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-016 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мосолов П.А./

«11» мая 2022 г.

Руководитель практики

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

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2022г.

**Содержание**

1. Постановка задачи 3
2. Описание используемых алгоритмов 2
3. Листинг программы 6
4. Результаты тестирования 12
5. Список использованных источников 15
6. **Постановка задачи**

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)

1. 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.

1. **Описание используемых алгоритмов**

**def merge\_information():** данная функция объединяет три массива в “структуру”, для дальнейшего использования в выводе графики.

**def mass\_clear():** функция очистки массивов после использования.

**def Pulling\_information\_site():** данная функция запускает считывание данных с сайта. Внутри функции стоит проверка, возможно ли подключится к нему и запустить парсинг. Если нет, то вызываются исключения. Если все хорошо, функция стягивает разметку сайта, и запускает процедуру поиска интересующих нас данных.

**def find\_name\_crypto(crypto\_information\_list):** функция стягивания названий крипто валют с сайта. Алгоритм ищет по заданным параметрам, названия и записывает их в массив “names\_crypto”. Принимает **“crypto\_information\_list” –** разметка страницы для осуществления поиска.

**def find\_price\_crypto(crypto\_information\_list):** функция стягивания цены крипто валют с сайта. Алгоритм ищет по заданным параметрам, цену и записывает ее в массив “price\_crypto”. Принимает **“crypto\_information\_list” –** разметка страницы для осуществления поиска.

**def find\_market\_cap\_crypto(crypto\_information\_list):** функция стягивания рыночной капитализации крипто валют с сайта. Алгоритм ищет по заданным параметрам, рыночную капитализацию и записывает ее в массив “**market\_cap\_crypto\_long**”. Принимает **“crypto\_information\_list” –** разметка страницы для осуществления поиска.

**def terminal\_table\_output(length):** функция вывода данных в табличном виде, использует библиотеку **“PrettyTable”**.

**def searching\_element\_array():** функция поиска по заданному ключу крипто валюты. Выводит ошибку если по заданному ключу ничего не было найдено, если нашел совпадение, создает таблицу и выводит ее на экран (использует библиотеку **“PrettyTable”**)

**def parsing\_csv():** функция считывания данных из файла с форматом “**csv**”. Делает проверку, возможно ли открыть такой файл, если нет, выводит ошибку и предлагает пользователю решение проблемы. Если файл найден, то запускает чтение из файла и запись данных по масcивам.

**def create\_data():** функция вывода данных о крипто валюте в виде таблицы.

**def gui\_pars():** функция запуска графического интерфейса с использованием библиотеки **“ BeautifulSoup”**. Реализуется простой и понятный для пользователя интерфейс, в котором есть две функции, вывод крипто валют с сайта и из файла.

**def UI():** самая главная функция в данном практическом задании. Запускает интерфейс в терминале и предлагает пользователю выбрать в каком режиме работать, либо в терминале, либо в графическом режиме. Режим терминала делится на две составляющие. Первая – это вывод данных из файла, вторая – вывод данных с сайта. Везде реализован выход из данного режима.

1. **Листинг программы**

import requests

from bs4 import BeautifulSoup

from prettytable import PrettyTable

import PySimpleGUI as sg

import os

import csv

FILE\_NAME = "currencies22.csv"

SITE\_URL = "https://coinmarketcap.com"

names\_crypto = []

market\_cap\_crypto\_long = []

price\_crypto = []

struct\_crypto = []

def merge\_information():

for i in range(len(names\_crypto)):

struct = {

"name": names\_crypto[i],

"market\_cap": market\_cap\_crypto\_long[i],

"price": price\_crypto[i]

