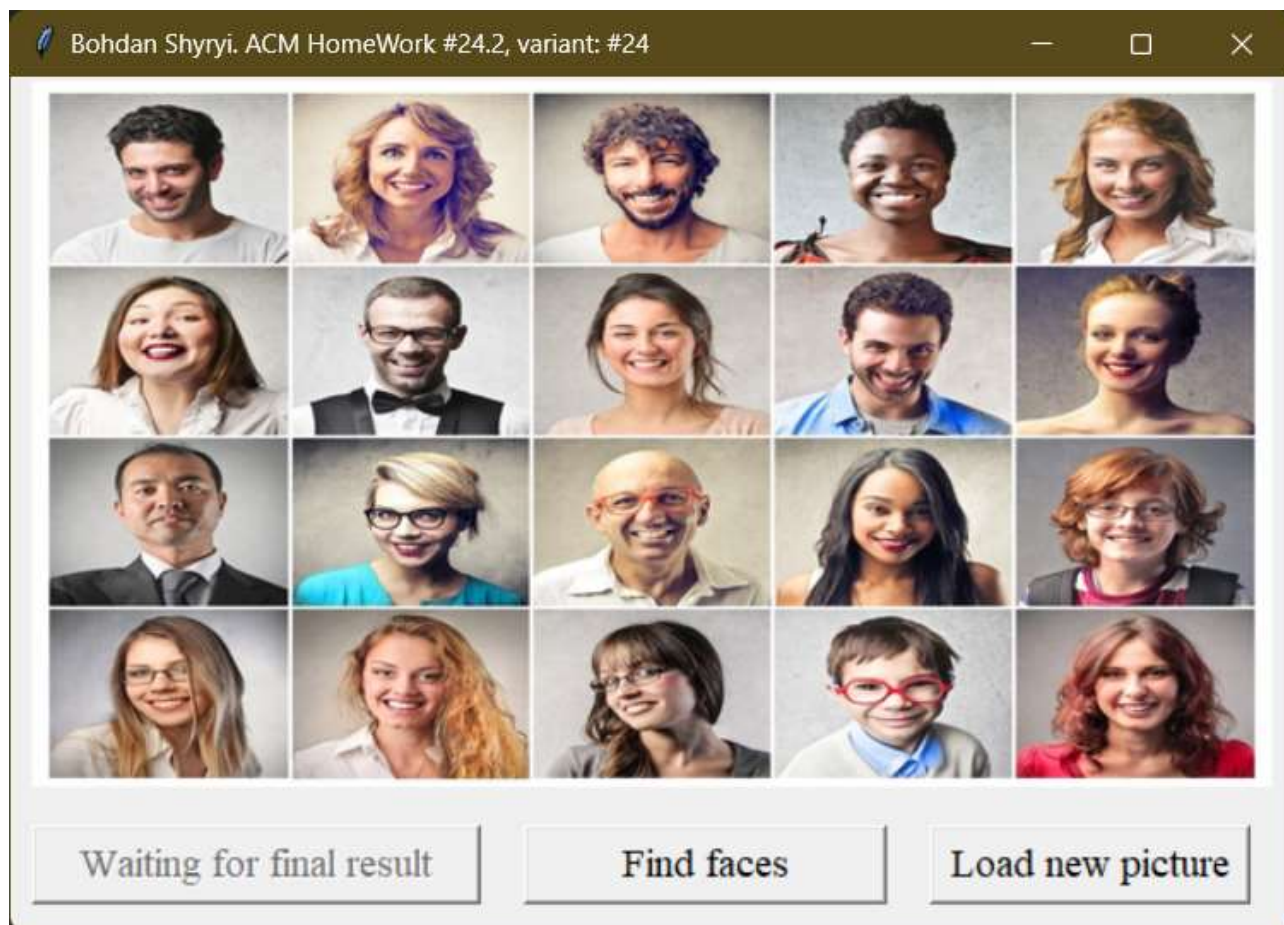


Рис. 4. Програма, після завантаження нового зображення.



Рис. 5. Програма, після здійснення пошуку облич.

