Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

**ОТЧЕТ**

по учебной практике

на кафедре Прикладной Математики и Кибернетики

Выполнил:

студент гр. ИП-014 \_\_\_\_\_\_\_\_\_\_\_\_\_/Альхимович М.В./

ФИО студента

«28» мая 2022г.

Руководитель практики

доцент каф. ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_/Приставка П.А./

«28» мая 2022г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2022 г.

Задание по учебной практике для студентов очной формы обучения направления

09.03.01 «Информатика и вычислительная техника» профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Текст задания

Разработать программу реализующую ввод, хранение и обработку данных о котировках криптовалют на основе данных сайта coinmarketcap.com.

Общие требования к программе:

1. Язык разработки: **Python версии не ниже 3.x**
2. Операционная система: определяются студентом
3. Набор свойств криптовалют:
   * Name – наименование
   * Market\_cap – рыночная капитализация
   * Price – стоимость 1 ед. в долларах США (USD)
4. Ввод данных

|  |  |
| --- | --- |
| Оценки «хорошо» и  «удовлетворительно» | Оценка «отлично» |
| Из файла currencies22.csv. Файл содержит данные о 25 наиболее ценных криптовалютах на 05.03.2022 в формате  Name; Market\_cap; Price  Файл доступен для скачивания в ЭИОС в директории с заданием на практику. | Непосредственно с главной страницы страницы сайта coinmarketcap.com в момент запуска программы.  Загрузка и парсинг веб-страницы производится с помощью библиотек Requests и Beautifulsoup или их аналогов  Примечание: допускается считывание строчек в количестве менее 25 (Например, 10 строчек с  данными о криптовалютах) |

1. Хранение

Типы и структуры для хранения данных: определяются студентом

1. Обработка

Реализовать функцию поиска информации о свойствах криптовалюты по ее названию.

**Описание алгоритмов**

**Стандарт Python:** Python 3.10.4

**Используемые библиотеки и модули:**

**CSV (comma-separated value)** - это формат представления табличных данных (например, это могут быть данные из таблицы или данные из БД).

В стандартной библиотеке Python есть модуль csv, который позволяет работать с файлами в CSV формате.

import csv

**Requests -** библиотека, позволяющая отправлять запросы HTTP в Python и получать ответы, соответственно.

import requests

**Requests.exceptions -** расширение для библиотеки Requests, которое необходимо для отлавливания исключений ответов с запроса, таких как:

ConnectionError - ошибка соединения, Timeout - превышено ожидание ответа, TooManyRedirects - слишком много перенаправлений.

from requests.exceptions import ConnectionError, Timeout, TooManyRedirects

**BeautifulSoup -** это пакет Python для анализа документов HTML и XML. Он создает дерево синтаксического анализа для проанализированных страниц, которое можно использовать для извлечения данных из HTML, что полезно для парсинга веб-страниц.

from bs4 import BeautifulSoup

**Используемые функции:**

**def fill\_struct\_list(): -** функция; создает двумерный массив, который содержит в себе структуры, сформированные с помощью csv.reader по разделителю ';' из исходной таблицы. Затем форматирует информацию в удобный для работы и чтения вид.

Возвращаемое значение: массив структур tmp\_struct\_list.

**def print\_table(struct\_list): -** процедурная функция; создает структуру с помощью конструктора PrettyTable(), а затем построчно добавляет информацию в данную структуру по заранее заданным нами полям "Name", "Market\_cap", "Price". Выводит всю информацию в виде приятной для глаза таблицы.

**def search\_table(struct\_list): -** функция; функция поиска по ключу в переданной структуре. Находит по ключу необходимое поле и создает новую структуру, в которую будут переданы все строки таблицы с необходимыми данными.

Возвращаемое значение: массив структур fined\_elements\_arr.

**def parse\_site(): -** функция; создает header, который был в ручную добавлен при первом вхождении мной на сайт: User-Agent. Он необходим для того, чтобы в последующих запросах алгоритма на сайт, сам сайт расценивал нас как пользователя, а не скрипт. Таким образом, мы сможем отправлять запросы без риска быть заблокированы защитой сайта. После этого, была считана вся информация с сайта. Затем выделена необходимая информация по тегу "tbody", а далее найдена вся информация по тегам "p", "span" и "div", для нахождения параметров таблицы Name, Market\_cap и Price, соответственно. Далее вся информация занесена по прошлому методу в массив структур с данными.

Возвращаемое значение: массив структур tmp\_struct\_list.

