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-115 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Панк А.С./

ФИО студента

«14» мая 2023г.

/Артеменко Т.С./

ФИО студента

Руководитель практики

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

«15» мая 2023г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2023 г.

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ 3

2 ОПИСАНИЕ МЕТОДОВ 4

2.1 Ввод данных 4

2.2 Обработка данных 4

2.3 Хранение данных 4

2.4 Графический интерфейс 4

3 ЛИСТИНГ ПРОГРАММЫ 6

4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ 9

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ…………………………. 12

1 ПОСТАНОВКА ЗАДАЧИ

Разработать программу реализующую простейшее меню, 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

• Symbol

• Price – стоимость 1 ед. в долларах США (USD)

• Market\_cap – рыночная капитализация

4. Ввод данных: 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.

5. Хранение Типы и структуры для хранения данных: определяются студентом.

6. Обработка: реализовать функцию поиска информации о свойствах криптовалюты по ее названию.

7. Графический интерфейс: реализовать оконное приложение с помощью методов библиотеки Tkinter.

Студентом Панк А.С. выполнено: парсинг с сайта, совмещение всех функций с графическим интерфейсом.

Студентом Артеменко Т.С. выполнено: парсинг с сайта, функция поиска, реализация сохранения в файл.

2 ОПИСАНИЕ МЕТОДОВ

2.1 Парсинг

Парсинг страницы реализован в функции def parsing()

browser = webdriver.Chrome()

browser.get("https://coinmarketcap.com/")

Сбор информации реализуется с помощью библиотеки selenium. Запускается браузер Chrome и в нем запускается указанная ссылка.

browser.execute\_script("window.scrollTo(0, %d);" % (800 \* i))

С помощью данной строки кода прогружает страницу путем так называемого “Scroll”.

browser.set\_page\_load\_timeout(15)

Далее ожидание обновление данных сайта.

table = browser.find\_element(By.XPATH, '//tbody/tr[%d]' % (number + 1))

С помощью поиска по коду сайта можно найти относительный путь до элементов, которые нам потребуются для обработки данных. В примере приведена функция как извлекается таблица из кода главной страницы.

name = table.find\_element(By.XPATH, "td[3]//p")

В данном отрывке кода заносится в переменную name значения названия криптовалюты и т.д. для остальных параметров.

data\_one\_coin.append(number + 1)

data\_all\_coins.append(data\_one\_coin)

Далее заносятся данные в структуру для одного элемента, после этого вся полученная информация об одном элементе помещается в структуру для всех элементов.

2.2 Обработка данных

Функция поиска реализована в def btnfunc()

str = pole.get()

Считывается информация с области ввода в Tkinter и далее путем прохождения по всем элементам находится заданная криптовалюта. После этого заносится в таблицу.

2.3 Хранение данных

Для хранения данных использовалась библиотека openpyxl. Данная возможность реализована в функции def btn3f(). В папке находится файл Save.xlsx. В него заносятся данные из переменной data.

2.4 Графический интерфейс

Графический интерфейс реализован с помощью библиотеки Tkinter. Главное меню программы реализованы три кнопки, таблица и строка с последним обновлением данныйх.

К каждой кнопке прикреплена функция, например, к первой «Update» реализована программа сбора данных с сайта, к «Save» функция сохранения данных в EXEL. Далее кнопка «Search», при нажатии открывается еще оно оконное приложение. В данном окне использованы таблица, кнопки и поле ввода. При введении названии криптовалюты, которую требуется найти, далее при нажатии кнопки «Search2 осуществляется поиск, так же можно очистить таблицу с помощью кнопки «2CLEAR2.

3 ЛИСТИНГ ПРОГРАММЫ

import datetime

import pickle

import os

from selenium import webdriver

from selenium.webdriver.common.by import By

from tkinter import \*

from tkinter import ttk

from functools import partial

import openpyxl

def parsing(data\_all\_coins):

    browser = webdriver.Chrome()

    browser.get("https://coinmarketcap.com/")

    for i in range(1, 12):

        browser.execute\_script("window.scrollTo(0, %d);" % (800 \* i))

    browser.set\_page\_load\_timeout(15)

    for number in range(50):

        table = browser.find\_element(By.XPATH, '//tbody/tr[%d]' % (number + 1))

        name = table.find\_element(By.XPATH, "td[3]//p")