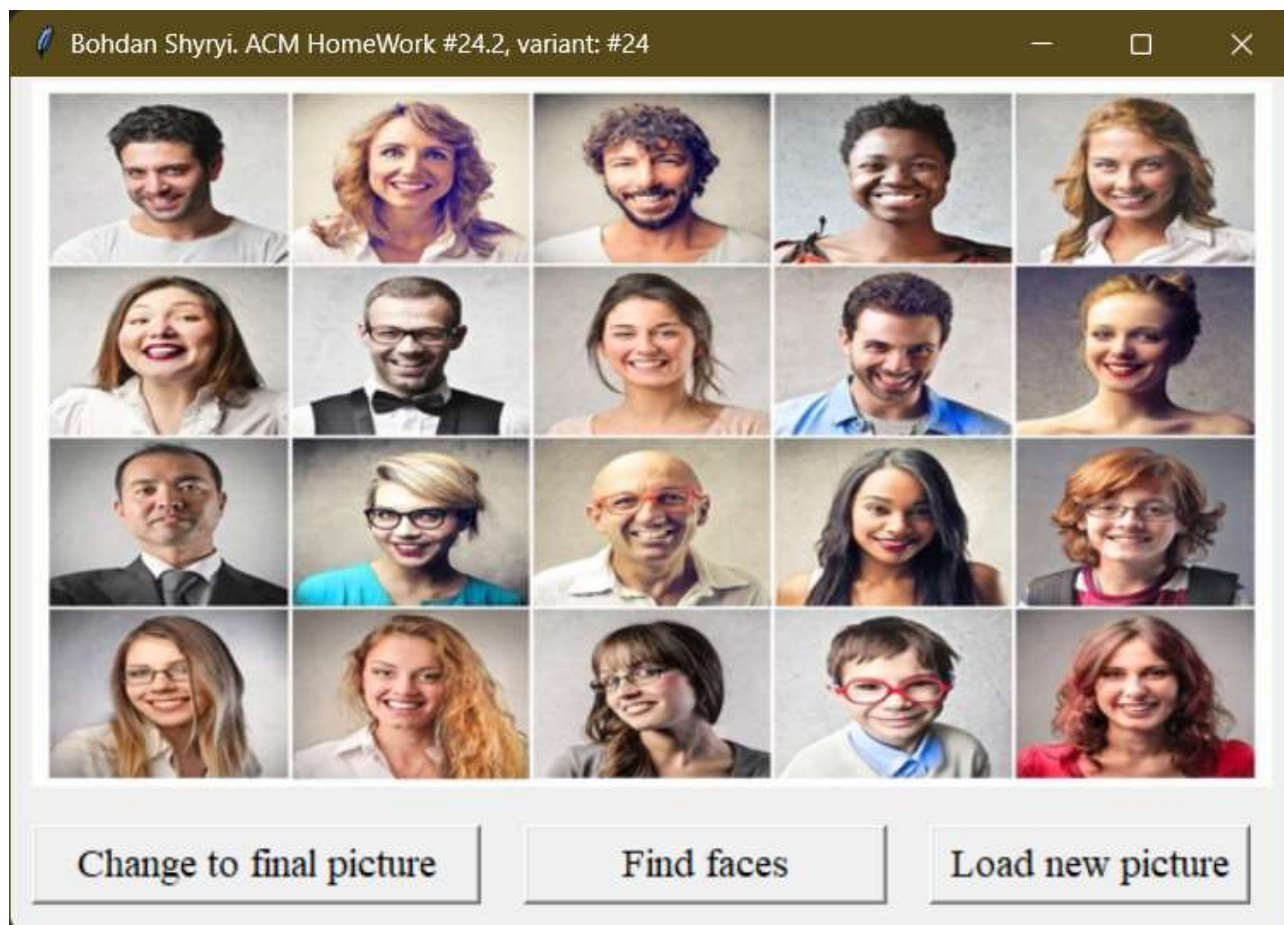


Рис. 6. Програма, після переходу до початкового зображення.



Рис. 7. Програма, після переходу до вихідного зображення.

Відповідно, у лістингу 1 навів код файлу acmShyryiHW24_2.py.