**main():** - интерфейс пользователя и разделение на задачи для считывания с файла и с сайта.

**Листинг программы**

import csv

import requests

from bs4 import BeautifulSoup

from prettytable import PrettyTable

from requests.exceptions import ConnectionError, Timeout, TooManyRedirects

def fill\_struct\_list():

tmp\_struct\_list = []

try:

file\_path = open("currencies22.csv", "r")

table = csv.reader(file\_path, delimiter=';')

for i in table:

item = {"Name": i[0], "Market\_cap": i[1], "Price": i[2]}

tmp\_struct\_list.append(item)

file\_path.close()

except FileNotFoundError:

print("No file")

return tmp\_struct\_list

def print\_table(struct\_list):

table = PrettyTable()

table.field\_names = ["Name", "Market\_cap", "Price"]

for i in struct\_list:

table.add\_row([i["Name"], i["Market\_cap"], i["Price"]])

print(table)

def search\_table(struct\_list):

print("Write key for search")

key = input()

fined\_elements\_arr = []

for i in struct\_list:

if i["Name"] == key:

fined\_elements\_arr.append(i)

return fined\_elements\_arr

def parse\_site():

tmp\_struct\_list = []

headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.84 Safari/537.36 OPR/85.0.4341.79"}

try:

response = requests.get(url="https://coinmarketcap.com", headers=headers)

print(response) # Response [200] - ответ: успешно полуили ответ с сервера и зашли на него

soup = BeautifulSoup(response.text, "lxml")

html\_table = soup.find("tbody")

name\_arr = html\_table.find\_all("p", class\_="sc-1eb5slv-0 iworPT")

market\_cap\_arr = html\_table.find\_all("span", class\_="sc-1ow4cwt-1 ieFnWP")

price\_arr = html\_table.find\_all("div", class\_="sc-131di3y-0 cLgOOr")

for i in range(len(name\_arr)):

item = {"Name": name\_arr[i].text, "Market\_cap": market\_cap\_arr[i].text, "Price": price\_arr[i].text}

tmp\_struct\_list.append(item)

except(ConnectionError, Timeout, TooManyRedirects) as exp:

print("Error", exp, sep=":")

return tmp\_struct\_list

def main():

print("Write programm mode:",

"1 - Read from file.",

"2 - Read from site.",

"e - Exit", sep='\n')

struct\_list = []

while True:

input\_key = input()

if input\_key in ['1', '2', 'e']:

if input\_key == 'e':

break;

elif input\_key == '1':

exit\_flag = 0

struct\_list = fill\_struct\_list()

print\_table(struct\_list)

print\_table(search\_table(struct\_list))

elif input\_key == '2':

struct\_list = parse\_site()

print\_table(struct\_list)

print\_table(search\_table(struct\_list))

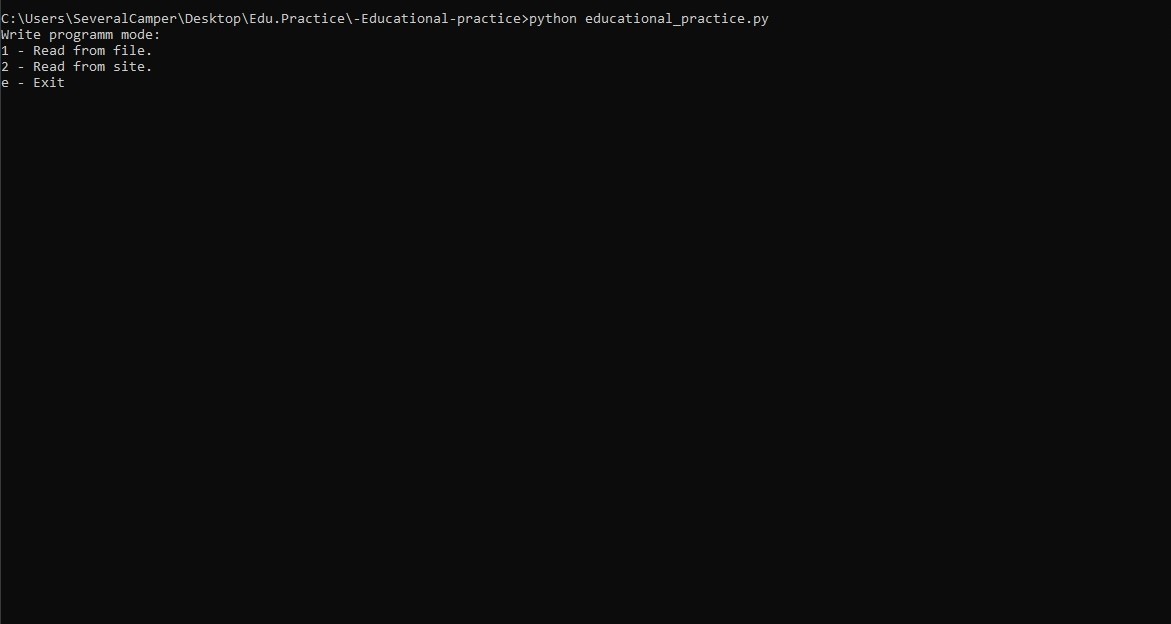
else:

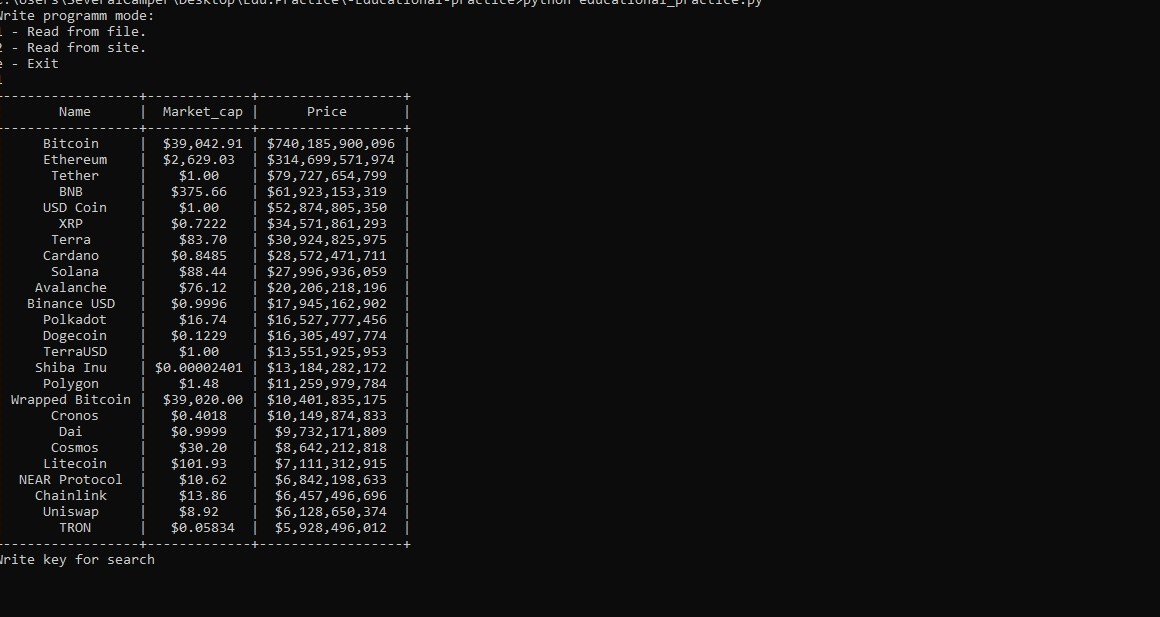
print("Unvalid input.")

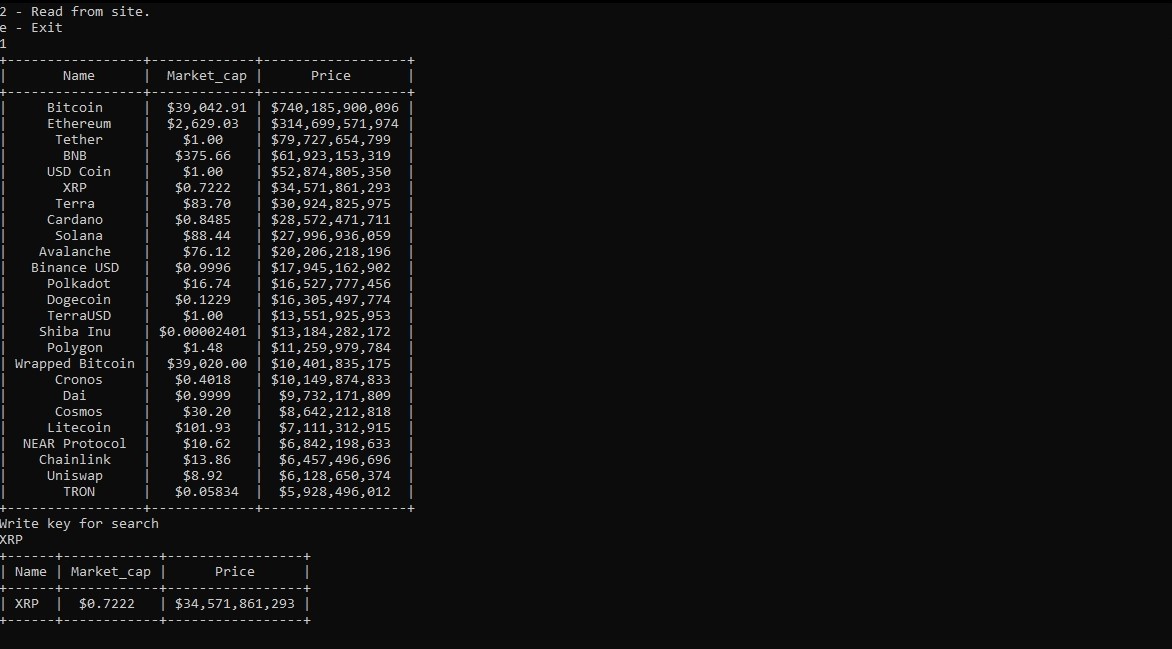
if \_\_name\_\_ == "\_\_main\_\_":

