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-811 \_\_\_\_\_\_\_\_\_\_\_\_\_/Мироненко К.А./

«04» июля 2020г.

Руководитель практики

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

«04» июля 2020г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2020г.

Оглавление

[Постановка задачи 3](#_Toc35194697)

[Описание используемых алгоритмов 4](#_Toc35194698)

[Листинг программы 5](#_Toc35194699)

[Результаты тестирования 6](#_Toc35194700)

[Список использованных источников 7](#_Toc35194701)

# 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 currencies.csv. Файл содержит данные о 25 наиболее ценных криптовалютах на 29.01.2020 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 |

1. Хранение

Типы и структуры для хранения данных: определеяются студентом

1. Обработка

Реализовать функцию поиска информации о свойствах криптовалюты по ее названию

# Описание используемых алгоритмов

def parser\_file() – проверяет наличие файла“currencies.csv” в директории на уровень выше исполняемого файлаю. В случае его наличия осуществляет парсинг файла, иначе вызывает исключение “FileNotFoundError” и завершает программу.

**Возвращает**: список словарей с ключами: name, marketCap, price

def parser\_web() – проверяет доступность сайта и если он доступен, осуществляет парсинг сайта “<https://coinmarketcap.com/>”. В случае недоступности сайта или отсутствии интернет-соединения вызывает исключение “ ConnectionError” и завершает программу.

**Возвращает**: список словарей с ключами: name, marketCap, price

def find\_by\_name(data, name) – осуществляет линейный поиск подстроки в строке по заданному ключу

**Принимает**:

* Data – список словарей
* Name – ключ поиска

**Возвращает**: список словарей, соответствующих ключу

def print\_data(data) – осуществляет вывод списка

**Принимает**:

* Data – список словарей

def main() – главная функция, осуществляет логическую составляющую скрипта(вызов необходимых функций).

# Листинг программы

**from** os **import** path

**import** csv

**import** requests

**from** bs4 **import** BeautifulSoup

FILE\_NAME = "../currencies.csv"

URL = "https://coinmarketcap.com/"

S\_BOLD = "\033[1m"

S\_RESET = "\033[0m"

C\_RED = "\033[31m"

C\_GREEN = "\033[32m"

C\_YELLOW = "\033[33m"

C\_CYAN = "\033[36m"

**def** parser\_file():

data = []

**try**:

file = open('/'.join(path.abspath(\_\_file\_\_).split('/')[:-1]) + '/' + FILE\_NAME, 'r')

table = csv.reader(file, delimiter=';')

**for** row **in** table:

item = {"name": row[0],

"marketCap": row[1],

"price": row[2]

}

data.append(item)

file.close()

**except** FileNotFoundError:

**print**(C\_RED + S\_BOLD + f"Ошибка! Файл \'{FILE\_NAME}\' не обнаружен!")

**print**(S\_RESET)

exit()

**return** data

**def** parser\_web():

**try**:

temp = 1

table = []

data = []

**print**(C\_YELLOW)

**print**("Парсинг сайта...")

**print**(S\_RESET)

**while** requests.get(URL + str(temp)).ok:

soup = BeautifulSoup(requests.get(URL + str(temp)).text, "html.parser")

page = soup.find\_all(class\_="cmc-table-row") # строки таблицы

**for** row **in** page:

item = {"name": row.find("td", class\_="cmc-table\_\_cell--sort-by\_\_name").div.a.text,

"marketCap": row.find("td", class\_="cmc-table\_\_cell--sort-by\_\_market-cap").div.text,

"price": row.find("td", class\_="cmc-table\_\_cell--sort-by\_\_price").a.text

}

**if** item["marketCap"] == "$?" **or** item["price"] == "$?":

**return** data

data.append(item)

temp += 1

**return** data

**except** requests.exceptions.ConnectionError:

**print**(C\_RED + "Ошибка! Неудалось подключиться к серверу!")

**print**(C\_CYAN + "Проверьте подключение к интернету, а также работоспособность сайта!")

**print**(S\_RESET)

exit()

**def** find\_by\_name(data, name):

items = []

**for** item **in** data:

**if** item.get("name").upper().startswith(name.upper()):

items.append(item)

