Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение высшего образования «Омский государственный технический университет»
Кафедра «Автоматизированные системы обработки информации и управления»

ОТЧЁТ
ПО КУРСОВОЙ РАБОТЕ

по дисциплине «Проектная деятельность»
студента Скутина Артёма Спартаковича, группа ПИН-222

Пояснительная записка

Шифр работы От-2068998-43-ПИН-222-22 ПЗ
Направление 09.03.04

Старший преподаватель А. А. Кабанов

Студент А. С. Скутин

Омск 2023

РЕФЕРАТ

Пояснительная записка, 17 с., 1 ч., 14 рис., 1 табл., 5 источника, 1 приложение

ПАРСЕР, ДИНАМИЧЕСКИЙ САЙТ, МАРКЕТПЛЕЙС

Объектом исследования являются динамические сайты.

Целью работы является разработка парсера динамического сайта (маркетплейса “Wildberries”).

В ходе работы над курсовой работой был проведён анализ задачи и определён алгоритм решения.

В результате работы была написана и протестирована программа для решения поставленной задачи.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины с соответствующими определениями.

Парсинг – это процесс извлечения нужной информации из структурированных данных или документов, например, из веб-страниц, с целью последующего анализа или использования этих данных в других приложениях.

Python – высокоуровневый язык программирования.

Библиотека (в ЯП) – это набор предварительно написанных модулей или функций, которые предоставляют различные инструменты и возможности для выполнения определенных задач.

Маркетплейс — это торговая площадка, которая продаёт товары и услуги разных продавцов через интернет, является лишь посредником.

JSON (англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем отчете применяют следующие сокращения и обозначения.

Id – идентификатор

URL – Uniform Resource Locator

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данная курсовая работа выполняется в рамках дисциплины “Проектная деятельность” в третьем семестре и направлена на программное решение задачи средней сложности, её последующее тестирование и оформление отчёта о проделанной работе в соответствии с ГОСТ 7.32-2017.

Задача является разработкой и реализаций парсера динамического сайта “Wildberries”.

В первом разделе описывается поставленная задача.

Во втором разделе описывается функционал парсера.

В третьем разделе описываются особенности реализации на языке программирования python, принцип его работы и методы, которые были рассмотрены и применены для получения окончательного результата.

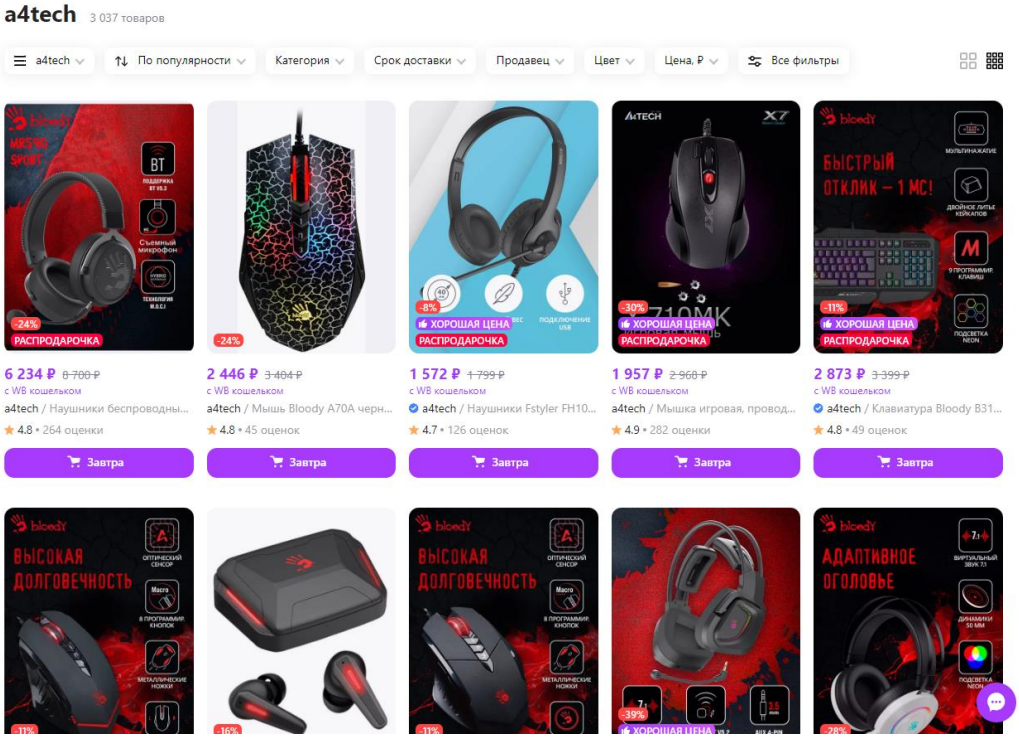
В четвертом разделе представлены результаты тестов, которые были проведены для проверки функциональности парсера.

