

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №6
«Разработка простого бота и бота на основе конечного автомата для Telegram с
использованием языка Python.»

Выполнил:
студент группы ИУ5-34Б:
Такташова Дарья Юрьевна
Подпись и дата:

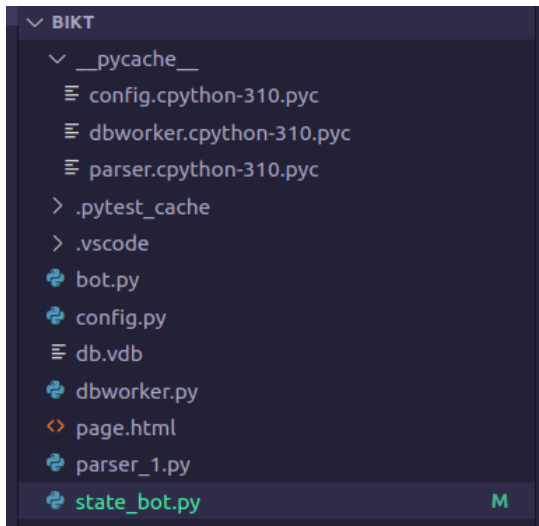
Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2022 г.

Задание:

1. Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.
2. Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

Код программы:



bot.py:

```
1 #5597125983:AAFcvSuyrWoSHXnNoJTntzYY3kNUd62ucvs
2 import os
3 import telebot
4 from telebot import types
5 from parser_1 import parse_user_datafile_bs
6
7
8 # Токент бота
9 TOKEN = '5597125983:AAFcvSuyrWoSHXnNoJTntzYY3kNUd62ucvs'
10
11 # Создание бота
12 bot = telebot.TeleBot(TOKEN)
13
14 @bot.message_handler(commands=['start'])
15 def start(message):
16     markup = types.InlineKeyboardMarkup()
17     markup.add(telebot.types.InlineKeyboardButton(text='Понедельник' , callback_data = 11))
18     markup.add(telebot.types.InlineKeyboardButton(text='Вторник' , callback_data = 12))
19     markup.add(telebot.types.InlineKeyboardButton(text='Среда' , callback_data = 13))
20     markup.add(telebot.types.InlineKeyboardButton(text='Четверг' , callback_data = 14))
21     markup.add(telebot.types.InlineKeyboardButton(text='Пятница' , callback_data = 15))
22     markup.add(telebot.types.InlineKeyboardButton(text='Суббота' , callback_data = 16))
23     bot.send_sticker(message.chat.id, 'CAACAgIAAxkBAAEGefdjed_07xn-xHhkPfaFd98KoBW7hQACLBgAAvdbgEjPGPx-6kVXrisE')
24     bot.send_message(message.chat.id, 'Приветтики студент ИУ5-34Б, выбери день недели, чтобы глянуть расписание' , reply_markup=markup)
25
26 group = 4
27
28 @bot.callback_query_handler(func=lambda call: True)
29 def query_handler(call):
30     global group
31     sh = []
32     days = ['11', '12', '13', '14', '15', '16']
33     markup2 = types.InlineKeyboardMarkup()
34
35     if call.data in days:
36         sh = parse_user_datafile_bs('page.html', group)
37         ch_r = []
38         zn_r = []
39         for s in sh[int(call.data) % 10 - 1]:
```

```

40     print(s)
41     print()
42     res1 = []
43     res2 = []
44     for key in s.keys():
45         if key == 'chisl':
46             res1.append(s[key])
47         elif key == 'znam':
48             res2.append(s[key])
49         else:
50             res1.append(s[key])
51             res2.append(s[key])
52     if len(res1) > 1:
53         ch_r.append(res1)
54     if len(res2) > 1:
55         zn_r.append(res2)
56     bot.send_message(call.message.chat.id, f"Числитель\n")
57     for s in ch_r:
58         bot.send_message(call.message.chat.id, f"{s[1]} {s[0]}")
59     bot.send_message(call.message.chat.id, f"Знаменатель\n")
60     for s in zn_r:
61         bot.send_message(call.message.chat.id, f"{s[1]} {s[0]}")
62     markup2.add(telebot.types.InlineKeyboardButton(text='Понедельник' , callback_data = 11))
63     markup2.add(telebot.types.InlineKeyboardButton(text='Вторник' , callback_data = 12))
64     markup2.add(telebot.types.InlineKeyboardButton(text='Среда' , callback_data = 13))
65     markup2.add(telebot.types.InlineKeyboardButton(text='Четверг' , callback_data = 14))
66     markup2.add(telebot.types.InlineKeyboardButton(text='Пятница' , callback_data = 15))
67     markup2.add(telebot.types.InlineKeyboardButton(text='Суббота' , callback_data = 16))
68     answer = 'Выбери день недели'
69     bot.send_message(call.message.chat.id, answer, reply_markup=markup2)
70
71
72 bot.infinity_polling()

