**Министерство Образования и Исследований**

**Республики Молдова**

**Технический Университет Молдовы**

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Программной Инженерии и Автоматики

**Отчет**

по лабораторной работе Nr.3 по

**Декларативное программирование**

Тема: **Обработка данных**.

Выполнила ст.гр. TI-217 Выртикуш Александру

Проверил Аркадий Лях

Кишинев – 2023

**Вариант 3–10**

1.Написать скрипт для формирования текстового файла-аннотации собранного датасета. Файл-аннотация должен представлять собой csv-файл, в котором в первой колонке будет указан абсолютный путь к файлу, во второй колонке относительный путь относительно вашего Python-проекта, третья колонка будет содержать текстовое название класса (метку класса), к которому относится данный экземпляр.

**Реализация:**

**Код:**

import os

import csv

# Указать путь к корневой директории датасета

dataset\_root = 'D:\\UTM\\PD\\PD\_git\_lab\\lab\_2'

project\_root = 'D:\\UTM\\PD\\PD\_git\_lab\\lab\_3'

# Создать\\открыть файл аннотации

with open('annotations.csv', 'w', newline='') as csv\_file:

    writer = csv.writer(csv\_file)

    writer.writerow(['D:\\UTM\\PD\\PD\_git\_lab\\lab\_2\\dataset', '..\\lab\_2\\dataset', 'zebra'])

    for root, dirs, files in os.walk(dataset\_root):

        for file in files:

            if file.endswith('.jpg'):

                absolute\_path = os.path.join(root, file)

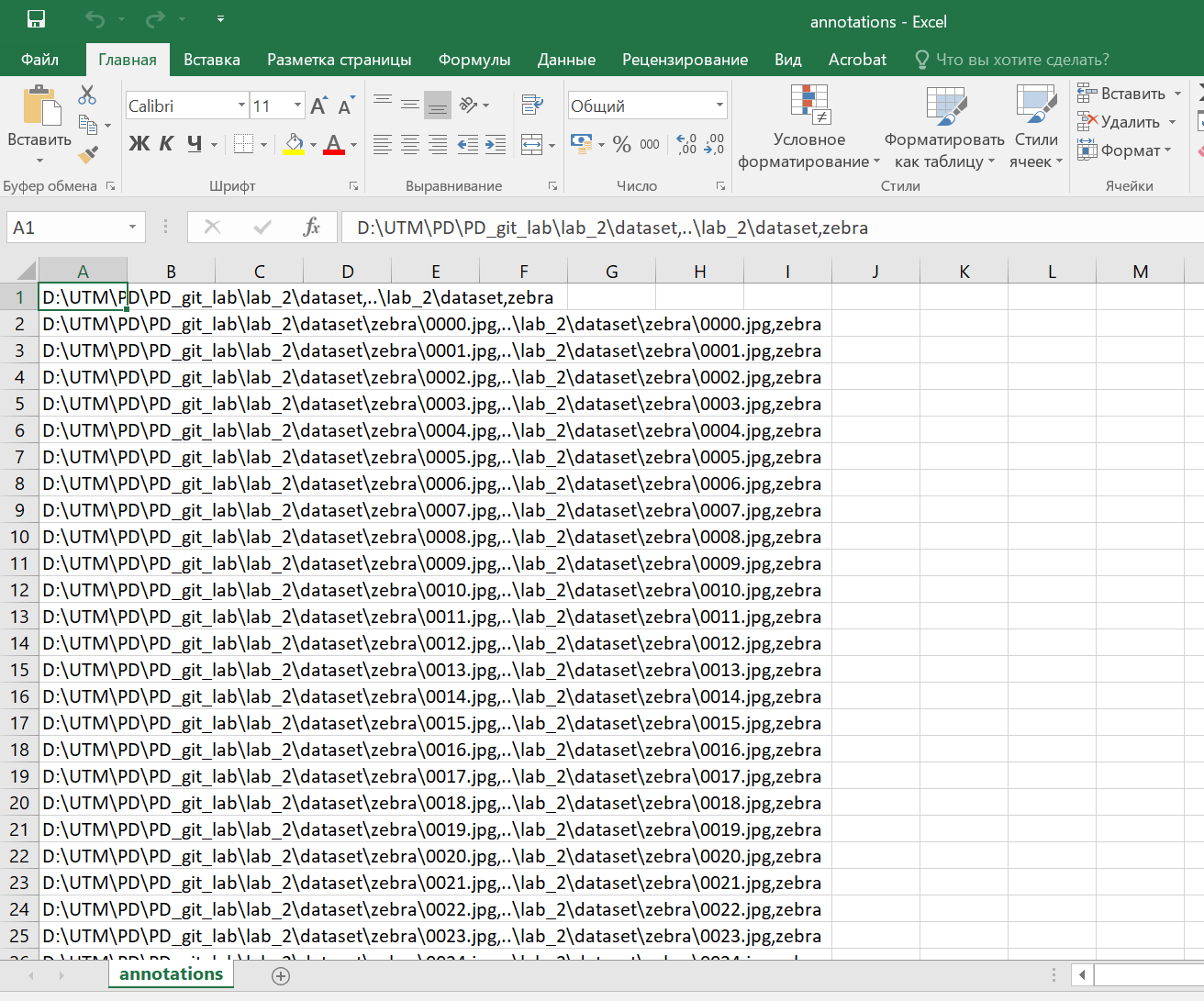
                relative\_path = os.path.relpath(absolute\_path, project\_root)

                label = os.path.basename(root)

