Министерство Просвещения, Культуры и Исследований Республики Молдова

Технический университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Микроэлектроники и Биоинженерии

**Отчет**

**По предмету Programarea Declarativă**

**По лабораторной работе №2**

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

Выполнил Cuțar Alexandr (TI-218)

Проверил A.Leah

Кишинев 2023

# **Задания к самостоятельной работе**

**Вариант 7.**

С использованием страницы <https://yandex.ru/images/> сформировать запросы для поиска изображений, контент на которых соответствует классам \***zebra**\* и \***bay horse***. Для каждого класса должно быть загружено не менее 1000 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 main.py:**

import os  
import requests  
from bs4 import BeautifulSoup  
from urllib.parse import urlparse, urljoin  
  
# Создаем папку dataset, если она не существует  
if not os.path.exists('dataset'):  
 os.mkdir('dataset')  
  
  
# Функция для загрузки изображений и сохранения их в подпапке dataset  
def download\_images(search\_query, class\_name, num\_images=1000):  
 # Создаем подпапку с соответствующим именем  
 folder\_path = os.path.join('dataset', class\_name)  
 if not os.path.exists(folder\_path):  
 os.mkdir(folder\_path)  
  
 # URL страницы поиска изображений на Yandex  
 search\_url = f'https://yandex.ru/images/search?text={search\_query}&type=photo'  
  
 # Загружаем страницу поиска  
 response = requests.get(search\_url)  
 soup = BeautifulSoup(response.text, 'html.parser')  
 print(soup.prettify())  
  
 # Находим все ссылки на изображения  
 img\_tags = soup.find\_all('img', class\_='serp-item\_\_thumb')  
 img\_urls = [img['src'] for img in img\_tags if 'src' in img.attrs]  
  
 # Счетчик для имен файлов  
 file\_count = 0  
  
 # Загружаем изображения и сохраняем их  
 for img\_url in img\_urls:  
 if file\_count >= num\_images:  
 break  
  
 try:  
 # Проверяем, есть ли схема в URL изображения  
 parsed\_url = urlparse(img\_url)  
 if not parsed\_url.scheme:  
 img\_url = urljoin(search\_url, img\_url)  
  
 response = requests.get(img\_url, stream=True)  
 if response.status\_code == 200:  
 # Генерируем имя файла с ведущими нулями  
 filename = str(file\_count).zfill(4) + '.jpg'  
 file\_path = os.path.join(folder\_path, filename)  
  
 # Сохраняем изображение  
 with open(file\_path, 'wb') as file:  
 for chunk in response.iter\_content(1024):  
 file.write(chunk)  
  
 file\_count += 1  
 except Exception as e:  
 print(f"Ошибка при загрузке изображения: {e}")  
  
  
# Загружаем изображения для класса "zebra"  
download\_images('zebra', 'zebra', num\_images=1000)  
  
# Загружаем изображения для класса "bay horse"  
download\_images('bay\_horse', 'bay\_horse', num\_images=1000)  
  
print("Загрузка изображений завершена.")

**Код main\_hard.py:**

import os  
import requests  
from bs4 import BeautifulSoup  
from urllib.parse import urlparse, urljoin  
import time;  
  
# Создаем папку dataset, если она не существует  
if not os.path.exists('dataset\_hard'):  
 os.mkdir('dataset\_hard')  
  
  
# Функция для загрузки изображений и сохранения их в подпапке dataset  
def download\_images(search\_query, class\_name, num\_images=1000):  
 # Создаем подпапку с соответствующим именем  
 folder\_path = os.path.join('dataset\_hard', class\_name)  
 if not os.path.exists(folder\_path):  
 os.mkdir(folder\_path)  
  
 # URL страницы поиска изображений на Yandex  
 search\_url = f'https://yandex.ru/images/search?text={search\_query}&type=photo'  
  
 # Загружаем страницу поиска  
 response = requests.get(search\_url)  
 soup = BeautifulSoup(response.text, 'html.parser')  
  
  
 img\_a = soup.find\_all('a', class\_='serp-item\_\_link')  
 img\_a\_urls = [img\_href['href'] for img\_href in img\_a]  
  
 # Находим все ссылки на изображения  
  
 # Счетчик для имен файлов  
 file\_count = 0  
  
 # Загружаем изображения и сохраняем их  
 for a in img\_a\_urls:  
 if file\_count >= num\_images:  
 break  
  
 try:  
 search\_url\_link = f'https://yandex.ru{a}'  
 response\_image = requests.get(search\_url\_link)  
 soup\_image = BeautifulSoup(response\_image.text, 'html.parser')  
 print(soup\_image.prettify())  
  
 img\_urls = soup\_image.find\_all('img', class\_='MMImage-Origin')  
 img\_url = [img['src'] for img in img\_urls]  
 # Проверяем, есть ли схема в URL изображения  
 parsed\_url = urlparse(img\_url)  
 if not parsed\_url.scheme:  
 img\_url = urljoin(search\_url, img\_url)  
  
 response = requests.get(img\_url, stream=True)  
 if response.status\_code == 200:  
 # Генерируем имя файла с ведущими нулями  
 filename = str(file\_count).zfill(4) + '.jpg'  
 file\_path = os.path.join(folder\_path, filename)  
  
 # Сохраняем изображение  
 with open(file\_path, 'wb') as file:  
 for chunk in response.iter\_content(1024):  
 file.write(chunk)  
 file\_count += 1  
 time.sleep(0.5)  
 except Exception as e:  
 print(f"Ошибка при загрузке изображения: {e}")  
  
  
  
# Загружаем изображения для класса "zebra"  
download\_images('zebra', 'zebra', num\_images=1000)  
  
# Загружаем изображения для класса "bay horse"  
download\_images('bay\_horse', 'bay\_horse', num\_images=1000)  
  
print("Загрузка изображений завершена.")