1 ПОСТАНОВКА ЗАДАЧИ

Требуется реализовать парсер, достающий с маркетплейса “Wildberries”, реализованного в виде динамического сайта, такие данные как: id, названия, цены, бренд, количество продаж, рейтинг и наличие товаров – и записывающий данные в файл.

2 ОБЗОР ФУНКЦИОНАЛА ПАРСЕРА

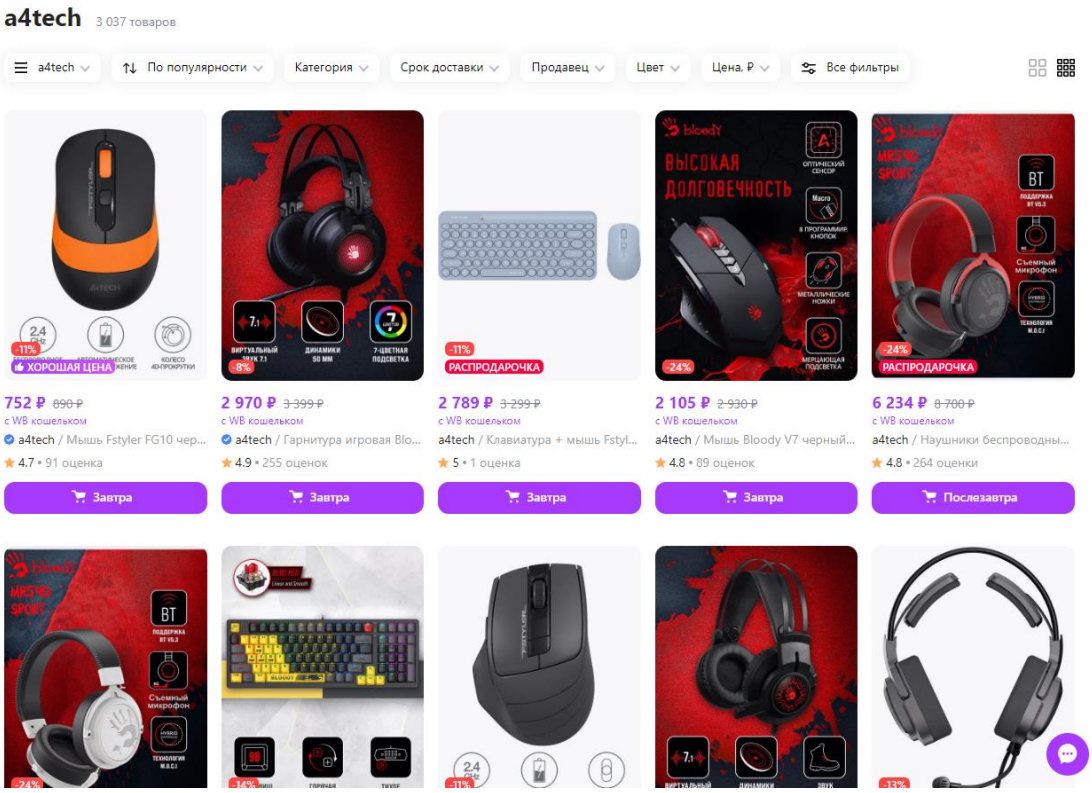
Итоговая программа, написанная на языке python, парсит данные о товаре определенного бренда и работает следующим образом.