```

config.py:

```

1  from enum import Enum
2
3  # Токент бота
4  TOKEN = '5597125983:AAFcvSuyrWoSHXnNoJTntzYY3kNUd62ucvs'
5
6  # Файл базы данных Vedis
7  db_file = "db.vdb"
8
9  # Ключ записи в БД для текущего состояния
10 CURRENT_STATE = "CURRENT_STATE"
11
12 # Состояния автомата
13 class States(Enum):
14     STATE_START = "STATE_START" # Начало нового диалога
15     STATE_GROUP = "STATE_GROUP"
16     STATE_DAY = "STATE_DAY"

```

dbworker.py:

```

1  from vedis import Vedis
2  import config
3
4  # Чтение значения
5  def get(key):
6      with Vedis(config.db_file) as db:
7          try:
8              return db[key].decode()
9          except KeyError:
10             # в случае ошибки значение по умолчанию - начало диалога
11             return config.States.STATE_START.value
12
13
14 # Запись значения
15 def set(key, value):
16     with Vedis(config.db_file) as db:
17         try:
18             db[key] = value
19             return True
20         except:
21             # тут желательно как-то обработать ситуацию
22             return False
23
24
25 # Создание ключа для записи и чтения
26 def make_key(chatid, keyid):
27     res = str(chatid) + '__' + str(keyid)
28     return res

```

parser_1.py(парсер расписания с lks):

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 def load_schedule_data(session, group):
5     url = ''
6     if group == 1:
7         url = 'https://lks.bmstu.ru/schedule/a98e624a-81ca-11eb-9739-005056960017'
8     elif group == 2:
9         url = 'https://lks.bmstu.ru/schedule/a990b73e-81ca-11eb-bb1b-005056960017'
10    elif group == 3:
11        url = 'https://lks.bmstu.ru/schedule/a98e3af4-81ca-11eb-9f8f-005056960017'
12    elif group == 4:
13        url = 'https://lks.bmstu.ru/schedule/a98ddaaa-81ca-11eb-a4e2-005056960017'
14    else:
15        url = 'https://lks.bmstu.ru/schedule/a9a1cbe6-81ca-11eb-b2e0-005056960017'
16    request = session.get(url)
17    return request.text
18
19 def read_file(filename):
20     with open(filename) as input_file:
21         text = input_file.read()
22     return text
23
24
25 def parse_user_datafile_bs(filename, group):
26     s = requests.Session()
27     s.headers.update({
28         'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0) Gecko/20100101 Firefox/106.0'
29     })
30     data = load_schedule_data(s, group)
31
32     with open('./page.html', 'w', encoding='utf-8') as output_file:
33         output_file.write(data)
34
35     days = []
36     text = read_file(filename)
37     soup = BeautifulSoup(text)
38     days_list = soup.find_all('div', {'class': "col-lg-6 d-none d-md-block"})
39     for day in days_list:
40
41         subj_list = day.find_all('tr')
42         day_tmp = []
43
44         for i in range(2, len(subj_list)):
45
46             sb_tmp = dict()
47
48             if subj_list[i].find('td', {'class': "text-info-bold"}) and subj_list[i].find('td', {'class': "text-info-bold"}).text != ' ':
49                 sb_tmp['chisl'] = subj_list[i].find('td', {'class': "text-info-bold"}).text
50             if subj_list[i].find('td', {'class': "text-primary"}) and subj_list[i].find('td', {'class': "text-primary"}).text != ' ':
51                 sb_tmp['znam'] = subj_list[i].find('td', {'class': "text-primary"}).text
52             if subj_list[i].find('td', {'colspan': "2"}):
53                 sb_tmp['gen'] = subj_list[i].find('td', {'colspan': "2"}).text
54             if len(sb_tmp) != 0:
55                 sb_tmp['time'] = subj_list[i].find('td', {'class': "bg-grey text-nowrap"}).text
56                 day_tmp.append(sb_tmp)
57
58         days.append(day_tmp)
59
60     return days
```

state_bot.py:(бот на конечных автоматах)