**return** items

**def** print\_data(data):

**print**(S\_BOLD + C\_GREEN)

**print**(f"{'Название':35} {'Рыночная капитализация':>25} {'Цена':>15}")

**print**(C\_CYAN)

**for** item **in** data:

**print**(f"{item['name']:35} {item['marketCap']:>25} {item['price']:>15}")

**print**(C\_YELLOW)

**print**("Кол-во элементов: ", len(data))

**print**(S\_RESET)

**def** main():

data = []

**print**("Считать данные из...")

**while** True:

choice = input("[F]ile|[W]eb: ").upper()

**if** choice **not** **in** ['F', 'W']:

**print**(C\_RED + "Ошибка! Некорректный ввод!" + S\_RESET)

**else**:

**break**

**if** choice == 'F':

data = parser\_file()

**elif** choice == 'W':

data = parser\_web()

print\_data(data)

**while** True:

found = find\_by\_name(data, input("Введите строку для поиска криптовалюты: "))

**if** found:

print\_data(found)

**else**:

**print**(C\_RED + "Криптовалюты не найдены!" + S\_RESET)

**print**("\n\nПовторить поиск?")

**while** True:

choice = input("[Y]es|[N]o: ").upper()

**if** choice **not** **in** ['Y', 'N']:

**print**(C\_RED + "Ошибка! Некорректный ввод!" + S\_RESET)

**else**:

**break**

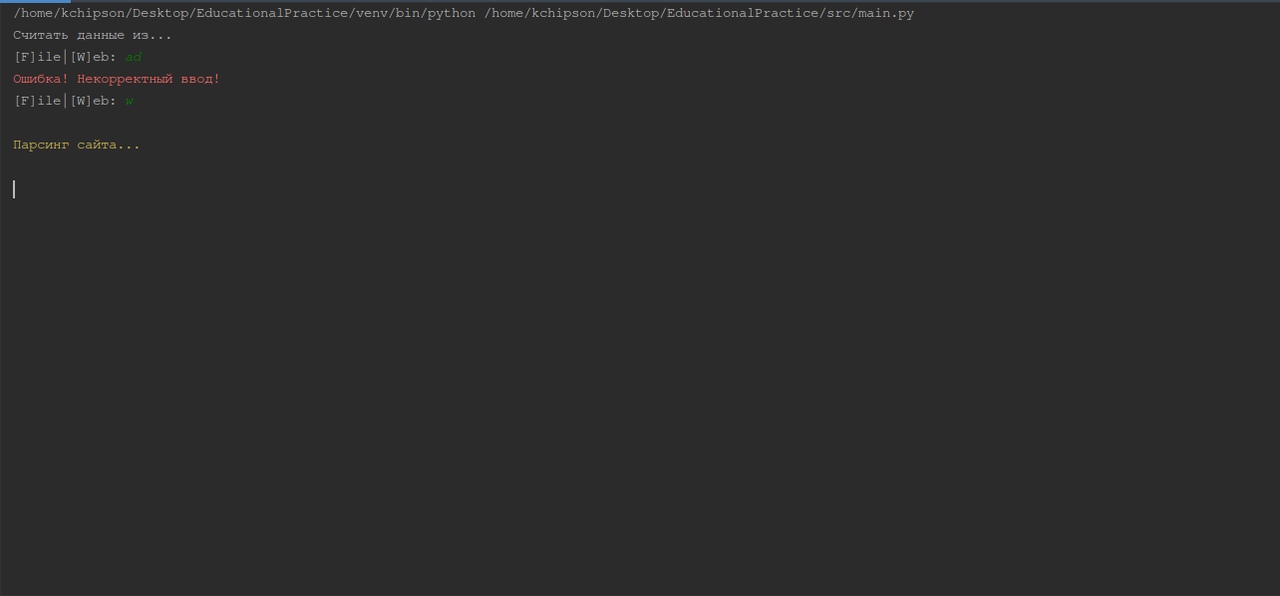
**if** choice == 'N':