        symbol = table.find\_element(By.XPATH, "td[3]/div/a/div/div/div/p")

        price = table.find\_element(By.XPATH, "td[4]")

        market\_cap = table.find\_element(By.XPATH, "td[8]")

        data\_one\_coin = []

        data\_one\_coin.append(number + 1)

        data\_one\_coin.append(name.text)

        data\_one\_coin.append(symbol.text)

        data\_one\_coin.append(price.text)

        data\_one\_coin.append(market\_cap.text)

        data\_all\_coins.append(data\_one\_coin)

    browser.close()

    return data\_all\_coins

def btn1f():

    table.delete(\*table.get\_children())

    data\_all\_coins = []

    data\_all\_coins = parsing(data\_all\_coins=data\_all\_coins)

    data = [(data\_all\_coins[i][0],data\_all\_coins[i][1],

             data\_all\_coins[i][2],data\_all\_coins[i][3],

             data\_all\_coins[i][4]) for i in range(50)]

    for inf in data :

        table.insert("",END,values=inf)

    today = datetime.datetime.today()

    lbl.config(text='Last update: '+today.strftime("%Y-%m-%d %H.%M.%S"))

    return data

def btn2f(data\_all\_coins):

    def DELETE():

        table.delete(\*table.get\_children())

    def btnfunc(data\_all\_coins=data\_all\_coins):

        str = pole.get()

        for i in range(len(data\_all\_coins)):

            if data\_all\_coins[i][1]==str:

                z = [data\_all\_coins[i][1],data\_all\_coins[i][2],data\_all\_coins[i][3],data\_all\_coins[i][4]]

                table.insert("",END,values=z)

                break

    search = Tk()

    search.title('Searching')

    search.resizable(width=0, height=0)

    lbl = Label(search, text="Введите название криптовалюты:")

    btn = Button(search, text="Search", command=btnfunc)

    btn1 = Button(search, text="CLEAR", command=DELETE)

    pole = Entry(search)

    columns = ('nam','symb','pric','market\_kup')

    table = ttk.Treeview(search, columns=columns,show='headings')

    table.heading('nam', text='Name')

    table.heading('symb', text='Symbol')

    table.heading('pric', text='Price')

    table.heading('market\_kup', text='Market Kup')

    lbl.grid(row=0,column=0,sticky="e")

    pole.grid(row=0,column=1,sticky="w")

    btn.grid(row=0,column=2,sticky="w")

    btn1.grid(row=0,column=3,sticky="w")

    table.grid(row=1, columnspan=4)

    search.mainloop()

def btn3f(data):

    wb = openpyxl.Workbook()

    wb.create\_sheet(index = 0, title = "Save Data")

    sheet = wb["Save Data"]

    sheet["A1"] = "Name"

    sheet["B1"] = "Symbol"

    sheet["C1"] = "Price"

    sheet["D1"] = "Market Cap"

    for i in range(50):

        sheet["A"+str(i+2)] = str(data[i][1])

        sheet["B"+str(i+2)] = str(data[i][2])

        sheet["C"+str(i+2)] = str(data[i][3])

        sheet["D"+str(i+2)] = str(data[i][4])

        wb.save("Save.xlsx")

data\_all\_coins = []

data\_all\_coins = parsing(data\_all\_coins=data\_all\_coins)

data = [(data\_all\_coins[i][0],data\_all\_coins[i][1],

        data\_all\_coins[i][2],data\_all\_coins[i][3],

        data\_all\_coins[i][4]) for i in range(50)]

window = Tk()

window.title("Parsing CoinMarketCap")

window.resizable(width=0, height=0)

btn1 = Button(window, text='Update', background="yellow", command=btn1f)

btn2 = Button(window, text='Search', background="yellow", command=partial(btn2f,data))

btn3 = Button(window, text='Save', background="yellow", command=partial(btn3f,data))

lbl = Label(text = 'Update data!',background="green")

columns = ('num','nam','symb','pric','marketkup')

table = ttk.Treeview(window,columns=columns,show='headings')

table.heading('num', text='№')

table.heading('nam', text='Name')

table.heading('symb', text='Symbol')

table.heading('pric', text='Price')

table.heading('marketkup', text='Market Kup')

btn1.grid(row=0,column=0)

btn2.grid(row=0,column=1)

btn3.grid(row=0,column=2)

table.grid(row = 1,columnspan=3, sticky='nsew')

lbl.grid(row=2, columnspan=3,sticky="nsew")

table.delete(\*table.get\_children())

for inf in data :

    table.insert("",END,values=inf)

today = datetime.datetime.today()

lbl.config(text='Last update: '+today.strftime("%Y-%m-%d %H.%M.%S"))

window.mainloop()