```

1 import telebot
2 from telebot import types
3 import config
4 import dbworker
5 from parser_1 import parse_user_datafile_bs
6
7
8 # Токент бота
9 TOKEN = '5597125983:AAFcvSuYrWoSHXnNoJTntzYY3kNUd62ucvs'
10
11 # Создание бота
12 bot = telebot.TeleBot(TOKEN)
13
14
15 # Начало диалога
16 @bot.message_handler(commands=['start'])
17 def cmd_start(message):
18     bot.send_sticker(message.chat.id, 'CAACAgIAAxkBAAEGfQRje0lu-9ziLvJJI2c1E4CRzYLB3AACLBUAAoGBWUtbo0Y-udNhQSSe')
19     dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_GROUP.value)
20     markup = types.InlineKeyboardMarkup()
21     markup.add(telebot.types.InlineKeyboardButton(text='1я группа', callback_data = 1))
22     markup.add(telebot.types.InlineKeyboardButton(text='2я группа', callback_data = 2))
23     markup.add(telebot.types.InlineKeyboardButton(text='3я группа', callback_data = 3))
24     markup.add(telebot.types.InlineKeyboardButton(text='4я группа', callback_data = 4))
25     markup.add(telebot.types.InlineKeyboardButton(text='5я группа', callback_data = 5))
26     bot.send_message(message.chat.id, 'Приветтики 2й курс ИУ5, выбери номер группы, чтобы глянуть расписание', reply_markup=markup)
27
28
29 # По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога
30 @bot.message_handler(commands=['reset'])
31 def cmd_reset(message):
32     bot.send_message(message.chat.id, 'Сбрасываем результаты предыдущего ввода.')
33     dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_GROUP.value)
34     markup = types.InlineKeyboardMarkup()
35     markup.add(telebot.types.InlineKeyboardButton(text='1я группа', callback_data = 1))
36     markup.add(telebot.types.InlineKeyboardButton(text='2я группа', callback_data = 2))
37     markup.add(telebot.types.InlineKeyboardButton(text='3я группа', callback_data = 3))
38     markup.add(telebot.types.InlineKeyboardButton(text='4я группа', callback_data = 4))
39     markup.add(telebot.types.InlineKeyboardButton(text='5я группа', callback_data = 5))
40     bot.send_message(message.chat.id, 'Приветтики 2й курс ИУ5, выбери номер группы, чтобы глянуть расписание', reply_markup=markup)
41
42
43 # Обработка номера группы
44 @bot.callback_query_handler(func=lambda call: dbworker.get(dbworker.make_key(call.message.chat.id, config.CURRENT_STATE)) == config.States.STATE_GROUP.value)
45 def group(call):
46     markup2 = types.InlineKeyboardMarkup()
47     # Меняем текущее состояние
48     dbworker.set(dbworker.make_key(call.message.chat.id, config.CURRENT_STATE), config.States.STATE_DAY.value)
49     # Сохраняем первое число
50     dbworker.set(dbworker.make_key(call.message.chat.id, config.States.STATE_GROUP.value), call.data)
51     markup2.add(telebot.types.InlineKeyboardButton(text='Понедельник', callback_data = 11))
52     markup2.add(telebot.types.InlineKeyboardButton(text='Вторник', callback_data = 12))
53     markup2.add(telebot.types.InlineKeyboardButton(text='Среда', callback_data = 13))
54     markup2.add(telebot.types.InlineKeyboardButton(text='Четверг', callback_data = 14))
55     markup2.add(telebot.types.InlineKeyboardButton(text='Пятница', callback_data = 15))
56     markup2.add(telebot.types.InlineKeyboardButton(text='Суббота', callback_data = 16))
57     markup2.add(telebot.types.InlineKeyboardButton(text='Вернуться к выбору группы', callback_data = 100))
58     bot.send_message(call.message.chat.id, 'Выберите день недели', reply_markup=markup2)
59
60
61 # Обработка второго числа
62 @bot.callback_query_handler(func=lambda call: dbworker.get(dbworker.make_key(call.message.chat.id, config.CURRENT_STATE)) == config.States.STATE_DAY.value)
63 def second_num(call):
64     # Меняем текущее состояние
65
66     if call.data == '100':
67         print("-----")
68         bot.send_sticker(call.message.chat.id, 'CAACAgIAAxkBAAEGfQRje0lu-9ziLvJJI2c1E4CRzYLB3AACLBUAAoGBWUtbo0Y-udNhQSSe')
69         dbworker.set(dbworker.make_key(call.message.chat.id, config.CURRENT_STATE), config.States.STATE_GROUP.value)
70         markup = types.InlineKeyboardMarkup()
71         markup.add(telebot.types.InlineKeyboardButton(text='1я группа', callback_data = 1))
72         markup.add(telebot.types.InlineKeyboardButton(text='2я группа', callback_data = 2))
73         markup.add(telebot.types.InlineKeyboardButton(text='3я группа', callback_data = 3))
74         markup.add(telebot.types.InlineKeyboardButton(text='4я группа', callback_data = 4))
75         markup.add(telebot.types.InlineKeyboardButton(text='5я группа', callback_data = 5))
76         bot.send_message(call.message.chat.id, 'Приветтики 2й курс ИУ5, выбери номер группы, чтобы глянуть расписание', reply_markup=markup)

```