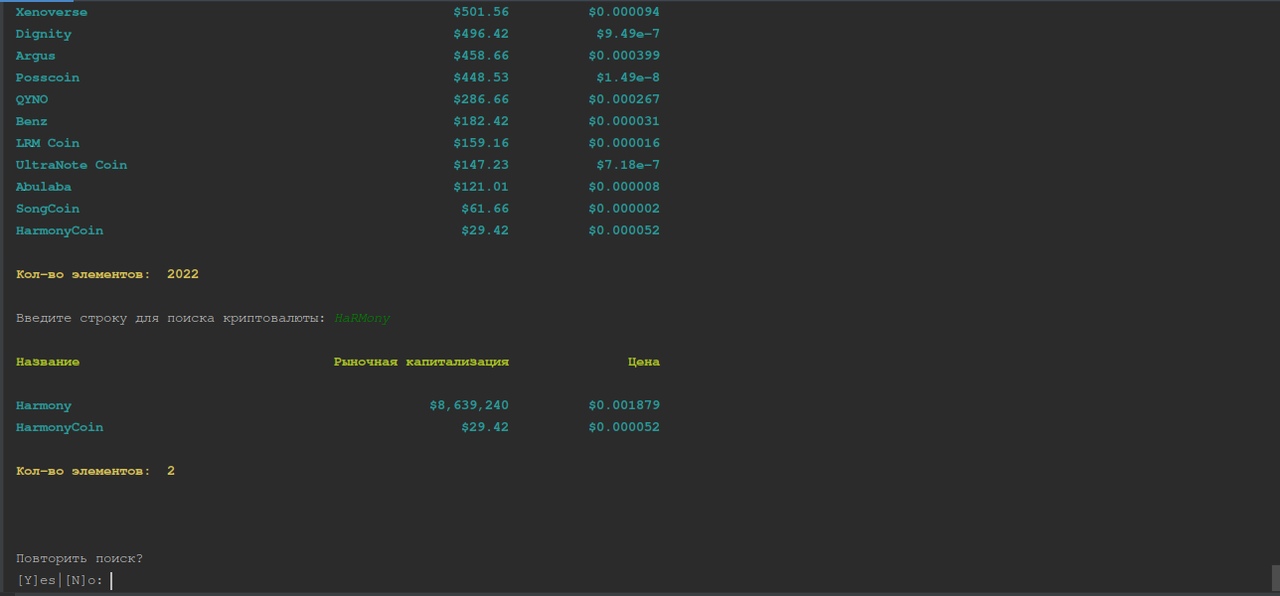
**break**

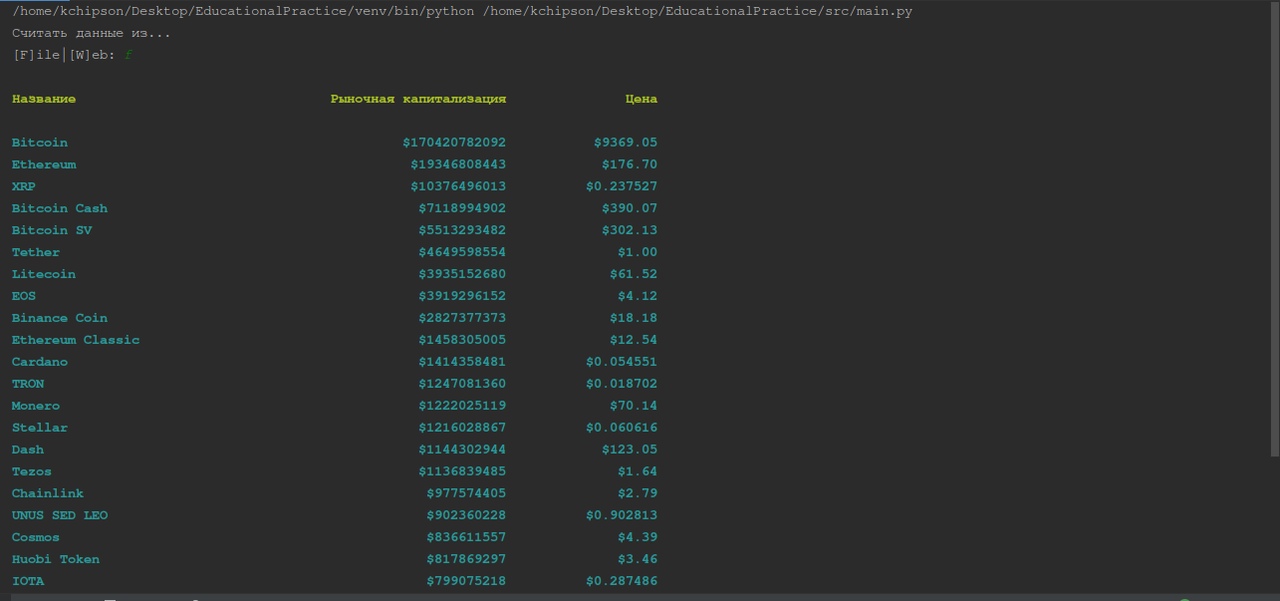
**if** \_\_name\_\_ == '\_\_main\_\_':

