

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

_____/Солодкин М.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.

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 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)

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

- **Class Coin:** Класс Монеты, в конструктор передается ссылка на страницу монеты и конструктор парсит данные со страницы, складывая их в атрибуты класса.
- **def getCoins():** Функция вытягивает ссылки на 100 монет с главной страницы и записывает в массив объекты Coin. После их создания, массив сортируется по имени.

3. Листинг программы

```
import requests
from bs4 import BeautifulSoup
import PySimpleGUI as sg
import threading

headings = ['Coin', 'Price', 'MarketCap']
values = []

col1=[[sg.Button('Download')],[sg.Text('Enter Coin
Name:'),sg.InputText(key='-INPUT-')],[sg.Button('Find')]]
col2=[[sg.Table(headings=headings,size=(50,20) ,auto_size_columns=False,
key='-TABLE-',col_widths=[30,10,20] ,values=values, justification='left')]]
layout = [[sg.Column(col1,size=(200,100)), sg.Column(col2, size=(600,400))]]

window = sg.Window('Coins', layout, finalize=True)

url = "https://coinmarketcap.com/"
req = requests.get(url)
soup = BeautifulSoup(req.text, 'html.parser')
l=soup.find('table', class_='h7vnx2-2 czTsgW cmc-table').tbody
l=l.findAll('tr')

coins = []
ncoins=[]

stop=False

class Coin:
    def __init__(self, url):
        req = requests.get(url)
        soup = BeautifulSoup(req.text, 'html.parser')
        text = soup.find('h2', class_='sc-1q9q90x-0 jCInrl h1')
        self.name = text.text[:-len(text.findNext().text)]
        self.price = soup.find('div', class_='priceValue').text
        self.marketCap = soup.find('div', class_='statsValue').text

    def __str__(self):
        return self.name + " " + self.price + ", marketCap: " +
self.marketCap

def getCoins():
    global l,coins,ncoins
    for i in l:
        if stop: return
        x = i.find('a', class_='cmc-link')
        coin = Coin(url[:-1] + x['href'])
        coins.append(coin)
        print(str(coin))
        ncoins.append([coin.name, coin.price, coin.marketCap])

    coins.sort(key=lambda x: x.name)
    ncoins.sort(key=lambda x: x[0])

while True:
    event, values = window.read(timeout=1)
    if event == 'Exit' or event == sg.WIN_CLOSED:
        stop=True
        break
    elif event=='Download':
        threading.Thread(target=getCoins).start()
```

```
elif event=='Find':
    val = values['-INPUT-']
    val= val.lower()
    ncoins=[]
    for i in coins:
        if i.name.lower().find(val)>=0:
            ncoins.append([i.name, i.price, i.marketCap])
    window['-TABLE-'].update(ncoins)

window.close()
```