```

107
108     bot.send_message(call.message.chat.id, f"Числитель\n")
109     for s in ch_r:
110         bot.send_message(call.message.chat.id, f"{s[1]} {s[0]}")
111     bot.send_message(call.message.chat.id, f"Знаменатель\n")
112     for s in zn_r:
113         bot.send_message(call.message.chat.id, f"{s[1]} {s[0]}")
114
115     markup2.add(telebot.types.InlineKeyboardButton(text='Понедельник' , callback_data = 11))
116     markup2.add(telebot.types.InlineKeyboardButton(text='Вторник' , callback_data = 12))
117     markup2.add(telebot.types.InlineKeyboardButton(text='Среда' , callback_data = 13))
118     markup2.add(telebot.types.InlineKeyboardButton(text='Четверг' , callback_data = 14))
119     markup2.add(telebot.types.InlineKeyboardButton(text='Пятница' , callback_data = 15))
120     markup2.add(telebot.types.InlineKeyboardButton(text='Суббота' , callback_data = 16))
121     markup2.add(telebot.types.InlineKeyboardButton(text='Вернуться к выбору группы' , callback_data = 100))
122     answer = 'Выбери день недели'
123     bot.send_message(call.message.chat.id, answer, reply_markup=markup2)
124
125 if __name__ == '__main__':
126     bot.infinity_polling()