}

struct\_crypto.append(struct)

def mass\_clear():

names\_crypto.clear()

price\_crypto.clear()

market\_cap\_crypto\_long.clear()

struct\_crypto.clear()

def find\_name\_crypto(crypto\_information\_list):

crypto\_information\_list\_statistics = crypto\_information\_list.find\_all('p', class\_='sc-1eb5slv-0 iworPT')

for crypto\_information in crypto\_information\_list\_statistics:

names\_crypto.append(str(crypto\_information.contents[0].text))

def find\_market\_cap\_crypto(crypto\_information\_list):

crypto\_information\_list\_statistics = crypto\_information\_list.find\_all('span', class\_='sc-1ow4cwt-1 ieFnWP')

for crypto\_information in crypto\_information\_list\_statistics:

market\_cap\_crypto\_long.append(crypto\_information.contents[0].text)

def find\_price\_crypto(crypto\_information\_list):

crypto\_information\_list\_statistics = crypto\_information\_list.find\_all('div', class\_='sc-131di3y-0 cLgOOr')

for crypto\_information in crypto\_information\_list\_statistics:

price\_crypto.append(crypto\_information.contents[0].text)

def terminal\_table\_output(length):

Table\_crypto = PrettyTable()

Table\_crypto.field\_names = ["№","Name","Price","Marcet Cap"]

for i in range(length):

Table\_crypto.add\_row([i+1,names\_crypto[i], price\_crypto[i], market\_cap\_crypto\_long[i]])

print(Table\_crypto)

def searching\_element\_array():

Table\_crypto = PrettyTable()

Table\_crypto.field\_names = ["№","Name","Price","Marcet Cap"]

myLetter = str(input("Enter the search key: "))

i =0

for word in names\_crypto:

if myLetter.lower() in word.lower():

i = names\_crypto.index(word)

Table\_crypto.add\_row([i+1,names\_crypto[i], price\_crypto[i], market\_cap\_crypto\_long[i]])

if i == 0:

print("Eror, invalid key, try again\n")

else:

print(Table\_crypto)

def Pulling\_information\_site():

try:

page = requests.get(SITE\_URL)

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

crypto\_information\_list = soup.find('tbody')

find\_name\_crypto(crypto\_information\_list)

find\_market\_cap\_crypto(crypto\_information\_list)

find\_price\_crypto(crypto\_information\_list)

except(ConnectionError, Timeout) as exp:

print("Mistake! The reason for the error: ", exp)

sys.exit(1)

def parsing\_csv():

try:

with open(FILE\_NAME, encoding='utf-8') as file:

file\_reader = csv.reader(file, delimiter = ";")

count = 0

for row in file\_reader:

names\_crypto.append(f'{row[0]}')

price\_crypto.append(f'{row[1]}')

market\_cap\_crypto\_long.append(f'{row[2]}')

except FileNotFound:

print("Mistake! The file '{}' could not be opened. Check the correctness of the name, or the presence of it in the pope.".format(FILE\_NAME))

sys.exit(1)

def create\_data():

new\_struct\_crypto = []

for i in struct\_crypto:

new\_struct\_crypto.append([i["name"], i["market\_cap"], i["price"]])

return new\_struct\_crypto

def gui\_pars():

sg.theme("DarkPurple")

head = ["Name of the cryptocurrency", " Marcet Cap ", "Cost in dollars"]

frame\_1 = [

[sg.Text("Select the data output mode:")],

[sg.Button("1. FILE '{}'.".format(FILE\_NAME))],

[sg.Button("2. WEBSITE '{}'.".format(SITE\_URL))],

[sg.Text("Information:")],

[sg.Table(values=struct\_crypto, headings=head, key="TABLE", size=(500, 420))]]

layout = [[sg.Column(frame\_1)]]

sg.set\_options(font=("Courier New", 10))

window = sg.Window("Parser", layout, size=(690, 620))

gui = []

while True:

event, values = window.read()

if event == sg.WIN\_CLOSED:

break

if event == "1. FILE 'currencies22.csv'.":

mass\_clear()

parsing\_csv()

merge\_information()

gui = create\_data()

if event == "2. WEBSITE 'https://coinmarketcap.com'.":

mass\_clear()

Pulling\_information\_site()

merge\_information()

gui = create\_data()

print(struct\_crypto)

window["TABLE"].update(values=gui)

window.close()

def UI():

os.system('cls')

flag = 1

flag\_two = 1

while flag == 1:

mass\_clear()

mode = input("Select the output method ('T' from terminal, 'G' using graphics,'E' - end): ")

if mode == 'T':

flag\_two = 1

mass\_clear()

while flag\_two == 1:

mode2 = input("Choose the parsing method ('F' from file, 'S' from the site, 'Q' - end): ")

if mode2 == 'S':

mass\_clear()

os.system('cls')