4. Результаты тестирования

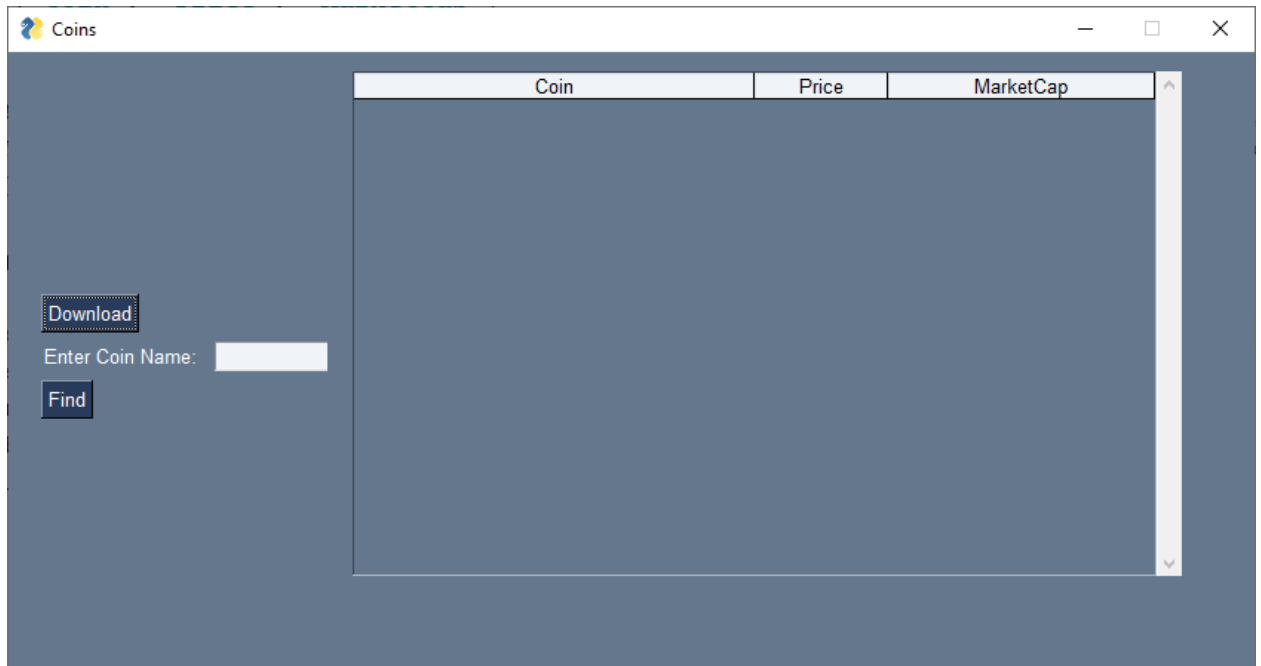


Рисунок 1. Программа при запуске

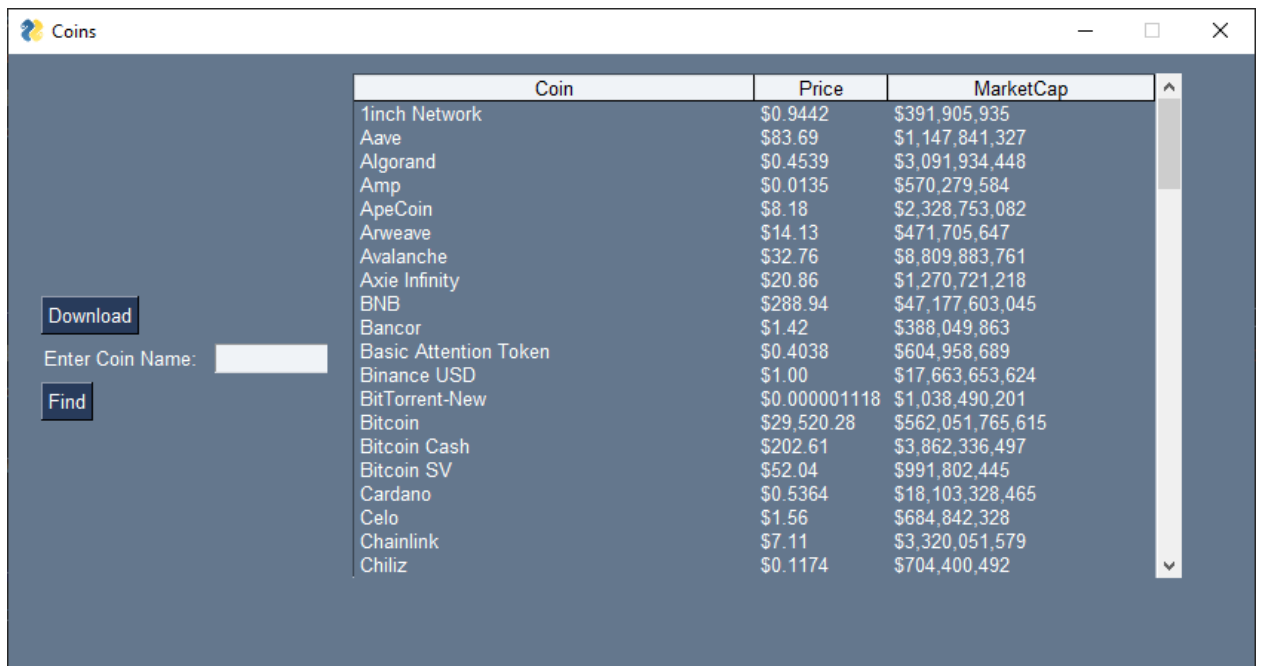


Рисунок 2. Результаты парсинга данных

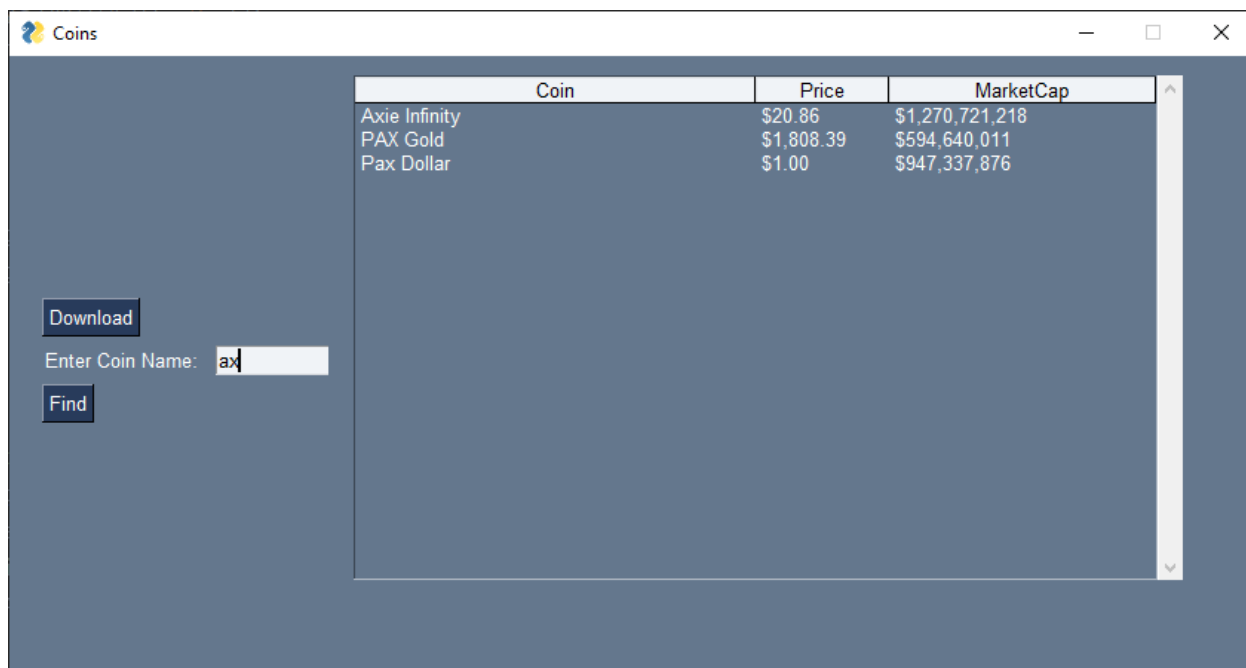


Рисунок 3. *Работа поиска*

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. Guide to Parsing HTML with BeautifulSoup in Python : сайт. – URL: <https://stackabuse.com/guide-to-parsing-html-with-beautifulsoup-in-python/> (дата обращения: 09.05.2022)
3. Python GUIs for Humans : сайт. – URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 9.05.2022)