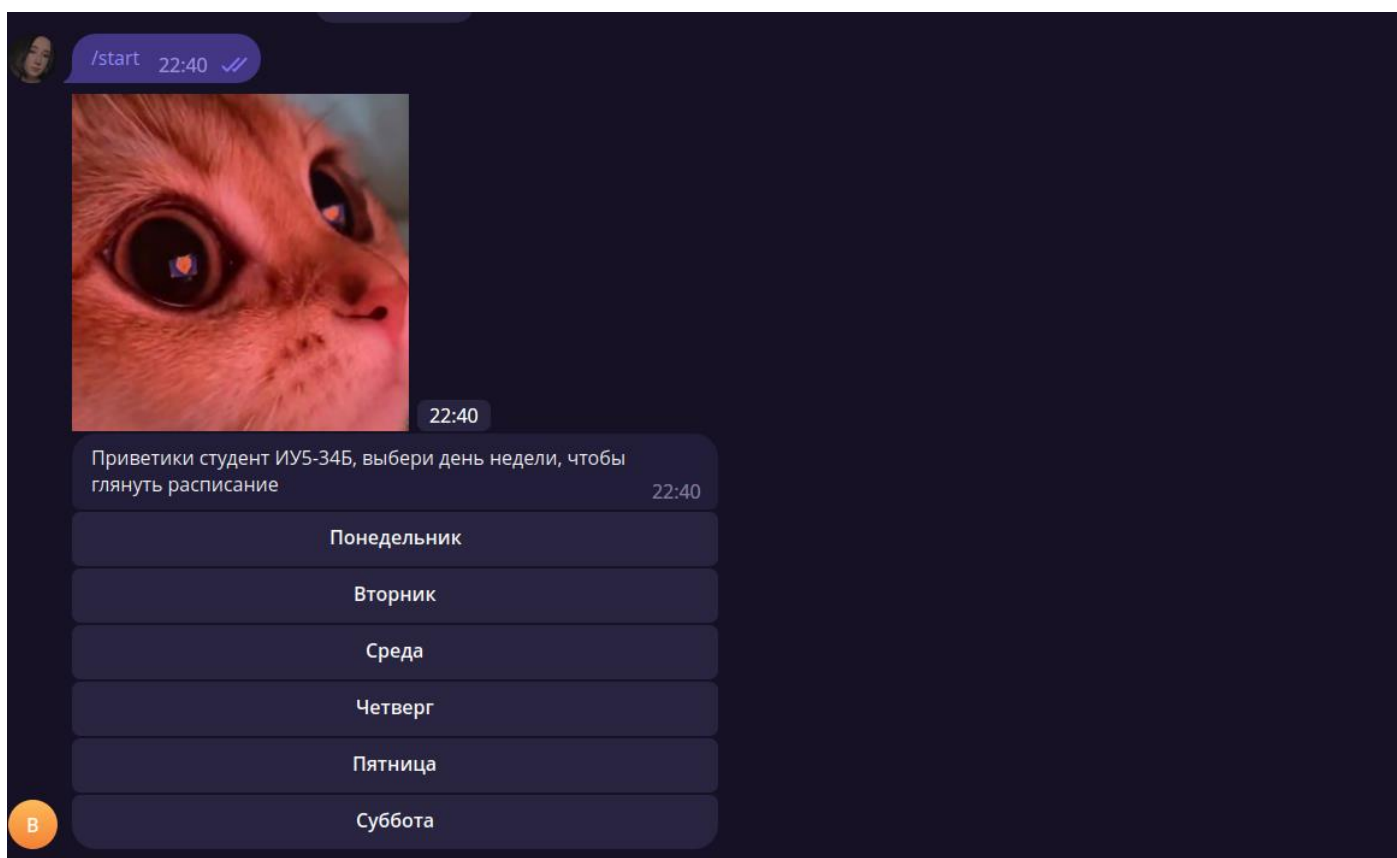
```