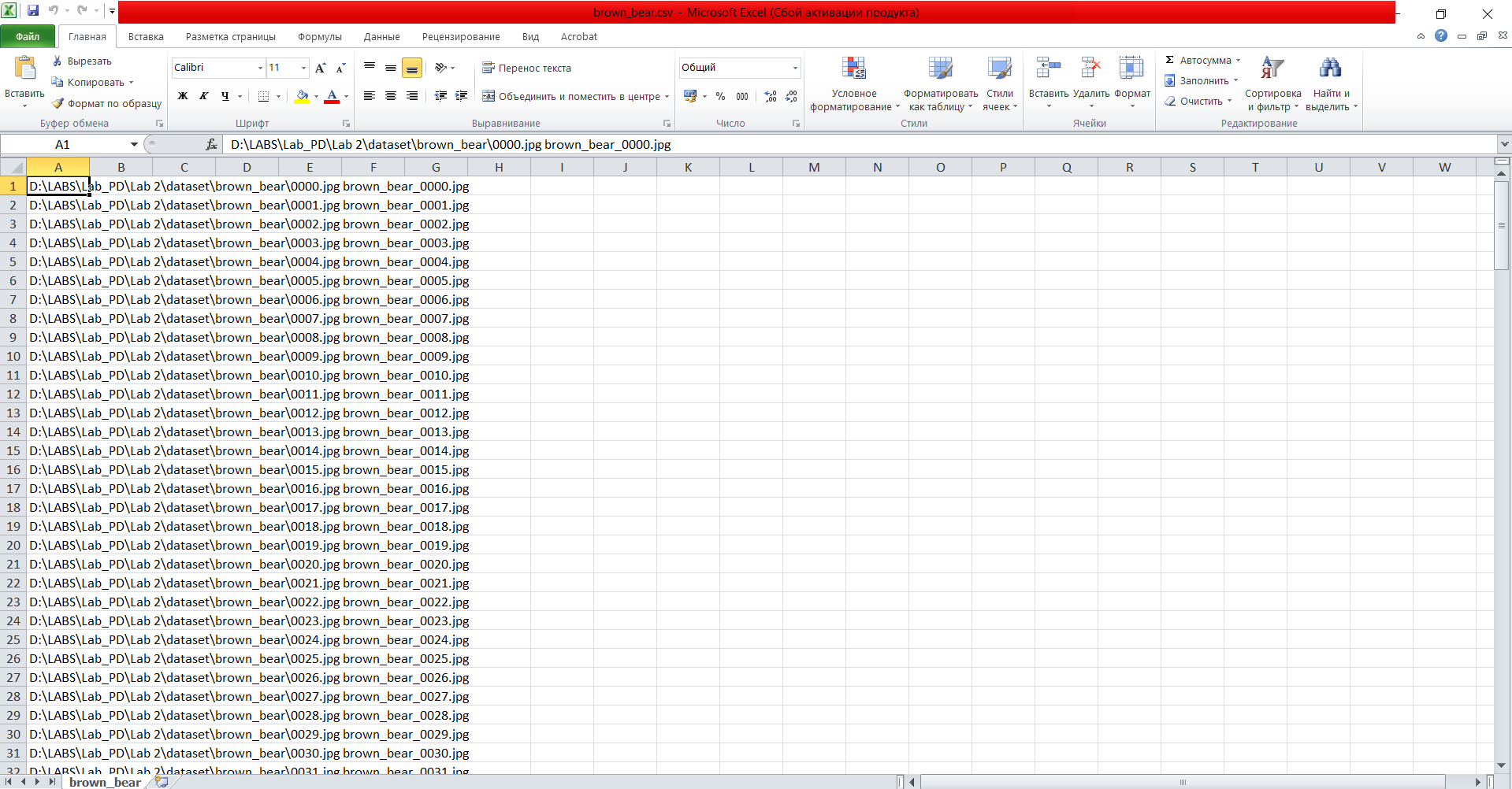
                writer.writerow([absolute\_path, relative\_path, label])

2.Написать скрипт для копирования датасета в другую директорию таким образом, чтобы имена файлов содержали имя класса и его порядковый номер. То есть из dataset/class/0000.jpg должно получиться dataset/class\_0000.jpg. Для того чтобы осталась возможность определить принадлежность экземпляра к классу создать файл-аннотацию (как в пункте 1).