Pulling\_information\_site()

terminal\_table\_output(len(names\_crypto))

searching\_element\_array()

elif mode2 == 'F':

mass\_clear()

os.system('cls')

parsing\_csv()

terminal\_table\_output(len(names\_crypto))

searching\_element\_array()

elif mode2 == 'Q':

flag\_two = 2

else:

os.system('cls')

print("Invalid argument")

elif mode == 'G':

mass\_clear()

os.system('cls')

gui\_pars()

elif mode == 'E':

flag = 2

else:

os.system('cls')

print("Invalid argument")

os.system('cls')

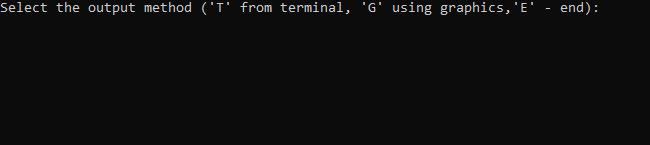
def main():

UI()

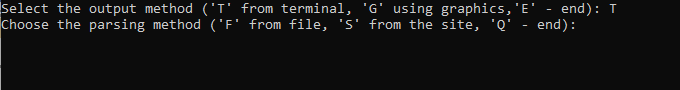
if \_\_name\_\_ == '\_\_main\_\_':

main()

1. **Результаты тестирования**



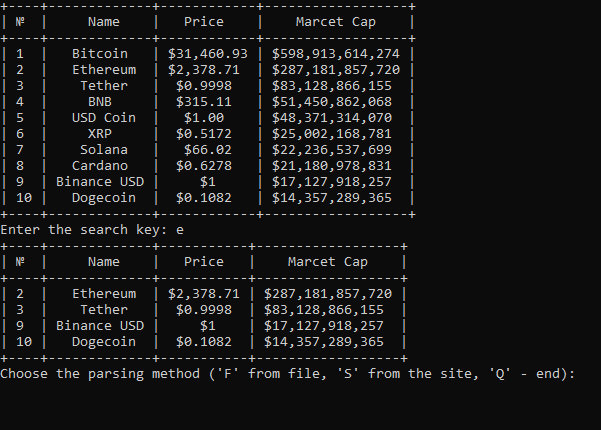
**Рисунок 1.** *Главное меню в консольном режиме*



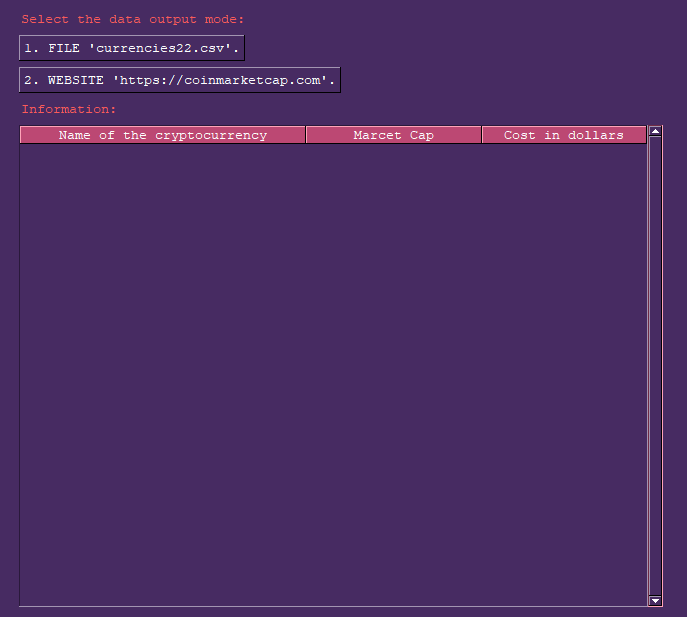
**Рисунок 2.** *Выбираем метод показа данных*

