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
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 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.

1. **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 некоректное 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 некоректное 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)

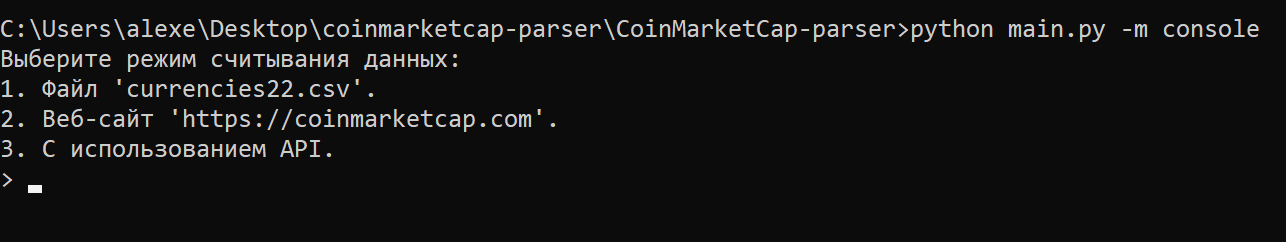
else:

print("Ошибка! Неизвестный параметр {}".format(parameter\_value))

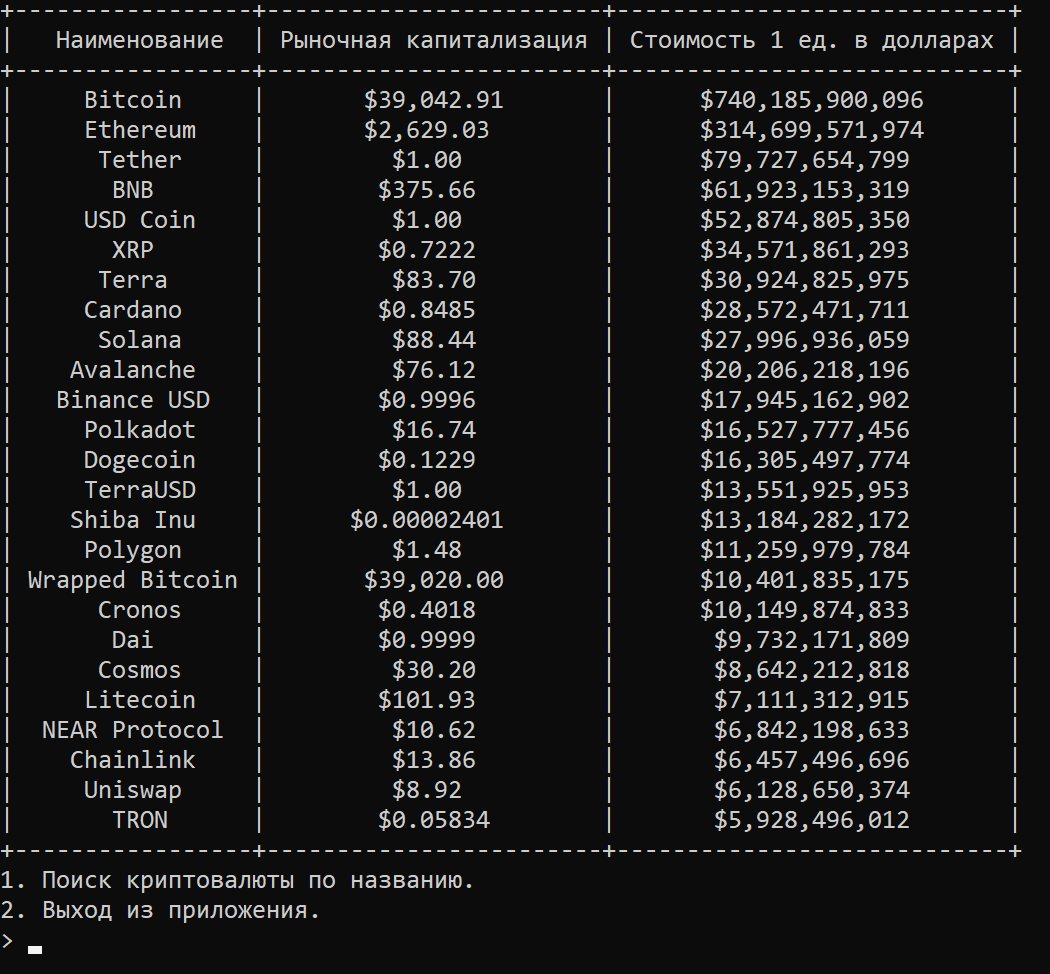
print("Пожалуйста, проверьте корректность ввода и попробуйте снова.")