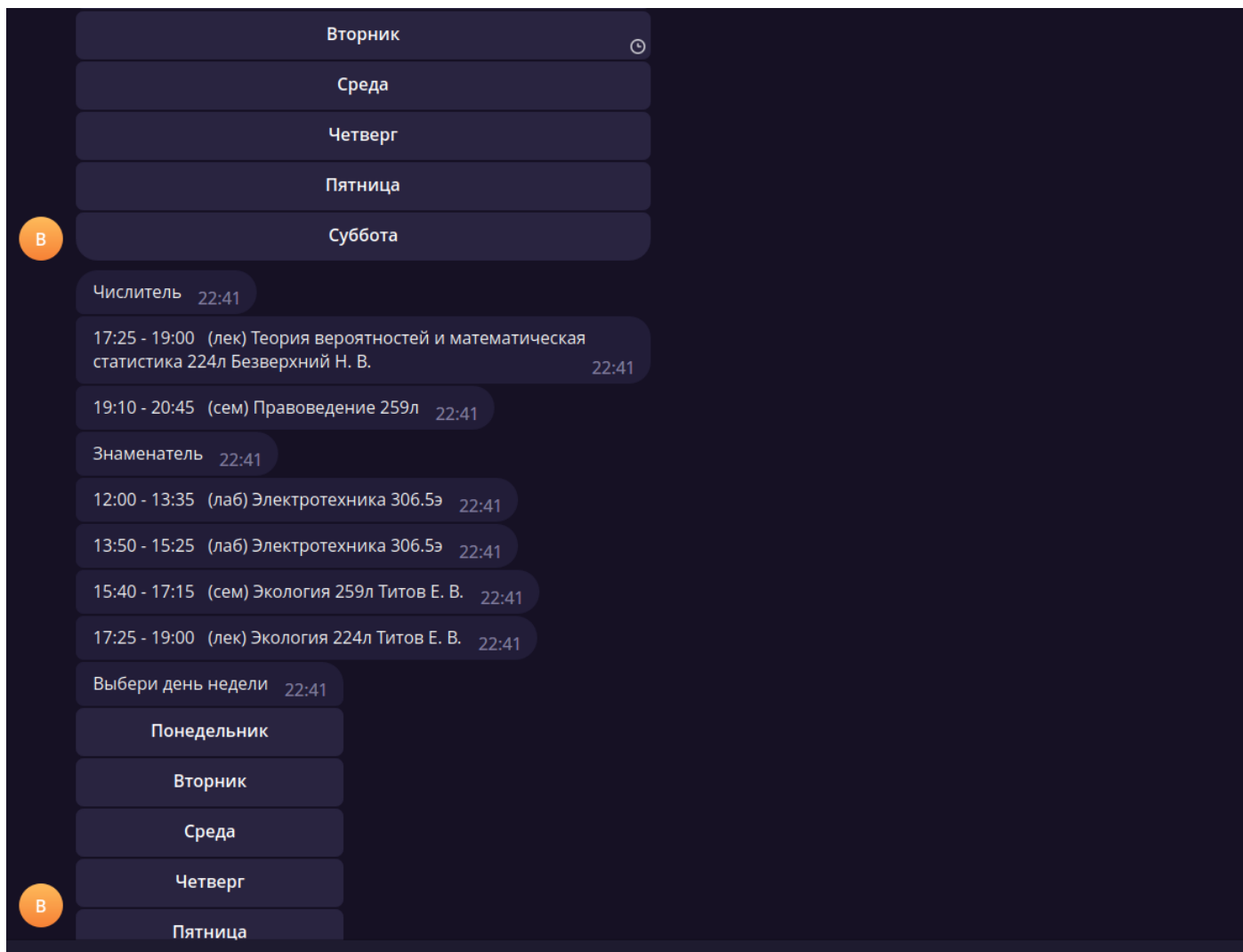
```