Лістинг 1. Код файлу acmShyryiHW24_2.py.

```
from tkinter import Tk, Button, Label, filedialog, messagebox
from PIL import Image, ImageTk
import tkinter.font as tkfont
import cv2

#-----#
# --- Виконання при натисканні кнопок --- #
#-----#

isFirst = False

# --- Перегляд станів зображень --- #
def change_picture_button_click():
    global isFirst

    if isFirst:
        # Зміна кнопки відповідно до ситуації
        change_picture_button.configure(text = "Change to origin picture")

        # Вивід вихідного зображення
        image = Image.open("images/OutputImage.jpg")

        # Зміна розміру зображення
        image = image.resize((580, 330), Image.ANTIALIAS)

        # Конвертація зображення у формат PhotoImage
        photo = ImageTk.PhotoImage(image)

        # Оновлення зображення у віджеті label
        label.configure(image=photo)
        label.image = photo # Запобігання знищення посилання на зображення
    else:
        # Зміна кнопки відповідно до ситуації
        change_picture_button.configure(text = "Change to final picture")

        # Вивід вихідного зображення
        image = Image.open("images/InputImage.jpg")

        # Зміна розміру зображення
        image = image.resize((580, 330), Image.ANTIALIAS)

        # Конвертація зображення у формат PhotoImage
        photo = ImageTk.PhotoImage(image)

        # Оновлення зображення у віджеті label
        label.configure(image=photo)
        label.image = photo # Запобігання знищення посилання на зображення
    isFirst = not isFirst

# --- Пошук облич --- #
def find_faces_button_click():

    # Завантаження фото
    image = cv2.imread("images/InputImage.jpg")

    # Перетворення в відтінки сірого (для покращення роботи алгоритму)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Завантаження каскадного класифікатора обличчя
    faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades +
"haarcascade_frontalface_alt.xml")

    # Виявлення обличчя
```

```

faces = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=3,
minSize=(30, 30))

# Межі областей обличчя
maxFaces = 9 # Максимальна кількість облич, яку ви хочете виявити
numFaces = min(len(faces), maxFaces)

# Позначення облич колами або овалами
for i in range(numFaces):
    (x, y, w, h) = faces[i]
    center = (x + w // 2, y + h // 2)
    axes = (w // 2, h // 2)
    cv2.ellipse(image, center, axes, 0, 0, 360, (0, 255, 0), 2)

# Збереження результату у файл
cv2.imwrite("images/OutputImage.jpg", image)

# Вивід вихідного зображення
image = Image.open("images/OutputImage.jpg")

# Зміна розміру зображення
image = image.resize((580, 330), Image.ANTIALIAS)

# Конвертація зображення у формат PhotoImage
photo = ImageTk.PhotoImage(image)

# Оновлення зображення у віджеті label
label.configure(image=photo)
label.image = photo # Запобігання знищення посилання на зображення

# Зміна кнопки відповідно до ситуації
change_picture_button.configure(text = "Change to origin picture", state =
"normal")

# --- Завантаження зображення --- #
def load_picture_button_click():
    global isFirst

    # Відкриття діалогового вікна вибору файлу
    file_path = filedialog.askopenfilename(filetypes=[("Images",
"*.png;*.jpg;*.jpeg")])

    try:
        # Зчитування зображення
        image = Image.open(file_path)

        # Зміна розміру зображення
        image = image.resize((580, 330), Image.ANTIALIAS)

        # Конвертація зображення у формат PhotoImage
        photo = ImageTk.PhotoImage(image)

        # Оновлення зображення у віджеті label
        label.configure(image=photo)
        label.image = photo # Запобігання знищення посилання на зображення

        # Збереження зображення
        Image.open(file_path).save("images/InputImage.jpg")

        # Зміна кнопок відповідно до ситуації
        find_faces_button.configure(text = "Find faces", state = "normal")
        change_picture_button.configure(text = "Waiting for final result", state =
"disabled")
        isFirst = False

    except Exception as e:
        messagebox.showinfo("That's an error my friend", e)

```

```
#-----#
# --- Створення форми та її елементів --- #
#-----#

# Створення вікна
window = Tk()
font = tkfont.Font(family="Times New Roman", size=14)

# Налаштування вікна
window.title("Bohdan Shyryi. ACM HomeWork #24.2, variant: #24")
window.geometry("600x400")

# Створення елементів
change_picture_button = Button(window, text="Waiting for final result", state =
"disabled",
                                width=20, height=1, font = font,
                                command=change_picture_button_click)
change_picture_button.place(x=10, y=350)

find_faces_button = Button(window, text="Waiting for loading", state = "disabled",
                            width=16, height=1, font = font,
                            command=find_faces_button_click)
find_faces_button.place(x=240, y=350)

load_picture_button = Button(window, text="Load new picture", width=14, height=1,
                              font = font, command=load_picture_button_click)
load_picture_button.place(x=430, y=350)

# Створення початкового зображення
image = Image
label = Label()
label.pack()

# Запуск головного циклу обробки подій
window.mainloop()
```