**Документация**

**к программному проекту по теме**

**“Парсер ЦИАН с использованием Телеграмм бота”**

**Выполнили: Яшин Кирилл Александрович, Поздняков Никита Эдуардович**

**Ликино-Дулёво**

**2023г**

**Введение:**

В настоящее время существует огромное количество сайтов ориентированные на разные сферы жизни общества. Некоторые из которых создают целую сеть, в которой легко запутаться обычному пользователю. Для этого был создан данный проект. Его цель помочь обычному пользователю подобрать себе квартиру по нужным ему условиям на сайте cian.ru, используя для этого Telegram-бота. Это позволит облегчить труды обычного человека, которому быстро и точно нужно найти желаемое по его запросам.

**Задача**:

Создать парсер для сайта ЦИАН с реализацией в формате телеграмм бота

**Реализация:**

Мой бот запускается по ссылке <https://t.me/cian_parse_bot>, но работать он скорее всего не будет, так как хоста нет. Если надо запустить бота, напишите автору, либо запустите на своём токене.

Код представляет класс Parser, который парсит данные о недвижимости на сайте cian.ru

**Модули:**

typing: библиотека для представления типов данных

bs4 (BeautifulSoup): библиотека для парсинга HTML и XML документов

cloudscraper: библиотека для обхода капчи сайта

json: библиотека для работы с JSON данными

transliterate: библиотека для транслитерации текста с русского на английский (в нашем случае)

**Параметры класса:**

minprice (int): минимальная цена квартиры (по умолчанию 0)

maxprice (int): максимальная цена квартиры (по умолчанию 0)

region (str): регион поиска (по умолчанию "Москва")

rooms (str): количество комнат (по умолчанию "1-6")

page (int): страница поиска (по умолчанию 1)

**Методы класса:**

**Метод** get\_phone\_number(link: str) -> Union[str, None]:

Получает номер телефона из объявления на сайте cian.

Параметры:

link (str): ссылка на объявление

Возвращает:

str: номер телефона

None: если номер не найден

**Метод** get\_rooms\_count(text: str) -> int:

Определяет количество комнат в квартире

Параметры:

text (str): текст объявления

Возвращает:

int: количество комнат в квартире

-1: если количество комнат не удалось определить

**Метод** get\_price(offer: BeautifulSoup) -> int:

Определяет цену квартиры из объявления.

Параметры:

offer (BeautifulSoup): объект BeautifulSoup с HTML кодом объявления

Возвращает:

int: цена квартиры

-1: если цена не найдена

**Метод** parse\_listing(offer):

Парсит объявление о продаже квартиры.

Параметры:

offer: объект BeautifulSoup с HTML кодом объявления

**Метод** start\_session() -> list:

Инициирует сессию и запускает парсинг страниц с объявлениями о продаже квартир на сайте Avito.

Возвращает:

list: список словарей с информацией о найденных квартирах (номер телефона, цена, количество комнат и т.д.).

**Документация к файлу tg\_bot.py:**

**Импорт библиотек:**

logging - модуль для логгирования

telegram.ext - модуль для работы с Telegram API

cian\_parser - модуль с функцией для парсинга сайта cian.ru

**Инициализация логгера:**

В формате basicConfig указываются формат сообщений и уровень логирования (INFO)

getLogger(name) создает логгер для данного модуля

**Инициализация переменных:**

TOKEN - токен для работы с Telegram API

application - объект Application для работы с Telegram API

user\_info - словарь для хранения настроек пользователя, по умолчанию установлены следующие значения: username='', city='Москва', rooms='1-6', pages=2, min\_price=0, max\_price=0

**Описание асинхронных функций:**

send\_file - отправляет пользователю json-файл с результатами парсинга

parse - основная функция, запускает парсинг сайта cian.ru и отправляет результаты пользователю в json-файл

settings - выводит на экран текущие настройки пользователя

available\_cities - представляет список доступных для поиска городов

start - приветствие пользователя и описание бота

min\_price - сохраняет в user\_info минимальную цену, введенную пользователем

max\_price - сохраняет в user\_info максимальную цену, введенную пользователем

city - сохраняет в user\_info город, введенный пользователем

rooms - сохраняет в user\_info количество комнат, введенное пользователем

pages - сохраняет в user\_info количество страниц, введенное пользователем

help - описание команд для работы с ботом

**Логика работы функции parse:**

Из объекта update получаем информацию о пользователе (user) и его username

Сохраняем username в user\_info

Отправляем сообщение пользователю о начале парсинга

Запускаем парсинг сайта cian.ru, используя модуль cian\_parser, с передачей параметров pages, minprice и maxprice из user\_info

Отправляем пользователю json-файл с результатами парсинга

**Обработка исключений:**

Если пользователь ввел город, которого нет в реестре, выводится соответствующее сообщение

Если пользователь ввел недопустимое количество комнат, выводится соответствующее сообщение.

**Пример работы бота:**

  