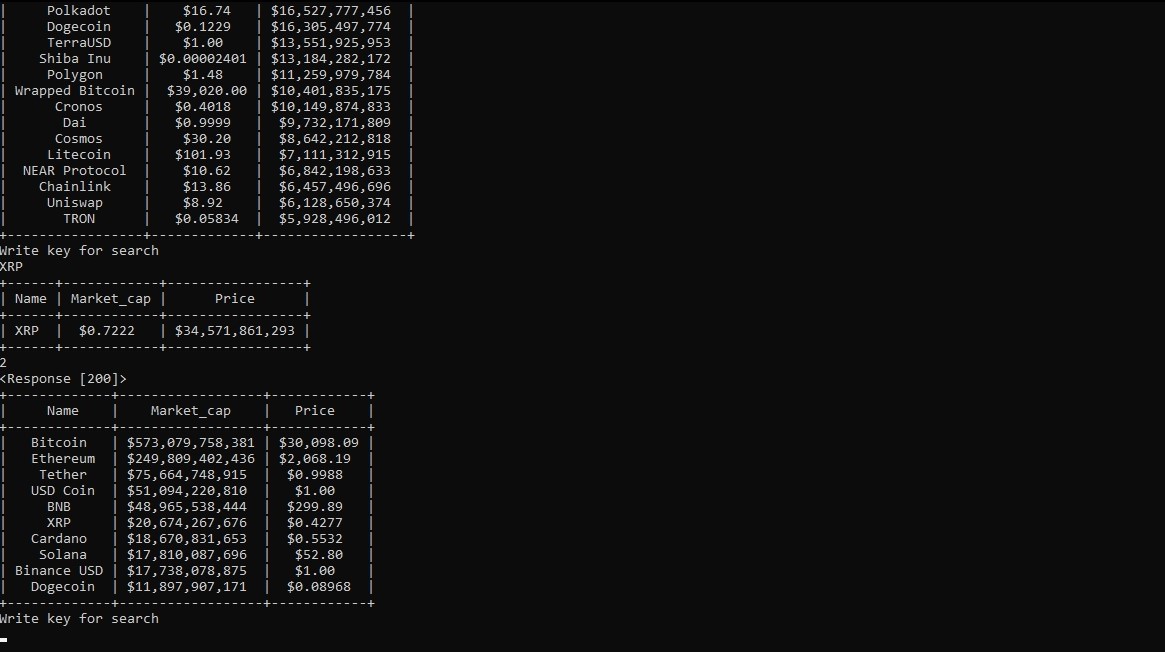
main()

