

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

\_\_\_\_\_/Егошин А.А./

«04» мая 2022 г.

Руководитель практики

доцент каф. ПМиК

\_\_\_\_\_/Приставка П.А./

«ДД» мм ГГГГ г.

Оценка \_\_\_\_\_

Новосибирск 2022г.

## Содержание

1. Постановка задачи .....	3
2. Описание используемых алгоритмов .....	4
3. Листинг программы .....	5
4. Результаты тестирования .....	10
5. Список использованных источников .....	13

## 1. Постановка задачи

Разработать программу реализующую ввод, хранение и обработку данных о котировках криптовалют на основе данных сайта [coinmarketcap.com](https://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 <code>currencies22.csv</code> . 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 сайта <a href="https://coinmarketcap.com">coinmarketcap.com</a> в момент запуска программы. Загрузка и парсинг веб-страницы производится с помощью библиотек <code>Requests</code> и <code>Beautifulsoup</code> или их аналогов  Примечание: допускается считывание строчек в количестве менее 25 (Например, 10 строчек с данными о криптовалютах)

## 5. Хранение

Типы и структуры для хранения данных: определяются студентом

## 6. Обработка

Реализовать функцию поиска информации о свойствах криптовалюты по ее названию.

## 2. Описание используемых алгоритмов

**def console():** одна из главных функций программы. Осуществляет управление всей программой в консольном режиме работы, вызывая необходимые функции.

**def gui():** одна из главных функций программы. Осуществляет управление всей программой в графическом режиме работы, вызывая необходимые функции и отображая элементы управления и данных.

**def parse\_file():** функция считывания данных из файла. Проверяет наличие файла *currencies22.csv* в директории и осуществляет его парсинг. В случае если данного файла нет, то вызывает исключение *FileNotFoundError*, выводит сообщение об ошибке на экран и завершает работу. Функция **возвращает** список словарей с ключами: *"name", "market\_cap", "price"*.

**def parse\_website():** функция считывания данных с сайта. Проверяет возможность подключения к нему и осуществляет парсинг. В случае если к сайту невозможно подключиться, то вызывает одно из исключений *ConnectionError, Timeout, TooManyRedirects*, выводит сообщение об ошибке на экран и завершает работу. Функция **возвращает** список словарей с ключами: *"name", "market\_cap", "price"*.

**def parse\_api():** функция получения данных с сайта с использованием API. Проверяет возможность подключения к сайту и отправляет на сервер запрос с просьбой вернуть в ответе данные о криптовалюте, после чего производит парсинг возвращенного *json* текста. В случае если к сайту невозможно подключиться, то вызывает одно из исключений *ConnectionError, Timeout, TooManyRedirects*, выводит сообщение об ошибке на экран и завершает работу. Функция **возвращает** список словарей с ключами: *"name", "market\_cap", "price"*.

**def find(data, key):** функция поиска криптовалюты по её названию. Поиск нечувствителен к регистру и для своей работы не требует полный ключ поиска. **Принимает:** *data* – список словарей, *key* – ключ поиска. **Возвращает** список словарей, соответствующих ключу поиска.

**def print\_data(data):** функция вывода данных о криптовалюте в виде таблицы. **Принимает:** *data* – список словарей.

**def create\_data\_for\_gui\_table(data):** функция, которая изменяет формат хранения данных о криптовалюте для вывода с помощью графического интерфейса. **Принимает:** *data* – список словарей. **Возвращает:** список тех же данных, но с изменённой структурой.

