**Национальный исследовательский технологический университет «МИСИС»**

**Институт Информационных технологий и Компьютерных наук**

**(ИТКН)**

**Курс «Сбор датасета»**

**Лабораторная работа №4**

Выполнили: студенты группы БИВТ-22-ИСАД-1

Ильющеня Дмитрий Павлович

Иванюта Михаил Игоревич

Изображение выглядит как черный, темнота

Контент, сгенерированный ИИ, может содержать ошибки.

Проверила: Прокофьева Е.Н.

**Москва, 2025**

**Сбор датасета**

**Основные этапы реализации**

1. **Написание скрипта для web скрпинга**
2. import requests
3. from bs4 import BeautifulSoup
4. import time
5. import random
6. class Application:
7. def \_\_init\_\_(self):
8. self.starter = 1
9. self.end = 5
10. self.all\_appartments\_dict = {}
11. self.pages = 150
13. def parcer(self):
14. if self.end >= self.pages:
15. return self.all\_appartments\_dict
16. else:
17. for i in range(self.starter, self.end):
18. # запрос на сайт
19. url = f'https://www.russianrealty.ru/%D0%9F%D1%80%D0%BE%D0%B4%D0%B0%D0%B6%D0%B0-%D0%BA%D0%B2%D0%B0%D1%80%D1%82%D0%B8%D1%80/{i}/'
20. headers = {
21. "Accept":
22. "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8",
23. "User-Agent":
24. "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 Firefox/106.0",
25. }
26. req = requests.get(url, headers=headers)
27. src = req.text
28. # запись в файл
29. with open("index.html", "w", encoding="utf-8") as file:
30. file.write(src)
31. with open("index.html", "r", encoding="utf-8") as file:
32. src = file.read()
33. # парс данных
34. soup = BeautifulSoup(src, "lxml")
35. all\_appartments\_hrefs = soup.find\_all("div", {"class": "hproduct"})
36. for div in all\_appartments\_hrefs:
37. # имя, ссылка, количество комнат
38. item\_name = div.find('a').text
39. item\_name = item\_name.replace("&sup2", "2")
40. item\_href = "https:" + div.find('a').get('href')
41. item\_rooms\_number = int(item\_name[0])
42. # цена и адрес
43. item\_price = int(div.find('span').text)
44. item\_adress\_details = div.find('p', class\_ = "adr").text
45. in\_point\_item\_adress\_details = item\_adress\_details.index("«")
46. item\_adress = item\_adress\_details[0 : in\_point\_item\_adress\_details]
47. # количество минут до метро
48. try:
49. in\_point\_item\_adress\_details\_ending = item\_adress\_details[::-1].index(",")
50. in\_point\_item\_adress\_details\_ending2 = item\_adress\_details[::-1].index(" ")
51. item\_minutes\_for\_subway = item\_adress\_details[len(item\_adress\_details) - in\_point\_item\_adress\_details\_ending + 1 : len(item\_adress\_details) - 1]
52. in\_point\_item\_minutes\_for\_subway = item\_minutes\_for\_subway.index(" ")
53. item\_minutes\_for\_subway = int(item\_minutes\_for\_subway[: in\_point\_item\_minutes\_for\_subway])
54. except ValueError:
55. item\_minutes\_for\_subway = 'not stated'
56. # площадь квартиры и кухни
57. in\_point\_item\_name\_beginning = item\_name.index(" ")
58. in\_point\_item\_name\_ending = item\_name[::-1].index(" ")
59. squares = item\_name[in\_point\_item\_name\_beginning + 1 : len(item\_name) - (in\_point\_item\_name\_ending + 1)]
60. in\_point\_squares\_beginning = squares.index("/")
61. appartment\_square = float(squares[0 : in\_point\_squares\_beginning])
62. in\_point\_squares\_ending = squares[::-1].index("/")
63. appartment\_kitchen\_square = float(squares[len(squares) - in\_point\_squares\_ending :])
64. appartment\_info = div.find\_all('li') # список li элементов, через которые берётся информация ниже
65. # этаж квартиры и дома
66. appartment\_floors\_info = appartment\_info[0].text
67. in\_point\_appartment\_floors\_info\_sliced = appartment\_floors\_info.index("-")
68. appartment\_floors\_info = appartment\_floors\_info[: in\_point\_appartment\_floors\_info\_sliced]
69. in\_point\_appartment\_floors\_info\_beginning = appartment\_floors\_info.index(" ")
70. appartment\_floor = int(appartment\_floors\_info[0 : in\_point\_appartment\_floors\_info\_beginning])
71. in\_point\_appartment\_floors\_info\_ending = appartment\_floors\_info[::-1].index(" ")
72. appartment\_floors\_in\_building = int(appartment\_floors\_info[len(appartment\_floors\_info) - in\_point\_appartment\_floors\_info\_ending :])
73. # наличие балкона или лоджии
74. appartment\_balcony\_lodge = appartment\_info[1].text
75. if appartment\_balcony\_lodge.find("балкон") != -1 or appartment\_balcony\_lodge.find("лоджия") != -1:
76. appartment\_balcony\_lodge = True
77. else:
78. appartment\_balcony\_lodge = False
79. # тип ремонта
80. if appartment\_balcony\_lodge:
81. appartment\_renovation\_type = appartment\_info[2].text
82. else:
83. appartment\_renovation\_type = appartment\_info[1].text
84. in\_point\_appartment\_renovation\_type = appartment\_renovation\_type.index(" ")
85. appartment\_renovation\_type = appartment\_renovation\_type[in\_point\_appartment\_renovation\_type + 1 : ]
86. if appartment\_renovation\_type == "требуется ремонт" or appartment\_renovation\_type == "без отделки":
87. appartment\_renovation\_type = "нет ремонта"
88. elif appartment\_renovation\_type == "косметический" or appartment\_renovation\_type == "под чистовую":
89. appartment\_renovation\_type = "эконом"
90. else:
91. appartment\_renovation\_type = "улучшенный"
92. file.close()
93. # словарь с ключом в виде названия квартиры и значением в виде списка переменных
94. self.all\_appartments\_dict[item\_name] = [ item\_href,                              # ссылка / string
95. item\_price,                             # цена / int
96. item\_adress,                            # адрес / string
97. item\_minutes\_for\_subway,                # кол-во минут до метро / int
98. item\_rooms\_number,                      # количество комнат / int
99. appartment\_square,                      # общая площадь квартиры / float
100. appartment\_kitchen\_square,              # площадь кухни / float
101. appartment\_floor,                       # этаж квартиры / int
102. appartment\_floors\_in\_building,          # этажность дома / int
103. appartment\_balcony\_lodge,               # наличие балкона или лоджии / bool
104. appartment\_renovation\_type,             # тип ремонта / string
105. ]
106. self.end += 5; self.starter += 5
107. print(f"процент выполениния: {round((self.end / self.pages) \* 100)}% всего спаршено: {len(self.all\_appartments\_dict)}")
108. return self.parcer()

111. if \_\_name\_\_ == "\_\_main\_\_":
112. x = Application()
113. print(x.parcer())
114. **Внос данных в удобное хранилище sqlite database**

**Изображение выглядит как текст, меню, документ, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

1. **Последующий анализ**

**Полученный результат**

По итогу работ – удалось собрать датасет по стоимости и расположению недвижимости в Москве. Датасет готов для дальнейшей обработки и анализа