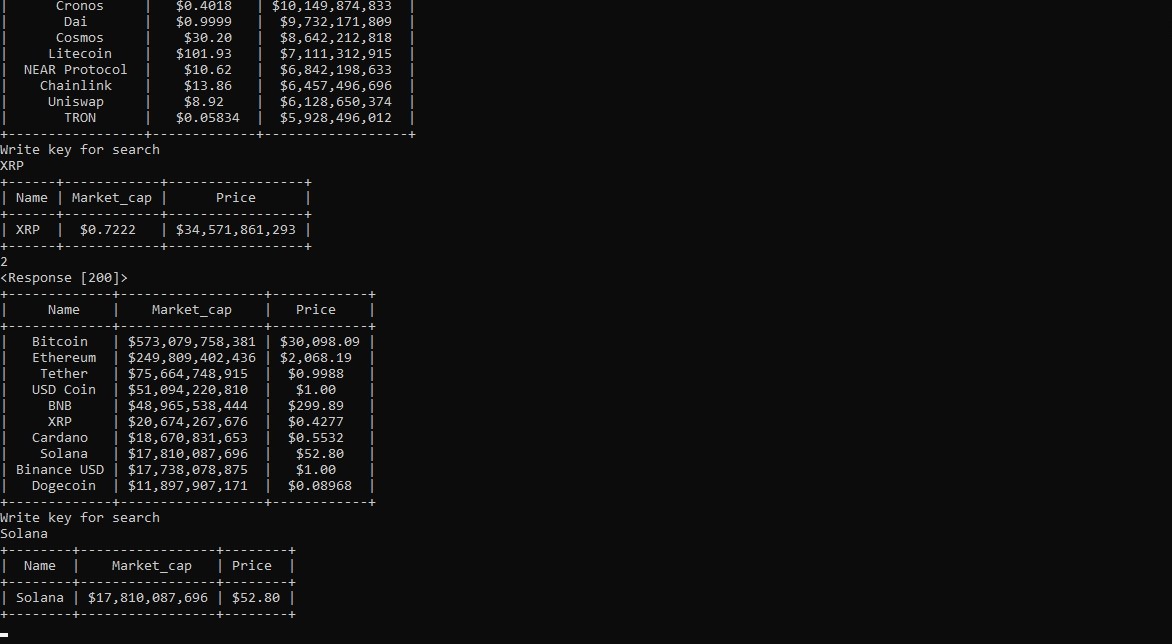
**Результаты тестирования**

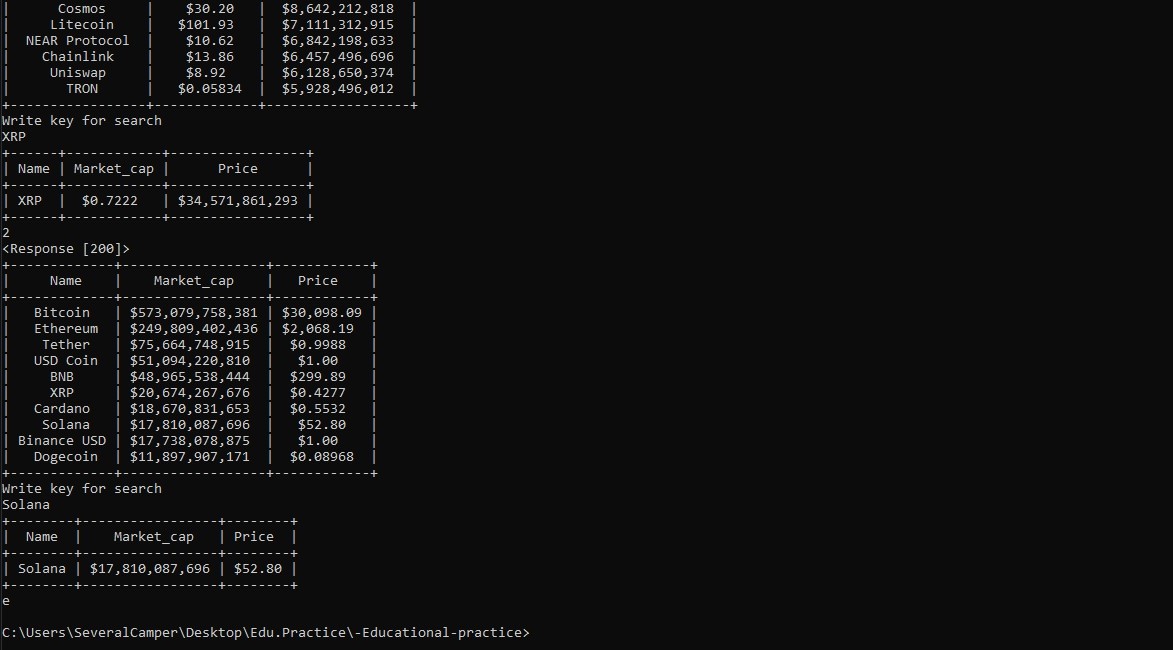












**Список использованных источников:**

https://docs.python.org/3/tutorial/index.html

https://[www.8host.com/blog/rabota-s-veb-](http://www.8host.com/blog/rabota-s-veb-) dannymi-s-pomoshhyu-requests-i-beautiful- soup-v-python-3/

https://[www.8host.com/blog/web-scraping-](http://www.8host.com/blog/web-scraping-) stranic-s-pomoshhyu-beautiful-soup-i-python- 3/

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://docs.python-requests.org/en/latest/

https://pythonru.com/biblioteki/kratkoe-rukovodstvo-po-biblioteke-python-requests

https://pythonworld.ru/moduli/modul-csv.html

Ссылка на github: https://github.com/SeveralCamper/Educational-practice