4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

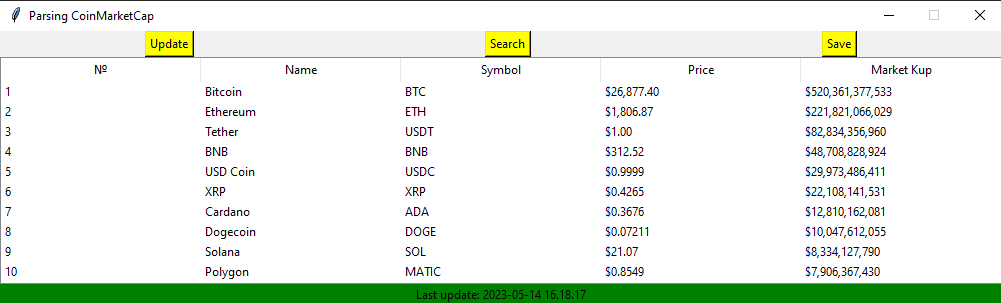
На рисунке 4.1 изображено главное меню оконного приложения.

Рисунок 4.1 - Главное меню

На рисунке 4.2 показано, что в таблице находится более 10 элементов.

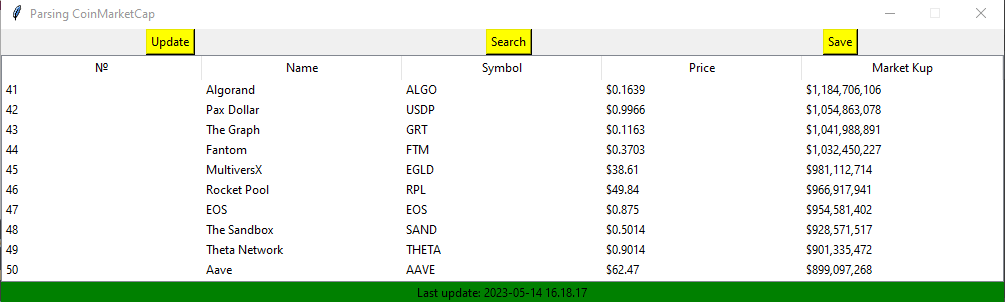


Рисунок 4.2 - Главное меню (Более 10 символов)

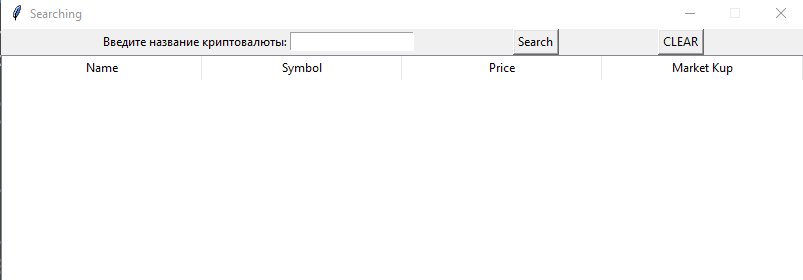
Меню поиска, реализованное в отдельном окне, показано на рисунке 4.3.

Рисунок 4.3 - Меню поиска

Примеры поиска изображены на рисунках 4.4 и 4.5.

