

**Додаток А**  
**Листінг програми Python**

```
import cv2
import numpy as np
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk, ImageDraw, ImageFont
from skimage.metrics import structural_similarity as ssim
from ultralytics import YOLO

model = YOLO("yolov8n.pt")
ref_mouse = cv2.imread("images/ mouse_etalon.jpg")
threshold_ssim = 0.56
last_image_pil = None # для збереження

def put_text_ua_cv2(img, text, position, font_path='C:/Windows/Fonts/arial.ttf',
font_size=32, color=(255,255,255)):
    img_pil = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    draw = ImageDraw.Draw(img_pil)
    font = ImageFont.truetype(font_path, font_size)
    draw.text(position, text, font=font, fill=color[::-1])
    return cv2.cvtColor(np.array(img_pil), cv2.COLOR_RGB2BGR)

def preprocess(img):
    lab = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
    cl = clahe.apply(l)
    limg = cv2.merge((cl, a, b))
    return cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

def compare_mouse_roi(roi_new, roi_ref, threshold=threshold_ssim):
```

```
roi_new = preprocess(roi_new)
roi_ref = preprocess(roi_ref)
gray_new = cv2.cvtColor(roi_new, cv2.COLOR_BGR2GRAY)
gray_ref = cv2.cvtColor(roi_ref, cv2.COLOR_BGR2GRAY)
gray_ref = cv2.resize(gray_ref, (gray_new.shape[1], gray_new.shape[0]))
score, _ = ssim(gray_ref, gray_new, full=True)
print(f"SSIM: {score:.3f}")
return score < threshold
```

```
def analyze_image(image_path):
    img = cv2.imread(image_path)
    results = model(image_path)[0]
    class_names = model.names

    mouse_present = False
    foreign_present = False
    defect = False
    mouse_box = None

    for box in results.bboxes:
        cls_id = int(box.cls)
        cls_name = class_names[cls_id]
        x1, y1, x2, y2 = box.xyxy[0].cpu().numpy().astype(int)

        if cls_name == "mouse":
            mouse_present = True
            mouse_box = (x1, y1, x2, y2)
        else:
            foreign_present = True
```

```
if mouse_present and mouse_box:
    x1, y1, x2, y2 = mouse_box
    roi_mouse = img[y1:y2, x1:x2]
    defect = compare_mouse_roi(roi_mouse, ref_mouse)

if mouse_present and mouse_box:
    if defect and foreign_present:
        status_text = "Дефект миші! І сторонній об'єкт виявлено!"
        color_status = (0, 0, 255)
    elif defect:
        status_text = "Дефект миші!"
        color_status = (0, 0, 255)
    elif foreign_present:
        status_text = "Миша і сторонній об'єкт виявлені!"
        color_status = (0, 165, 255)
    else:
        status_text = "Миша в порядку"
        color_status = (0, 255, 0)
else:
    if foreign_present:
        status_text = "Сторонній об'єкт без миші!"
        color_status = (0, 0, 255)
    else:
        status_text = "Мишу не знайдено"
        color_status = (0, 0, 255)

for box in results.bboxes:
    cls_id = int(box.cls)
    cls_name = class_names[cls_id]
    x1, y1, x2, y2 = box.xyxy[0].cpu().numpy().astype(int)
```

```

if cls_name == "mouse":
    color = (0, 0, 255) if defect else (0, 255, 0)
    label = "mouse (defect)" if defect else "mouse"
else:
    color = (0, 0, 255)
    label = cls_name

cv2.rectangle(img, (x1, y1), (x2, y2), color, 2)
cv2.putText(img, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.8, color, 2)

img = put_text_ua_cv2(img, status_text, (10, 30), font_size=32,
color=color_status)
return img, status_text

def open_and_process():
    global last_image_pil
    path = filedialog.askopenfilename(filetypes=[("Image Files", "*.jpg;*.png")])
    if not path:
        return

    img, status = analyze_image(path)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_pil = Image.fromarray(img_rgb)
    img_pil.thumbnail((800, 600))
    last_image_pil = img_pil.copy()

    imgtk = ImageTk.PhotoImage(img_pil)
    image_label.configure(image=imgtk)

```

```

image_label.image = imgtk
result_text.set(status)

def save_result():
    if last_image_pil is None:
        messagebox.showinfo("Збереження", "Спочатку оберіть та обробіть
зображення.")
        return
    path = filedialog.asksaveasfilename(defaultextension=".jpg",
filetypes=[("JPEG files", "*.jpg")])
    if path:
        last_image_pil.save(path)
        messagebox.showinfo("Збереження", f"Зображення збережено у
{path}")

def show_about():
    messagebox.showinfo("Про програму", "Система комп'ютерного зору
для автоматизованої системи управління промисловими роботами.\nАвтор:
Фірсов Д.В.")

# === GUI ===
root = tk.Tk()
root.title("Система комп'ютерного зору")
root.geometry("900x720")

# Меню
menubar = tk.Menu(root)
filemenu = tk.Menu(menubar, tearoff=0)
filemenu.add_command(label="Зберегти зображення",
command=save_result)

```

```
filemenu.add_separator()
filemenu.add_command(label="Вийти", command=root.quit)
menubar.add_cascade(label="Файл", menu=filemenu)

helpmenu = tk.Menu(menubar, tearoff=0)
helpmenu.add_command(label="Про програму", command=show_about)
menubar.add_cascade(label="Допомога", menu=helpmenu)

root.config(menu=menubar)

# Інтерфейс
tk.Label(root, text="Розпізнавання миші та сторонніх об'єктів",
font=("Arial", 18, "bold")).pack(pady=10)
tk.Button(root, text="Обрати зображення", font=("Arial", 16),
command=open_and_process).pack(pady=10)

image_label = tk.Label(root)
image_label.pack()

result_text = tk.StringVar()
tk.Label(root, textvariable=result_text, font=("Arial", 16),
fg="blue").pack(pady=10)

root.mainloop()
```

**Додаток Б**  
**Демонстраційний матеріал**