sys.exit(1)

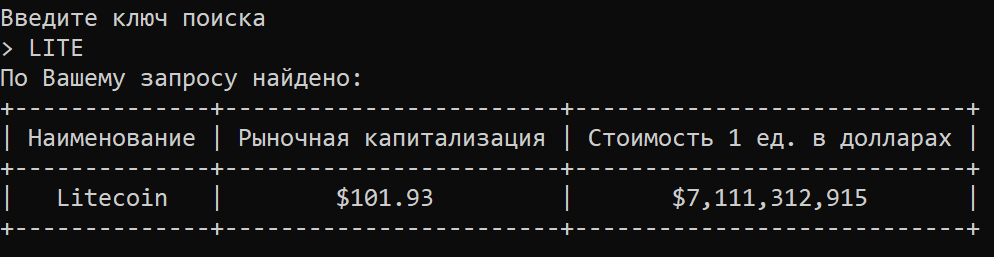
1. **Результаты тестирования**



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



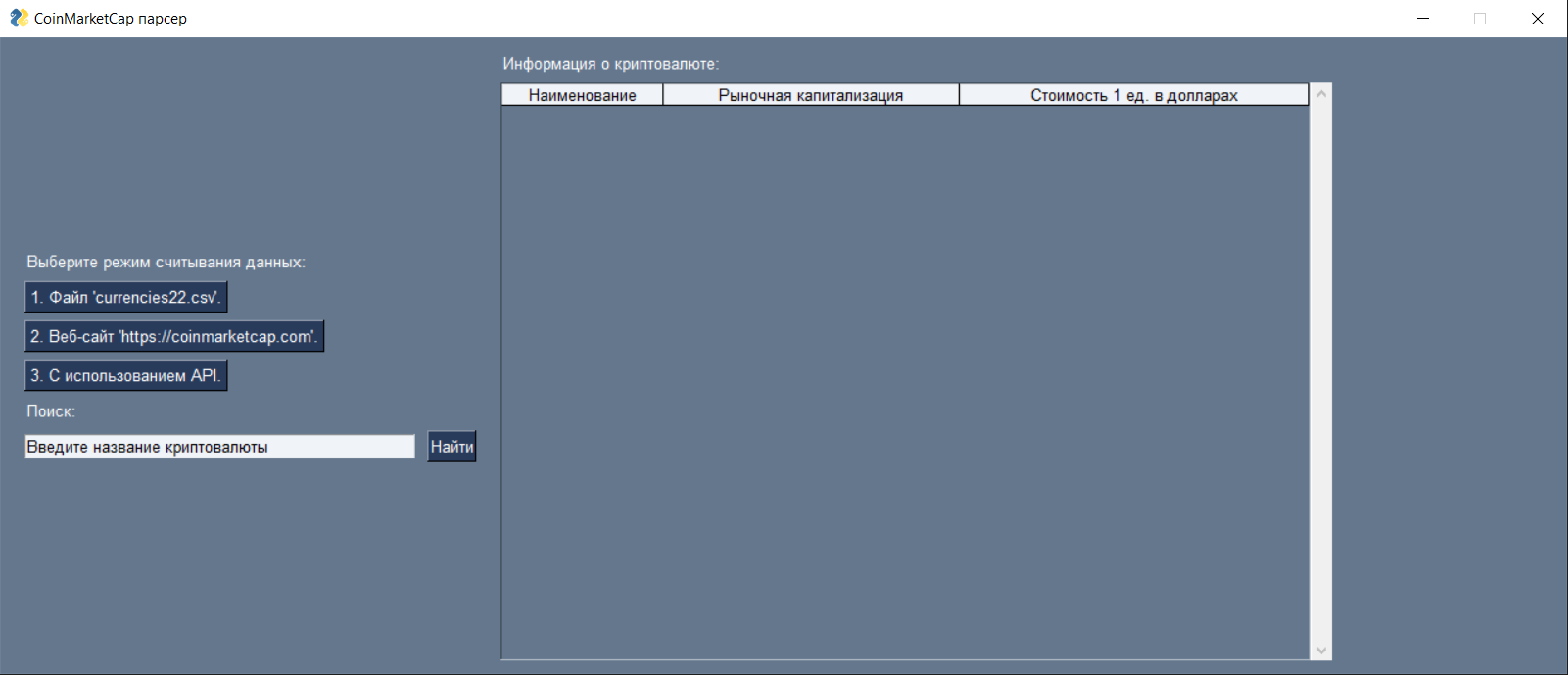
**Рисунок 2.** *Результаты парсинга данных из файла*