### 3. Листинг программы

```
import csv
import requests
import sys
import os
import json
from requests.exceptions import ConnectionError, Timeout, TooManyRedirects
from bs4 import BeautifulSoup
from prettytable import PrettyTable
import PySimpleGUI as sg

FILE_NAME = "currencies22.csv"
WEBSITE = "https://coinmarketcap.com"

def parse_api():
    parameters = {
        "start": "1",
        "limit": "25",
        "convert": 'USD'
    }

    headers = {
        "Accepts": "application/json",
        "X-CMC_PRO_API_KEY": "41aad901-cad5-4596-854f-a69a6833b3c3"
    }

    data = []

    try:
        response = requests.get(url="https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest", headers=headers,
                                params=parameters)

        file = json.loads(response.text)

        for element in file["data"]:
            item = {
                "name": element["name"],
                "market_cap": element["quote"]["USD"]["market_cap"],
                "price": element["quote"]["USD"]["price"]
            }

            data.append(item)

    except (ConnectionError, Timeout, TooManyRedirects) as exp:
        print("Ошибка!", exp)
        sys.exit(1)

    return data

def parse_website():
    data = []

    headers = {
        'Accept':
        "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
        image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
    }
```

```

        'User-Agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.132
YaBrowser/22.3.1.892 Yowser/2.5 Safari/537.36"
    }

    try:
        response = requests.get(url=WEBSITE, headers=headers)

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

        table = soup.find("tbody")

        names = table.find_all("p", class_="sc-1eb5slv-0 iworPT")
        market_caps = table.find_all("span", class_="sc-1ow4cwt-1 ieFnWP")
        prices = table.find_all("div", class_="sc-131di3y-0 cLgOOr")

        for i in range(len(names)):
            item = {
                "name": names[i].text,
                "market_cap": market_caps[i].text,
                "price": prices[i].text
            }

            data.append(item)

    except (ConnectionError, Timeout, TooManyRedirects) as exp:
        print("Ошибка!", exp)
        sys.exit(1)

    return data

def parse_file():
    data = []

    try:
        file = open(FILE_NAME, 'r')
        table = csv.reader(file, delimiter=';')

        for element in table:
            item = {
                "name": element[0],
                "market_cap": element[1],
                "price": element[2]
            }

            data.append(item)

        file.close()

    except FileNotFoundError:
        print("Ошибка! Не удалось открыть файл '{}'.format(FILE_NAME))
        sys.exit(1)

    return data

def find(data, key):
    find_items = []

```

```

    for item in data:
        if item.get("name").lower().startswith(key.lower()):
            find_items.append(item)

    return find_items

def print_data(data):
    table = PrettyTable()

    table.field_names = ["Наименование", "Рыночная капитализация", "Стоимость
1 ед. в долларах"]

    for element in data:
        table.add_row([element["name"], element["market_cap"],
element["price"]])

    print(table)

def create_data_for_gui_table(data):
    new_data = []

    for i in data:
        new_data.append([i["name"], i["market_cap"], i["price"]])

    return new_data

def gui():
    data = []

    frame_1 = [
        [sg.Text("Выберите режим считывания данных: ")],
        [sg.Button("1. Файл '{}'.format(FILE_NAME) )],
        [sg.Button("2. Веб-сайт '{}'.format(WEBSITE) )],
        [sg.Button("3. С использованием API. ")],
        [sg.Text("Поиск: ")],
        [sg.Input("Введите название криптовалюты"), sg.Button("Найти",
size=(4,1))]]
    ]

    headings = ["Наименование", "Рыночная капитализация", "Стоимость 1 ед. в
долларах"]

    frame_2 = [
        [sg.Text("Информация о криптовалюте: ")],
        [sg.Table(values=data, headings=headings, key="TABLE", size=(400,
420))]]
    ]

    layout = [
        [sg.Column(frame_1), sg.Column(frame_2)]
    ]

    window = sg.Window("CoinMarketCap парсер", layout, size=(1280, 520))

    while True:
        event, values = window.read()

        if event == sg.WIN_CLOSED:

```

```

        break
    if event == "1. Файл 'currencies22.csv'.":
        data = parse_file()
        gui_data = create_data_for_gui_table(data)
    if event == "2. Веб-сайт 'https://coinmarketcap.com'.":
        data = parse_website()
        gui_data = create_data_for_gui_table(data)
    if event == "3. С использованием API.":
        data = parse_api()
        gui_data = create_data_for_gui_table(data)
    if event == "Найти":
        data = find(data, values[0])
        gui_data = create_data_for_gui_table(data)

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

window.close()

def console():
    data = []

    print("Выберите режим считывания данных:")
    print("1. Файл '{}'.format(FILE_NAME))
    print("2. Веб-сайт '{}'.format(WEBSITE))
    print("3. С использованием API.")

    while True:
        choice = input("> ")

        if choice not in ['1', '2', '3']:
            print("Ошибка! Введено некорректное значение.")
            print("Пожалуйста, попробуйте снова.")
        else:
            break

    os.system("cls")

    if choice == '1':
        data = parse_file()
    elif choice == '2':
        data = parse_website()
    else:
        data = parse_api()

    print_data(data)

    print("1. Поиск криптовалюты по названию.")
    print("2. Выход из приложения.")

    while True:
        choice = input("> ")

        if choice not in ['1', '2']:
            print("Ошибка! Введено некорректное значение.")
            print("Пожалуйста, попробуйте снова.")
        else:
            break

```



```

if choice == '1':
    print("Введите ключ поиска")
    key = input("> ")

    find_data = find(data, key)

    if len(find_data) == 0:
        print("По Вашему запросу ничего не найдено.")
    else:
        print("По Вашему запросу найдено:")
        print_data(find_data)
else:
    sys.exit(1)

if __name__ == "__main__":
    if len(sys.argv) == 1:
        print("coinmarketcap парсер - выполненное задание на учебную практику  
СибГУТИ 2022")
        print("Работу выполнил: студент 2 курса ИП-016 Егошин А.А.")
        print("Запуск: python main.py --mode (-m) console / gui")

    else:
        if len(sys.argv) < 3:
            print("Ошибка! Параметров, необходимых для запуска мало.")
            print("Возможно, Вы не указали режим в котором хотите запустить  
программу.")
            print("Пожалуйста, проверьте и попробуйте снова.")
            sys.exit(1)

        if len(sys.argv) > 3:
            print("Ошибка! Параметров, необходимых для запуска много.")
            print("Пожалуйста, удалите лишние и попробуйте снова.")
            sys.exit(1)

        parameter_name = sys.argv[1]
        parameter_value = sys.argv[2]

        if (parameter_name == "--mode" or parameter_name == "-m"):
            if (parameter_value == "console"):
                console()

            elif (parameter_value == "gui"):
                gui()
            else:
                print("Ошибка! Неверное значение параметра mode.")
                print("Пожалуйста, проверьте правильность выбора режима и  
попробуйте снова.")
                sys.exit(1)
        else:
            print("Ошибка! Неизвестный параметр {}".format(parameter_value))
            print("Пожалуйста, проверьте корректность ввода и попробуйте  
снова.")
            sys.exit(1)

```

## 4. Результаты тестирования

```
C:\Users\alexe\Desktop\coinmarketcap-parser\CoinMarketCap-parser>python main.py -m console
Выберите режим считывания данных:
1. Файл 'currencies22.csv'.
2. Веб-сайт 'https://coinmarketcap.com'.
3. С использованием API.
> -
```

Рисунок 1. Главное меню в консольном режиме работы

Наименование	Рыночная капитализация	Стоимость 1 ед. в долларах
Bitcoin	\$39,042.91	\$740,185,900,096
Ethereum	\$2,629.03	\$314,699,571,974
Tether	\$1.00	\$79,727,654,799
BNB	\$375.66	\$61,923,153,319
USD Coin	\$1.00	\$52,874,805,350
XRP	\$0.7222	\$34,571,861,293
Terra	\$83.70	\$30,924,825,975
Cardano	\$0.8485	\$28,572,471,711
Solana	\$88.44	\$27,996,936,059
Avalanche	\$76.12	\$20,206,218,196
Binance USD	\$0.9996	\$17,945,162,902
Polkadot	\$16.74	\$16,527,777,456
Dogecoin	\$0.1229	\$16,305,497,774
TerraUSD	\$1.00	\$13,551,925,953
Shiba Inu	\$0.00002401	\$13,184,282,172
Polygon	\$1.48	\$11,259,979,784
Wrapped Bitcoin	\$39,020.00	\$10,401,835,175
Cronos	\$0.4018	\$10,149,874,833
Dai	\$0.9999	\$9,732,171,809
Cosmos	\$30.20	\$8,642,212,818
Litecoin	\$101.93	\$7,111,312,915
NEAR Protocol	\$10.62	\$6,842,198,633
Chainlink	\$13.86	\$6,457,496,696
Uniswap	\$8.92	\$6,128,650,374
TRON	\$0.05834	\$5,928,496,012

