Министерство Образования, Культуры и Исследований Технический Университет Молдовы Департамент Программной Инженерии и Автоматики

**ОТЧЕТ**

Лабораторная работа №2

По предмету: **Programarea Declarativa**

Тема: **Автоматизированный сбор данных. Работа со строками.**

Выполнил: Решетников Максим

Проверил: Лях Аркадий

Кишинев 2023 г.

**Вариант 7.**

С использованием страницы <https://google.com/>сформировать запросы для поиска изображений, контент на которых соответствует классам \***zebra**\* и \***bay horse***. Для каждого класса должно быть загружено не менее 20 изображений. Изображения для каждого класса должны находиться в подпапке папки***dataset**\* с соответсвующим названием.

**Не допускается**:

1. Создание папок вручную. В коде должен быть отражен процесс создания папок и перемещения/загрузки в них файлов.
2. Дублирование изображений для класса.

**Примечания**

Каждое изображение должно иметь расширение *.jpg*

Именовать файлы необходимо порядковым номером (от 0 до 999).

Для дальнейшего удобства необходимо дополнять имя файла ведующими нулями (например, 0000, 0001, ..., 0999). Для этого необходимо использовать один из методов класса **str**.

После загрузки всех изображений, необходимо их просмотреть на соответствие классу. В случае замеченных несоответствий необходимо будет дополнить набор данных до минимального размера. Для избежания подобных ситуаций рекомендуется загружать изображения с запасом.

Вариант подразумевает два уровня сложности:

1. Для первого уровня сложности достаточно загрузить лишь миниатюры изображений.
2. Для второго уровня сложности необходимо загрузить полноразмерные изображения.

Код:

import os

import requests

from bs4 import BeautifulSoup

import time

# Функция для создания папок и загрузки изображений по URL с задержкой

def download\_images(url, folder\_path, class\_name, num\_images):

    os.makedirs(os.path.join(folder\_path, 'dataset\_' + class\_name), exist\_ok=True)

    response = requests.get(url)

    soup = BeautifulSoup(response.text, 'html.parser')

    img\_tags = soup.find\_all('img')

    downloaded\_count = 0

    for img\_tag in img\_tags:

        img\_url = img\_tag.get('src')

        if img\_url and img\_url.startswith('http'):

            img\_data = requests.get(img\_url).content

            img\_name = str(downloaded\_count).zfill(4) + '.jpg'

            img\_path = os.path.join(folder\_path, 'dataset\_' + class\_name, img\_name)

            with open(img\_path, 'wb') as img\_file:

                img\_file.write(img\_data)

            downloaded\_count += 1

            if downloaded\_count >= num\_images:

                break

            # Задержка между запросами

            time.sleep(3)

    return downloaded\_count

# URL-ы для поиска изображений зебры и байской лошади на Google Images

zebra\_url = 'https://www.google.com/search?q=zebra&tbm=isch'

horse\_url = 'https://www.google.com/search?q=bay+horse&tbm=isch'

# Путь к папке, где будут сохранены изображения

folder\_path = 'D:/Test/dataset'

# Количество изображений, которые нужно загрузить для каждого класса

num\_images = 30

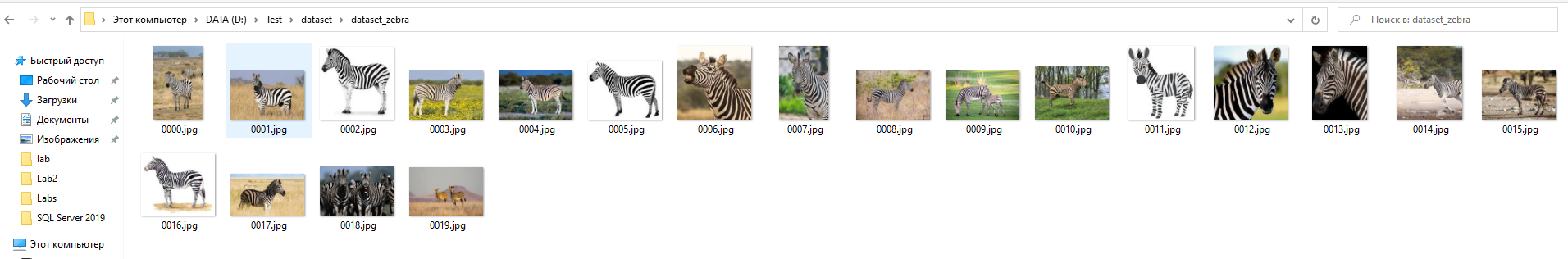
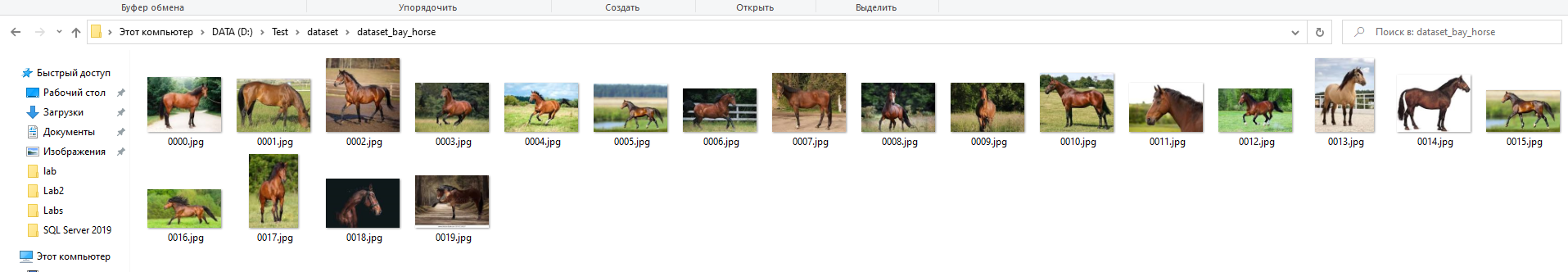
# Загрузка изображений зебры

zebra\_count = download\_images(zebra\_url, folder\_path, 'zebra', num\_images)

# Загрузка изображений байской лошади

horse\_count = download\_images(horse\_url, folder\_path, 'bay\_horse', num\_images)

GitHub: <https://github.com/YumeGit/Labs>

Выполнение: 

**Вывод:**

В ходе выполнения данной лабораторной работы мы изучили использование Python и составили программу по заданию.