**Реализация:**

Файл-аннотация для класса “zebra”: 

Файл-аннотация для класса “brown\_bear”:



**Код:**

import os

import shutil

# Путь к исходному датасету

source\_directory = "D:\\UTM\\PD\\PD\_git\_lab\\lab\_2\\dataset"

# Путь к целевой директории

target\_directory = "D:\\UTM\\PD\\PD\_git\_lab\\lab\_3"

# Создайте список классов

classes = ["zebra"]

# Функция для создания файла аннотации

def create\_annotation\_file(class\_name, original\_filename, target\_filename):

    annotation\_file = os.path.join(target\_directory, f"{class\_name}.csv")

    with open(annotation\_file, "a") as f:

        f.write(f"{original\_filename} {target\_filename}\n")

# Пройдем по всем классам

for class\_name in classes:

    class\_source\_directory = os.path.join(source\_directory, class\_name)

    if not os.path.exists(target\_directory):

        os.makedirs(target\_directory)

    file\_list = os.listdir(class\_source\_directory)

    for i, filename in enumerate(file\_list):

        if filename.endswith(".jpg"):

            original\_file\_path = os.path.join(class\_source\_directory, filename)

            new\_file\_name = f"{class\_name}\_{i:04d}.jpg"

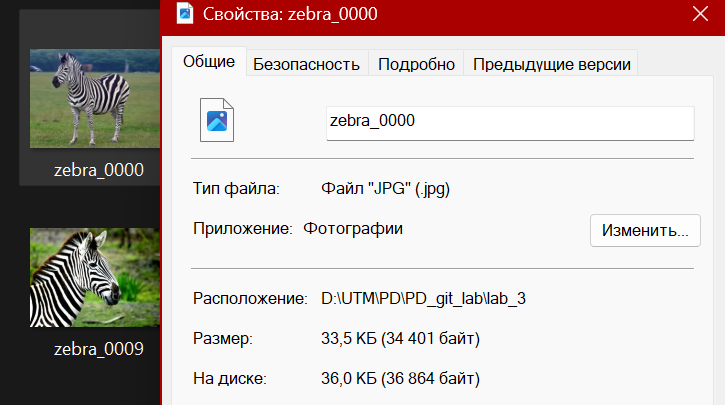
            target\_file\_path = os.path.join(target\_directory, new\_file\_name)

            shutil.copy(original\_file\_path, target\_file\_path)

            create\_annotation\_file(class\_name, original\_file\_path, new\_file\_name)

3.Написать скрипт, создающий копию датасета таким образом, чтобы каждый файл из исходного датасета получил случайный номер от 0 до 10000, и датасет представлял собой следующую структуру dataset/номер.jpg.

**Реализация:**

****

**Код:**

import os

import random

import shutil

# Путь к исходному датасету

source\_directory = 'D:\\UTM\\PD\\PD\_git\_lab\\lab\_2\\dataset'

classes = ["zebra"]

# Путь к новой копии датасета

target\_directory = 'D:\\UTM\\PD\\PD\_git\_lab\\lab\_3\\dataset3'

# Создаем новую директорию для копии датасета

os.makedirs(target\_directory, exist\_ok=True)

# Получаем список файлов из исходного датасета

for class\_name in classes:

    class\_source\_directory = os.path.join(source\_directory, class\_name)

    files = os.listdir(class\_source\_directory)

    for filename in files:

        # Генерируем случайный номер от 0 до 10000

        random\_number = random.randint(0, 10000)

        # Создаем новое имя файла

        new\_filename = os.path.join(target\_directory, f"{random\_number}.jpg")

        # Копируем файл с новым именем

        shutil.copy(os.path.join(class\_source\_directory ,filename), new\_filename

4.Написать скрипт, содержащий функцию, получающую на входе метку класса и возвращающую следующий экземпляр (путь к нему) этого класса. Экземпляры идут в любом порядке, но не повторяются. Когда экземпляры заканчиваются, функция возвращает None.

**Реализация:**

****

**Код:**

import os

import random

def get\_next\_instance(class\_label, dataset\_path):

    class\_files = [f for f in os.listdir(dataset\_path) if f.startswith(class\_label)]

    if class\_files:

        random.shuffle(class\_files)

        return os.path.join(dataset\_path, class\_files.pop())

    else:

        return None

# Пример использования функции

next\_instance = get\_next\_instance('zebra', "D:\\UTM\\PD\\PD\_git\_lab\\lab\_3\\dataset3")

print(f'Следующий экземпляр класса zebra: {next\_instance}')

**Вывод:**

В процессе выполнения лабораторной работы я освоил навык обработки данных. В рамках первой задачи был разработан скрипт для формирования файла-аннотации в формате CSV для датасета, содержащего информацию о путях к файлам, их относительных путях и метках классов для каждого экземпляра. Во второй задаче был создан скрипт для перемещения датасета в другую директорию с переименованием файлов, включающим в себя имя класса и порядковый номер. Третья задача включала в себя написание скрипта для создания копии датасета, где каждый файл получает уникальный случайный номер от 0 до 10000 в имени. Наконец, в четвертой задаче была разработана функция, которая по метке класса возвращает следующий экземпляр этого класса.