На скриншоте страница бренда a4tech, отсюда будут парситься данные о товарах.

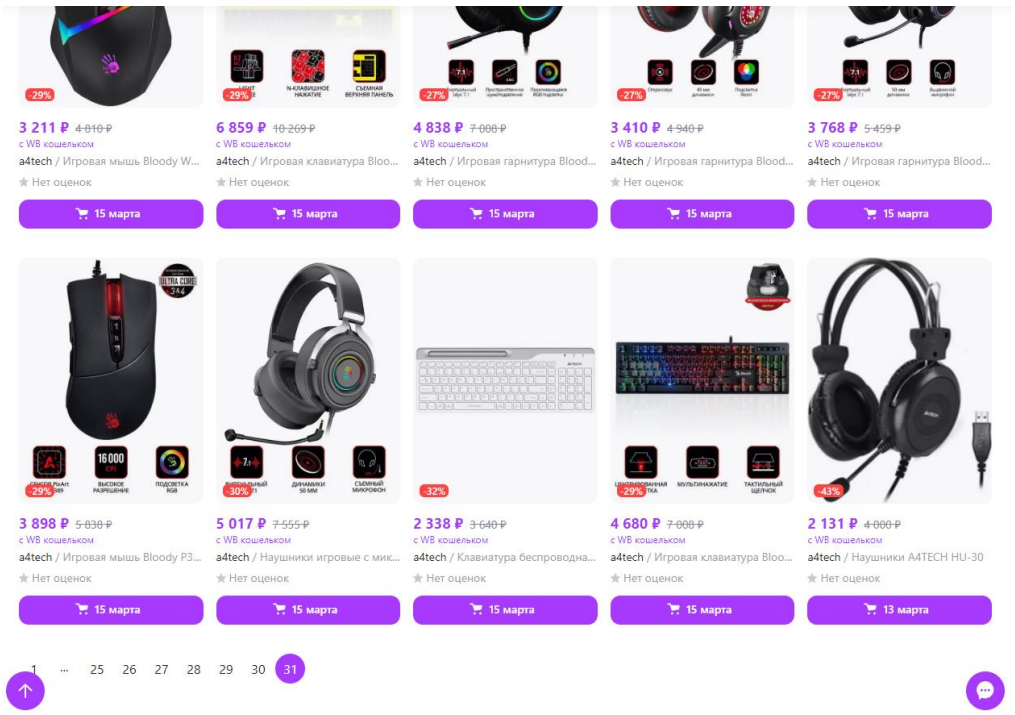
```
id,название,цена,бренд,продаж,рейтинг,в наличии
203454487,Наушники беспроводные большие с микрофоном Bloody MR590,6563.0,a4tech,24,5,5
198126300,Мышь Bloody A70A черный оптическая (6200dpi) USB (7but),2575.0,a4tech,24,5,14
123328952,Наушники Fstyler FH100U,1655.0,a4tech,8,5,19
179742126,"Мышка игровая, проводная, X7, X-710MK",2060.0,a4tech,30,5,13
139271651,Клавиатура Bloody B310N цвет черный,3025.0,a4tech,11,5,60
27386730,Игровая мышь компьютерная Bloody V8,1779.0,a4tech,11,5,13
102345257,Наушники игровые Bloody M70 беспроводные,4160.0,a4tech,16,5,9
27382261,Игровая мышь компьютерная Bloody V7,1850.0,a4tech,11,5,10
146586053,Гарнитура игровая BLOODY MR575 (MR575),5703.0,a4tech,39,5,61
184093246,"Наушники проводные, большие, игровые, Bloody G521 WHITE",3162.0,a4tech,28,5,71
163363205,Мышь Bloody A90 черный оптическая (6200dpi),1701.0,a4tech,29,5,27
74337803,Наушники игровые с микрофоном Bloody G535,3127.0,a4tech,8,5,39
163335536,Мышь Bloody W95 Max Sports,4004.0,a4tech,17,5,25
186762464,Мышь беспроводная A4 Fstyler FG10,729.0,A4Tech,64,5,22
27386769,Мышь X-710BK черный,1601.0,a4tech,11,5,3
47675223,Клавиатура Fstyler FBK25,2491.0,a4tech,11,5,31
160014346,Клавиатура FStyler FK15,1111.0,a4tech,43,5,54
47675513,Наушники игровые с микрофоном Bloody G575 Punk,3772.0,a4tech,8,5,57
168242595,Клавиатура Bloody S98 Naraka,6229.0,a4tech,19,5,50
74424771,Комплект клавиатура+мышь 3330N черный черный,2046.0,a4tech,11,5,8
47675200,Клавиатура игровая механическая Bloody B865N,5339.0,a4tech,11,5,56
75074133,Наушники с микрофоном HS-11 черный 2м накладные оголо,797.0,a4tech,40,4,50
27382257,Игровая мышь проводная Bloody P91s,1957.0,a4tech,11,5,11
```