**

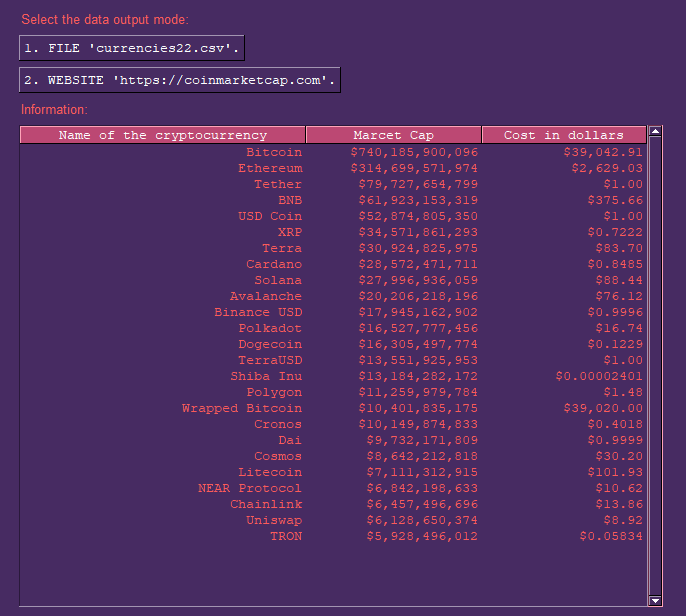
**Рисунок 3.** *Вывод из файла и поиск в файле по ключу*

**

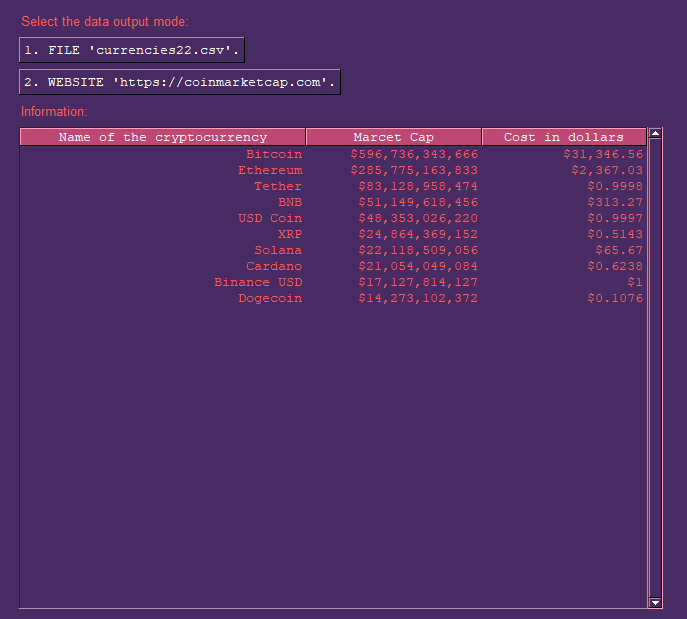
**Рисунок 4.** *Результат парсинга сайта и поиска по ключу*

**

**Рисунок 5.** *Графический режим работы программы*

**

**Рисунок 6.** *Результат вывода таблицы*

**

**Рисунок 7.** *Результат вывода с сайта*

1. **Список использованных источников**
2. Лутц М. Изучаем Python, том 1, 5-е издание / М. Лутц. – СПб : Диалектика, 2019. – 832 с. – ISBN 978-5-907144-52-1
3. CSV File Reading and Writing : сайт. – URL: https://docs.python.org/3/library/csv.html (дата обращения: 20.04.2022)
4. Guide to Parsing HTML with BeautifulSoup in Python : сайт. – URL: https://stackabuse.com/guide-to-parsing-html-with-beautifulsoup-in-python/ (дата обращения: 24.04.2022)
5. CoinMarketCap API Documentation : сайт. – URL: https://coinmarketcap.com/api/documentation/v1/ (дата обращения: 30.04.2022)
6. Python GUIs for Humans : сайт. – URL: https://pysimplegui.readthedocs.io/en/latest/ (дата обращения: 30.04.2022)