1. Поиск криптовалюты по названию.  
2. Выход из приложения.  
> -

Рисунок 2. Результаты парсинга данных из файла

Введите ключ поиска  
> LITE  
По Вашему запросу найдено:

Наименование	Рыночная капитализация	Стоимость 1 ед. в долларах
Litecoin	\$101.93	\$7,111,312,915

Рисунок 3. Работа поиска

Наименование	Рыночная капитализация	Стоимость 1 ед. в долларах
Bitcoin	\$723,362,768,940	\$38,017.51
Ethereum	\$334,341,557,976	\$2,772.37
Tether	\$83,172,268,450	\$1.00
BNB	\$62,295,066,087	\$381.53
USD Coin	\$49,273,567,029	\$1.00
Solana	\$29,329,681,305	\$87.81
XRP	\$28,433,714,251	\$0.5911
Terra	\$27,341,003,856	\$79.18
Cardano	\$25,786,229,577	\$0.764
TerraUSD	\$18,468,729,842	\$0.9986

1. Поиск криптовалюты по названию.  
2. Выход из приложения.  
> ☐

Рисунок 4. Результат парсинга сайта



Рисунок 5. Графический режим работы программы

Наименование	Рыночная капитализация	Стоимость 1 ед. в долларах
Bitcoin	723275335973.521	38012.91852483829
Ethereum	334414067137.6703	2772.9683435202405
Tether	83167389143.81004	1.0001745223523906
BNB	62259830390.2388	381.31420876290156
USD Coin	49255485080.87298	1.0000471765772576
Solana	29282020057.246143	87.6684990648782
XRP	28434642665.7989	0.5910924820355179
Terra	27320486650.661	79.1200110986753
Cardano	25829190725.3024	0.76525119411663
TerraUSD	18482504379.34025	0.999351478704618
Binance USD	17667908037.08761	0.998792383361779
Dogecoin	17245070088.172047	0.12998395071570273
Avalanche	15648015549.076122	58.254706622895654
Polkadot	14647765492.355812	14.83198895573499
Shiba Inu	11201747462.057983	2.0401560062408793e-05
Wrapped Bitcoin	10697722038.196096	37947.829770870565
Cronos	8909445060.616154	0.3526675468456257
Dai	8810788780.503996	0.9999354769624967
Polygon	8321223542.715802	1.0601815525957166
NEAR Protocol	7003779028.12667	10.379593453837758
Litecoin	6754130433.691254	96.22931357358028
TRON	6412789479.069828	0.06311000742043607
UNUS SED LEO	5387480555.797385	5.647525794345463
Bitcoin Cash	5301307233.998581	278.2675937690558
FTX Token	5241159447.234479	38.28498715503158

Рисунок 6. Результат работы API в графическом режиме работы программы



**Рисунок 7.** *Результат работы поиска в графическом режиме работы программы*

## **5. Список использованных источников**

1. Лутц М. Изучаем Python, том 1, 5-е издание / М. Лутц. – СПб : Диалектика, 2019. – 832 с. – ISBN 978-5-907144-52-1
2. CSV File Reading and Writing : сайт. – URL: <https://docs.python.org/3/library/csv.html> (дата обращения: 20.04.2022)
3. 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)
4. CoinMarketCap API Documentation : сайт. – URL: <https://coinmarketcap.com/api/documentation/v1/> (дата обращения: 30.04.2022)
5. Python GUIs for Humans : сайт. – URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 30.04.2022)