Как видно, позиции на первой странице совпадают.




```
38135291,Игровая мышь компьютерная 8000 dpi Bloody J90s,2661.0,a4tech,11,5,8
202382273,Игровые наушники с микрофоном Bloody G575 Royal Violet,3495.0,a4tech,30,5,66
44190525,Мышь Fstyler FG10 черный оранжевый (2000dpi),792.0,a4tech,11,5,8
65095569,Гарнитура игровая Bloody G525 черный (g525 black),3127.0,a4tech,8,5,63
193635178,Клавиатура + мышь Fstyler FG3200 Air,2936.0,a4tech,11,5,26
163275339,Мышь Bloody V7 черный оптическая (3200dpi),2216.0,a4tech,24,5,29
203454485,Наушники беспроводные большие с микрофоном Bloody MR590,6563.0,a4tech,24,5,4
203454486,Наушники беспроводные большие с микрофоном Bloody MR590,6563.0,a4tech,24,5,5
166764951,Клавиатура Bloody S98 механическая,5784.0,a4tech,14,5,50
44190528,Мышь Fstyler FG30 серый оптическая (2000dpi),1245.0,a4tech,11,5,11
27380493,"Гарнитура игровая Bloody J437, черный (j437)",2759.0,a4tech,8,5,72
145448764,"Гарнитура игровая Bloody G575 Pro,цвет серый",4047.0,a4tech,13,5,7
197903014,Мышь компьютерная игровая X7,1737.0,A4tech,45,0,17
27380490,"Гарнитура игровая Bloody G501, черный (g501)",4047.0,a4tech,8,5,58
112125474,Игровой коврик для мыши Bloody B-080S,1557.0,a4tech,77,0,28
166764950,Клавиатура Bloody S98 механическая,6051.0,a4tech,11,5,50
49854258,Наушники с микрофоном (G528C) Bloody G528C черный,2943.0,a4tech,8,5,74
15527713,Коврик для мыши игровой A4 Bloody B-035S черный 350x280x2мм,489.0,a4tech,11,5,10
136056623,"Мышь Bloody ES5,цвет черный",1504.0,a4tech,11,5,3
18459710,Наушники игровые с микрофоном Bloody G570 черный серыйY),3549.0,a4tech,8,5,55
17782387,Мышь Fstyler FG30S белый серый оптическая (2000dpi),1099.0,a4tech,11,5,6
172943797,Наушники накладные Bloody MH360 белый беспроводные,2475.0,a4tech,21,5,19
38135289,Игровая мышь компьютерная Bloody X5 Pro,3381.0,a4tech,11,5,3
17782386,Мышь Fstyler FG30S серый оранжевый (2000dpi),1023.0,a4tech,11,5,6
```