**

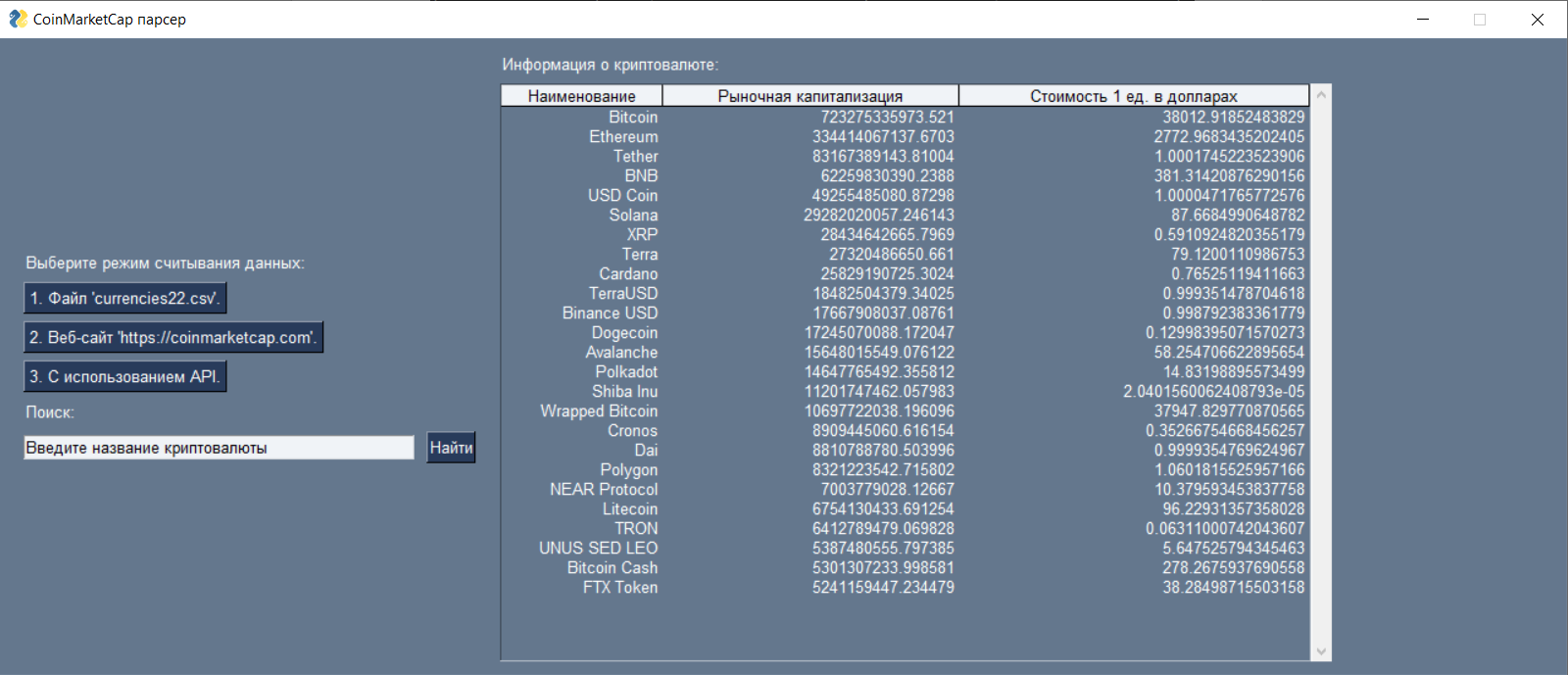
**Рисунок 3.** *Работа поиска*

**

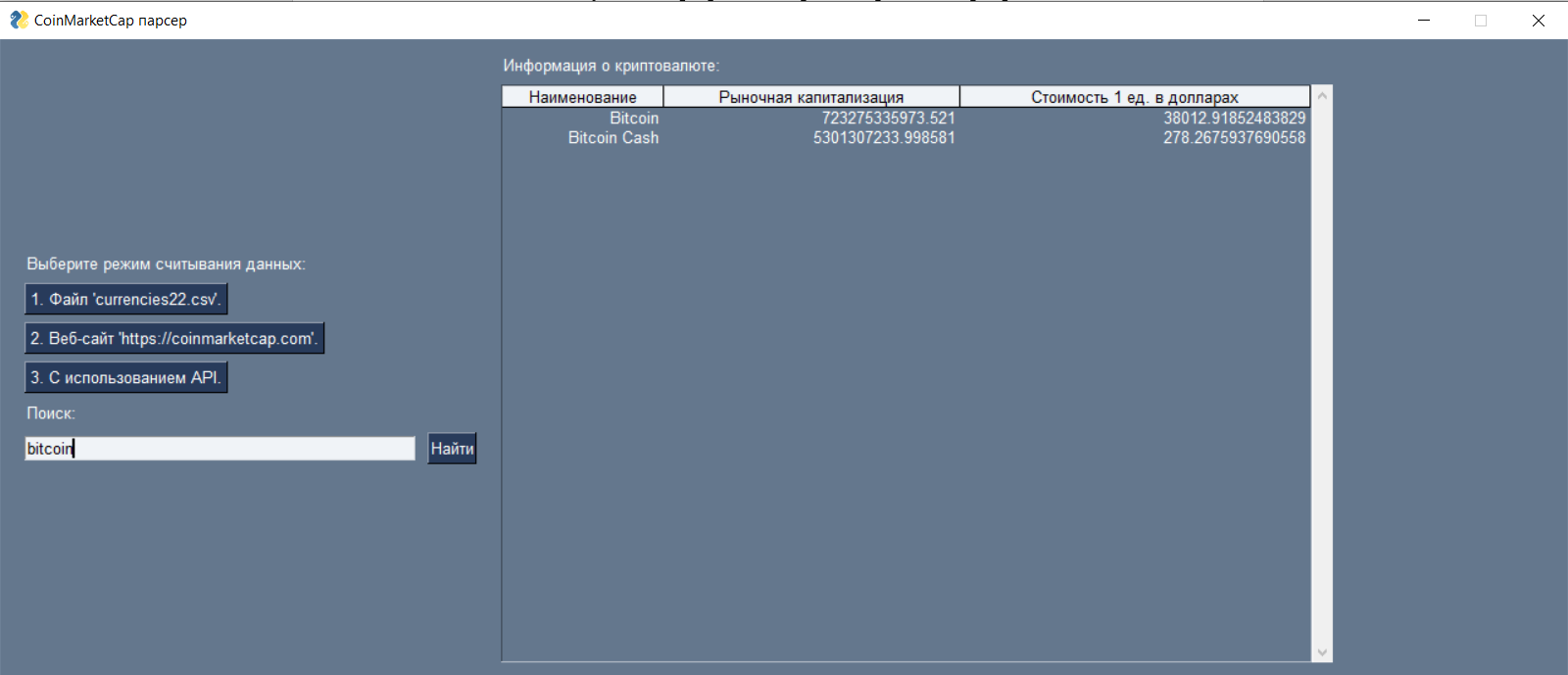
**Рисунок 4.** *Результат парсинга сайта*

**

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

**

**Рисунок 6.** *Результат работы API в графическом режиме работы программы*

**

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

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)