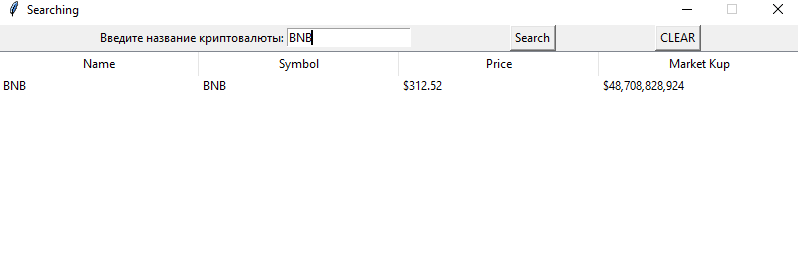


Рисунок 4.4 - Поиск криповалюты по названию

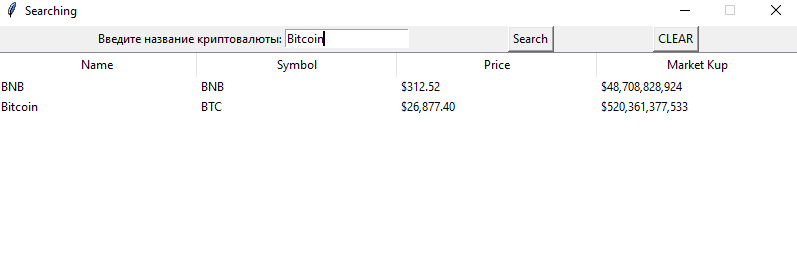


Рисунок 4.5 - Поиск криптовалюты по названию (2)

На рисунке 4.6 изображен функционал кнопки «CLEAR».

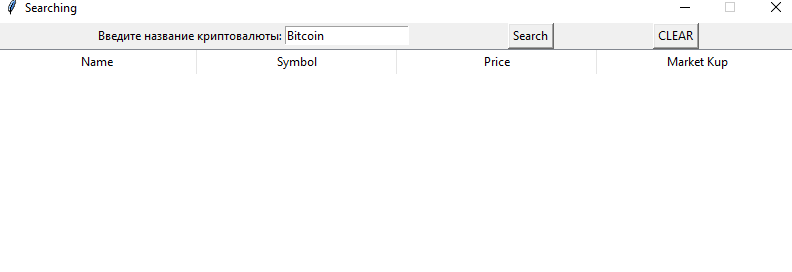


Рисунок 4.6 - Отчистка таблицы

На рисунках 4.7 и 4.8 показано сохранение данных в таблицу.

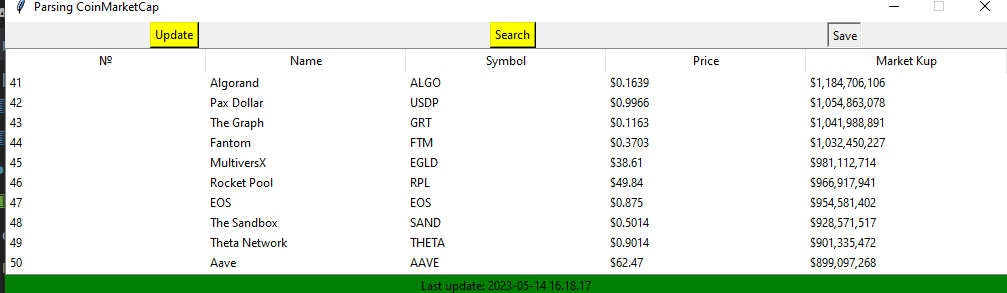


Рисунок 4.7 - Сохранение данных

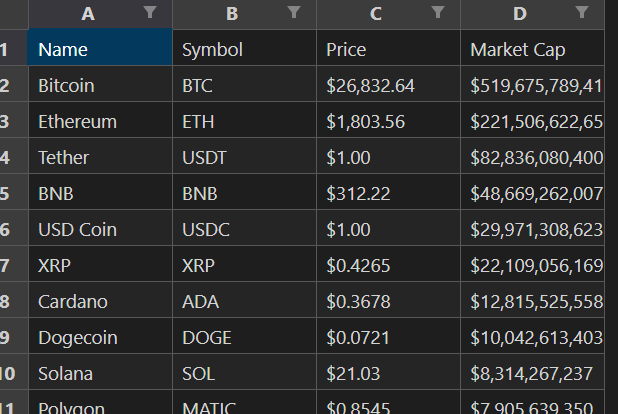


Рисунок 4.8 - Хранение данных в файле

СПИСОК ИСПОЛЬЗОВАННЫЗ ИСТОЧНИКОВ

1. Selenium

<https://selenium-python.readthedocs.io/>

1. Tkinter

<https://docs.python.org/3/library/tkinter.html>

1. Openpyxl

<https://openpyxl.readthedocs.io/en/stable/>

1. functools

<https://docs.python.org/3/library/functools.html>

1. Habr

<https://habr.com/>