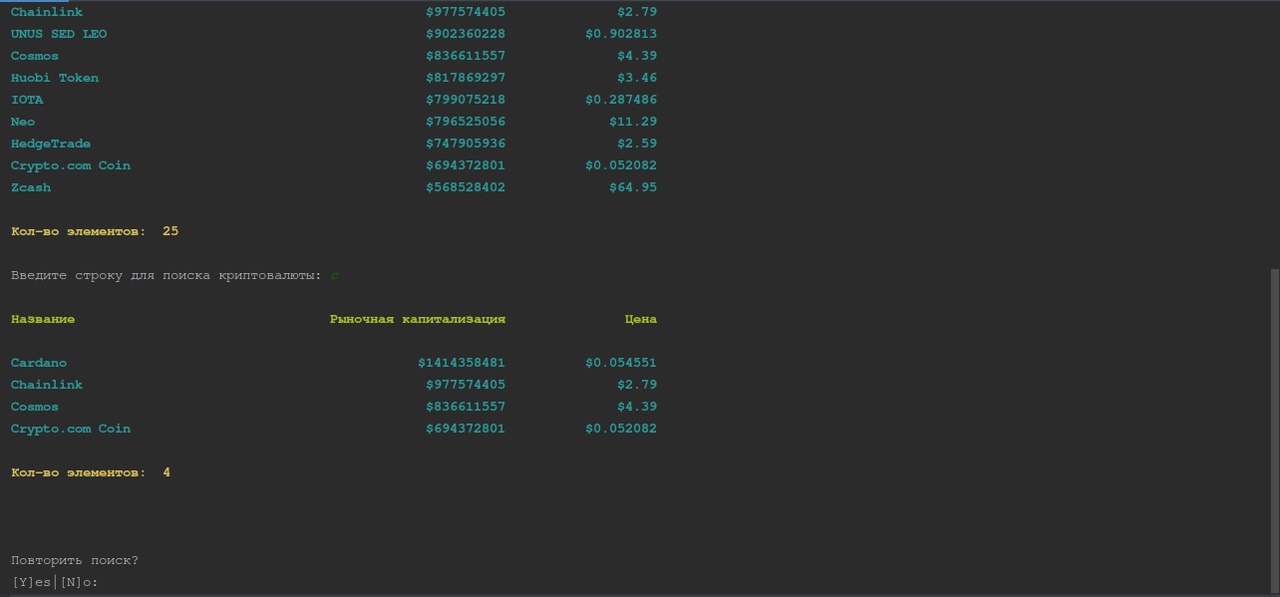
main()

# Результаты тестирования









# Список использованных источников

1. Python 3.8.2 documentation : The Python Tutorial : сайт – URL: <https://docs.python.org/3/tutorial/index.html> (дата обращения: 15.03.2020)
2. Python 3 для начинающих и чайников - уроки программирования : Самоучитель Python : сайт – URL: <https://pythonworld.ru/samouchitel-python> (дата обращения: 15.03.2020)
3. Python 3 - Изучение языка программирования : Виртуальная среда Python : сайт – URL: <https://python-scripts.com/virtualenv> (дата обращения: 15.03.2020)
4. Python 3 - Изучение языка программирования : Обрабатываем csv файлы — Модуль csv : сайт – URL: <https://python-scripts.com/import-csv-python> (дата обращения: 15.03.2020)
5. Python 3 - Изучение языка программирования : BeautifulSoup – парсинг HTML в Python на примерах : сайт – URL: <https://python-scripts.com/beautifulsoup-html-parsing> (дата обращения: 15.03.2020)