На второй странице позиции также совпадают (102 позиция в списке – первая на второй странице).




```
158892169,Клавиатура A4 Bloody B188 USB Multimedia,3516.0,a4tech,32,0,50
165219015,Компьютерная мышь A4Tech,2581.0,a4tech,11,0,36
164994053,Беспроводная мышь A4Tech G3-310N,2225.0,a4tech,11,0,38
158885012,Игровая мышь Bloody W70-Max,3473.0,a4tech,29,0,13
158895601,Клавиатура + Мышка беспроводные USB Fstyler FG-1010,2682.0,a4tech,32,0,50
109825948,Коврик для мыши черный,363.0,a4tech,69,4,5
160631897,Мышь беспроводная G10-770FL 2000 dpi,3003.0,a4tech,33,0,45
159189215,Мышка проводная USB N-708X Black,1246.0,a4tech,31,0,27
153944422,Мышь oscar neon gaming mouse X77,1922.0,a4tech,11,0,57
158810988,Игровая гарнитура Bloody G500,4243.0,a4tech,27,0,34
164732376,Коврик для мыши X7-200MP,469.0,a4tech,28,0,32
158884744,Игровая мышь Bloody W60-Max,3381.0,a4tech,29,0,13
158878968,"Игровая клавиатура Bloody B810RC, Yellow",7220.0,a4tech,29,0,50
158811087,Игровая гарнитура Bloody G528C,5093.0,a4tech,27,0,34
158810849,Игровая гарнитура Bloody G300,3590.0,a4tech,27,0,34
158810948,Игровая гарнитура Bloody G350,3967.0,a4tech,27,0,34
158884478,Игровая мышь Bloody P30 PRO,4104.0,a4tech,29,0,13
159226068,Наушники игровые с микрофоном Bloody G535,5282.0,a4tech,30,0,34
158893192,Клавиатура беспроводная USB Fstyler FBK25,2462.0,a4tech,32,0,50
158878611,"Игровая клавиатура Bloody B500N, Black (B500N)",4927.0,a4tech,29,0,50
167753039,Наушники A4TECH HU-30,2244.0,a4tech,43,0,101
```

Все позиции совпадают вплоть до последней, 31 страницы.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ

Парсер реализован методом get/post. В нем используются такие библиотеки как: requests (отвечает за отправку веб-запросов), re (формирование регулярных выражений), csv (работа с файлами формата .csv) и pydantic (работа с json). Далее приведены пояснения к коду программы.

При запуске файла parser.py считывается ссылка на страницу сайта:

```
53 ▶  if __name__ == "__main__":
54        Parse("https://www.wildberries.ru/brands/9292-a4tech").parse()
```

С помощью регулярного выражения вычленяется id бренда:

```
6   class Parse:
        Forestjaba
7       def __init__(self, url: str):
8           self.brand_id = self.__get_id(url)
9
        1 usage   Forestjaba
10      @staticmethod
11      def __get_id(url: str):
12          regex = "(?<=brands/).+(?=-)"
13          brand_id = re.search(regex, url)[0]
14          return brand_id
```

Создается файл в формате .csv, в него записываются названия столбцов:

```
36      def __create_csv(self):
37          with open("data.csv", mode="w", encoding="utf-8", newline="") as file:
38              writer = csv.writer(file)
39              writer.writerow(['id', 'название', 'цена', 'бренд', 'продаж', 'рейтинг', 'в наличии'])
```

В файле models.py указываем названия нужных параметров из json:

```
19  class Items(BaseModel):
20      products: list[Item]
```

```
3   class Item(BaseModel):
4       id: int
5       name: str
6       salePriceU: float
7       brand: str
8       sale: int
9       rating: int
10      volume: int
```

Так как цена указана в копейках, необходимо разделить ее на 100:

```python
@root_validator(pre=True)
def convert_price(cls, values:dict):
    price = values.get("salePriceU")
    if price is not None:
        values["salePriceU"] = price / 100
    return values
```

В файле parser.py эти данные записываются в файл .csv:

```python
def __save_csv(self, items):
    with open("data.csv", mode="a", encoding="utf-8", newline="") as file:
        writer = csv.writer(file)
        for article in items.products:
            writer.writerow([article.id,
                             article.name,
                             article.salePriceU,
                             article.brand,
                             article.sale,
                             article.rating,
                             article.volume])
```