97         res1.append(s[key])
98     elif key == 'znam':
99         res2.append(s[key])
100     else:
101         res1.append(s[key])
102         res2.append(s[key])
103     if len(res1) > 1:
104         ch_r.append(res1)
105     if len(res2) > 1:
106         zn_r.append(res2)

```

Результаты работы программы бота(bot.py):





Результаты работы бота на конечных автоматах (state_bot.py):



/start 22:43 ✓

December 19



Приветик 2й курс ИУ5, выбери номер группы, чтобы глянуть расписание

22:43

1я группа



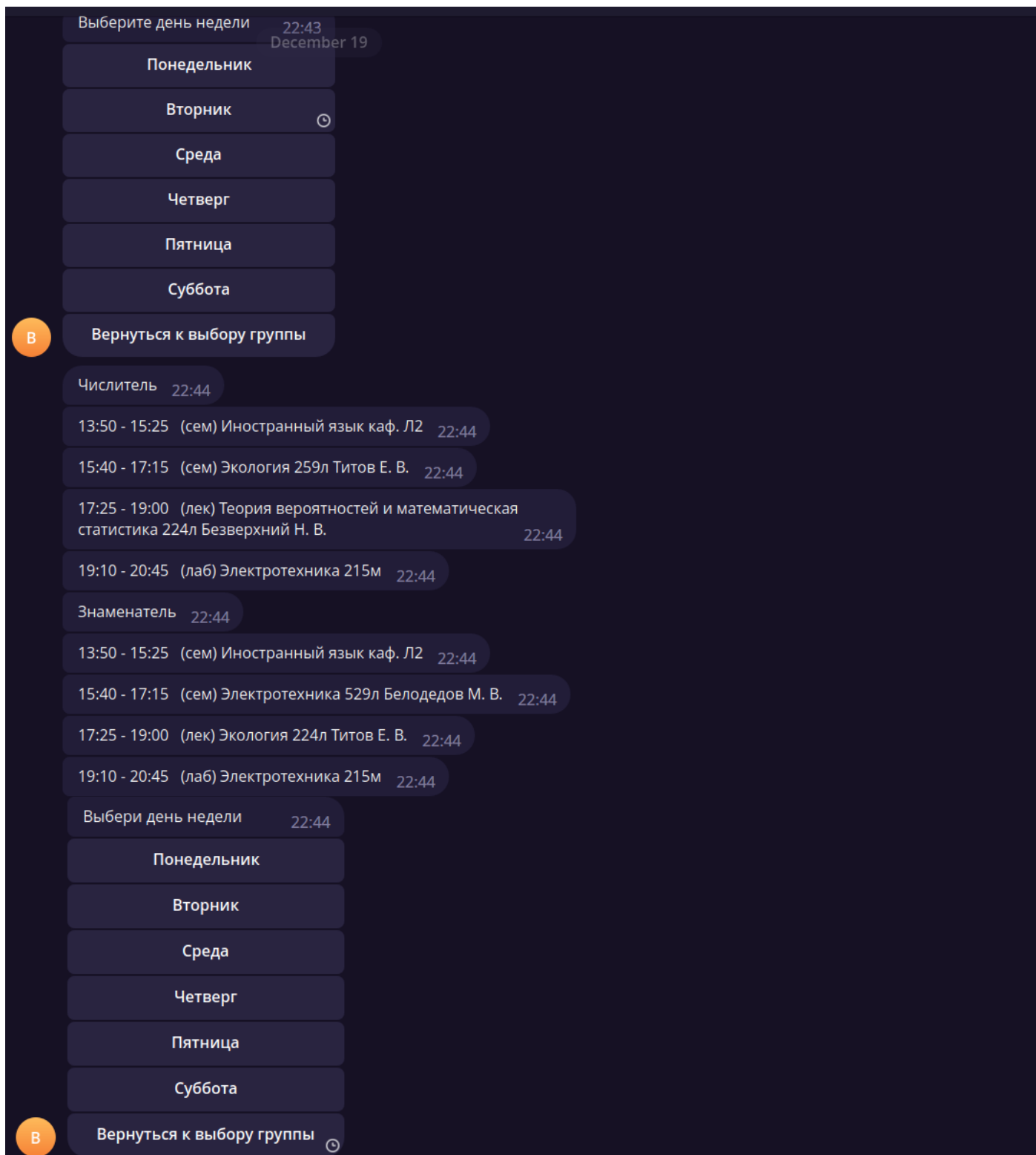
2я группа

3я группа

4я группа

5я группа

В



При нажатии «Вернуться к выбору группы», бот возвращается к состоянию STATE_GROUP



22:44

Приветтики 2й курс ИУБ, выбери номер группы, чтобы глянуть расписание

22:44

1я группа

2я группа

3я группа

4я группа

5я группа

В