В цикле открывается json страницы (в параметрах изменяются бренд и номер страницы), посредством метода __save_csv из него достаются нужные данные, номер страницы изменяется на следующий:

```python
def parse(self):
    i = 1
    self.__create_csv()
    while True:
        params = {
            'appType': '1',
            'brand': f'{self.brand_id}',
            'curr': 'rub',
            'dest': '-1257786',
            'page': f'{i}',
            'sort': 'popular',
            'spp': '30',
        }
        response = requests.get('https://catalog.wb.ru/brands/a/catalog', params=params)
        info = Items.parse_obj(response.json()["data"])
        if not info.products:
            break
        self.__save_csv(info)
        i += 1
```

ТЕСТИРОВАНИЕ ПРОГРАММЫ

Результаты тестирования представлены в таблице 1.

Таблица 1 – Тестирования программы

№ теста	Описание теста	Предполагаемый результат	Тест
	Товары расположены на нескольких страницах	Отображаются все, пагинация работает	
	Товаров меньше, чем помещается на 1 странице	Работоспособность не нарушится	
	Цена, отображаемая в файле (цена без скидки, цена со скидкой, цена с WB кошельком)	Отображается цена со скидкой	
	Цена отображается в копейках	Нет, отображается в рублях	
	Рейтинг товара указывается корректно	Да, с точностью до десятых	
	Товары на странице заканчиваются	Условие завершения работы работает корректно	

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы был разработан парсер динамического сайта “Wildberries” с возможностью дальнейшей его модификации и добавления функций. В процессе разработки были укреплены навыки работы с библиотекой requests, получен опыт работы с библиотеками re, csv и pydantic (с классами BaseModel и root_validator).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Веб-скрапинг динамических сайтов с помощью Python – URL:

2) Scrape a Dynamic Website with Python – URL:

https://scrapingant.com/blog/scrape-dynamic-website-with-python

Продвинутый парсинг сайтов на Python – URL:

Основы парсинга на Python: от Requests до Selenium / Хабр – URL:

ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код программы на языке Python

parser.py

```
import requests
import re
import csv
from models import Items

class Parse:
    def __init__(self, url: str):
        self.brand_id = self.__get_id(url)

    @staticmethod
    def __get_id(url: str):
        regex = "(?<=brands/).+(?=-)"
        brand_id = re.search(regex, url)[0]
        return brand_id

    def parse(self):
        i = 1
        self.__create_csv()
        while True:
            params = {
                'appType': '1',
                'brand': f'{self.brand_id}',
                'curr': 'rub',
                'dest': '-1257786',
                'page': f'{i}',
                'sort': 'popular',
                'spp': '30',
            }
            response = requests.get('https://catalog.wb.ru/brands/a/catalog', params=params)
            info = Items.parse_obj(response.json()["data"])
            if not info.products:
                break
            self.__save_csv(info)
            i += 1

    def __create_csv(self):
        with open("data.csv", mode="w", encoding="utf-8", newline="") as file:
            writer = csv.writer(file)
            writer.writerow(['id', 'название', 'цена', 'бренд', 'продаж', 'рейтинг', 'в наличии'])
```

```
    def __save_csv(self, items):
        with open("data.csv", mode="a", encoding="utf-8", newline="") as file:
            writer = csv.writer(file)
            for article in items.products:
                writer.writerow([article.id,
                                 article.name,
                                 article.salePriceU,
                                 article.brand,
                                 article.sale,
                                 article.rating,
                                 article.volume])

if __name__ == "__main__":
    Parse("https://www.wildberries.ru/brands/9292-a4tech").parse()
```

models.py

```
from pydantic import BaseModel, root_validator

class Item(BaseModel):
    id: int
    name: str
    salePriceU: float
    brand: str
    sale: int
    rating: int
    volume: int

    @root_validator(pre=True)
    def convert_price(cls, values:dict):
        price = values.get("salePriceU")
        if price is not None:
            values["salePriceU"] = price / 100
        return values

class Items(BaseModel):
    products: list[Item]
```