**Модуль 4**

**Урок 1. Работа с файлами и форматированный вывод**

**Форматирование строк, вывод на консоль, работа с таблицами Excel**

OpenPyXL — это библиотека, используемая для чтения и записи файлов Excel 2010 xlsx/xlsm/xltx/xltm, которую мы будем использовать в этом занятиии для работы с документами Excel.

Первое, что нам нужно сделать, чтобы использовать эту библиотеку, это установить OpenPyXL .

**Установка OpenPyXL**

Для установки OpenPyXL мы будем использовать pip. OpenPyXL можно просто установить, введя следующую команду в терминале:

**pip install openpyx**

Открытие документа Excel

После установки OpenPyXL мы готовы начать работу с документами Excel. Первая обычная задача, которую мы выполняем для документа Excel, — это открыть этот документ. Загрузите файл Excel sample.xlsx , чтобы следовать руководству, или вы можете использовать любой файл Excel, который вам нравится.

Прежде чем мы сможем использовать OpenPyXL, нам нужно импортировать его следующим образом:

import openpyxl

Для открытия документа Excel нам нужен метод load\_workbook() . Если вам интересно, что подразумевается под книгой, это просто документ электронной таблицы Excel. Сценарий, который нам нужен для открытия документа Excel, выглядит следующим образом:

import openpyxl

excel\_document = openpyxl.load\_workbook(‘sample.xlsx’)

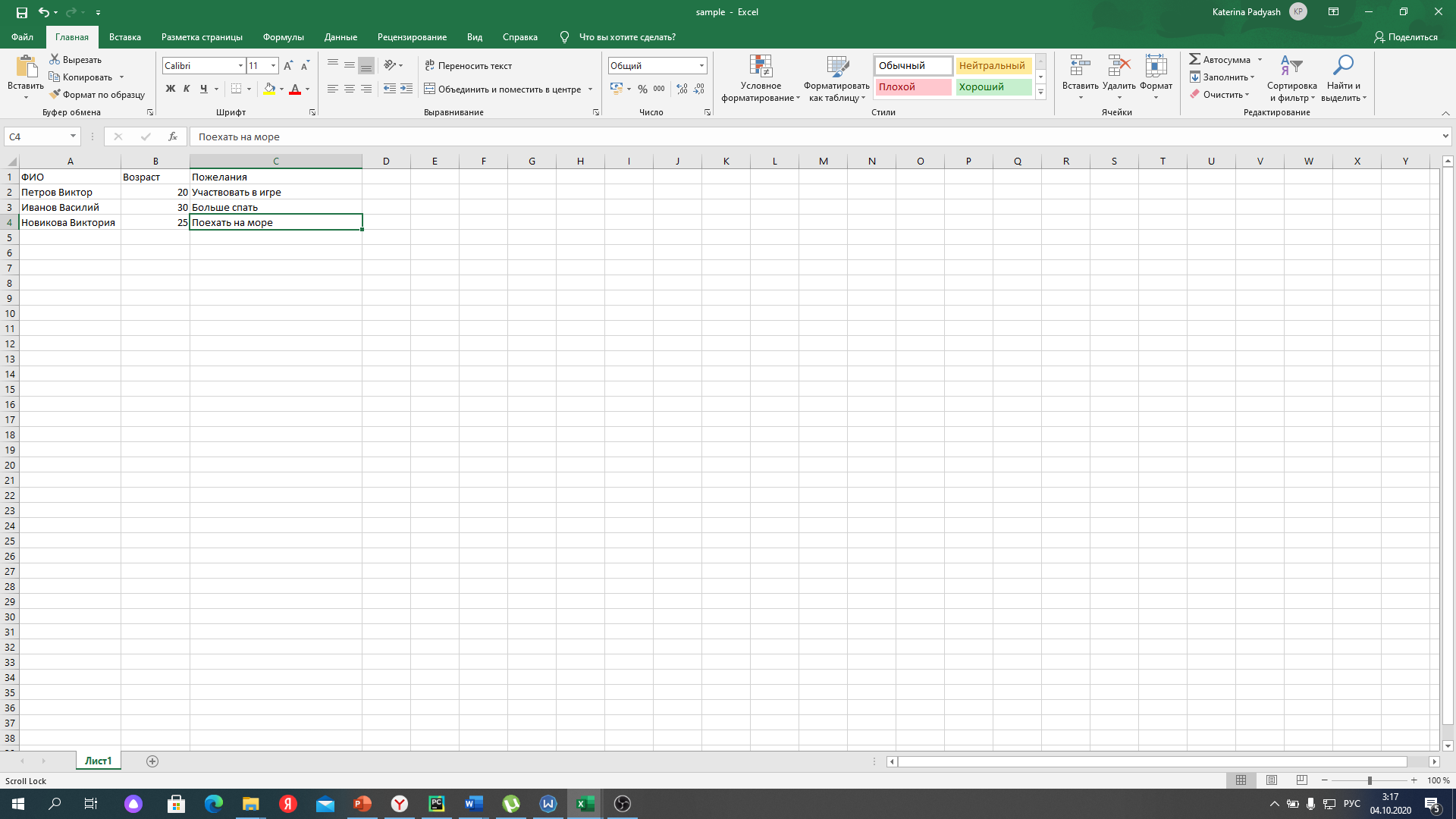
Давайте теперь посмотрим type возвращаемый из load\_workbook() . Это можно сделать следующим образом:

print(type(excel\_document))

Это вернет следующее:

<class ‘openpyxl.workbook.workbook.Workbook’>

Как мы видим, возвращаемым объектом является Workbook , тип данных workbook . Объект Workbook здесь представляет файл Excel.



**Название листа**

Листы в Excel состоят из столбцов (буквы начинаются с A, B, C и т. Д.) И строк (начиная с 1, 2, 3 и т. Д.). В нашем примере у нас есть один лист, который называется Лист1.

**Доступ к клеткам**

Теперь, когда мы узнали, как открыть файл Excel и получить лист, давайте посмотрим, насколько просто получить доступ к ячейке на этом листе. Все, что вам нужно сделать, это получить этот лист, а затем определить местоположение (координату) ячейки. Допустим, мы хотим получить доступ к столбцу A строки 2 в имеющемся у нас документе Excel, то есть A2 . Это может быть реализовано следующим образом:

sheet = excel\_document.get\_sheet\_by\_name(‘Sheet1’)

print ( sheet[‘A2’].value

В этом случае вам будет возвращено следующее значение: Петров Василий .

Мы также можем использовать запись строки-столбца. Например, если мы хотим получить доступ к ячейке в строке 3 и столбце 2, мы набираем следующее:

sheet.cell(row = 3, column = 2).value

**Для нашего файла получим результат: 30**

Если мы хотим увидеть тип объекта, представляющий ячейку, мы можем напечатать:

print (type(sheet['A2']))

или:

print (sheet.cell(row = 3, column = 2))

В этом случае вы получите следующий вывод:

<class 'openpyxl.cell.cell.Cell'>

Это означает, что объект имеет тип Cell.

Доступ к ряду ячеек

Что если вы заинтересованы в доступе к диапазону ячеек, а не только к одной ячейке? Допустим, мы хотим получить доступ к ячейкам от A1 до B3 , как это выглядит в нашем документе Excel?

Это можно сделать с помощью следующего скрипта:

multiple\_cells = sheet[“A1”:”B3”]

for row in multiple\_cells:

for cell in row:

print (cell.value)

**В этом случае вы получите следующий результат:**

ФИО

Возраст

Пожелания

Петров Виктор

20

Участвовать в игре

Иванов Василий

30

Больше спать

**Доступ ко всем строкам и столбцам**

OpenPyXL позволяет вам получить доступ ко всем строкам и столбцам в вашем документе Excel, используя методы OpenPyXL rows() и columns() соответственно.

Строковый метод format() возвращает отформатированную версию строки, заменяя идентификаторы в фигурных скобках. Идентификаторы могут быть позиционными, числовыми индексами, ключами словарей, именами переменных. Аргументов в format() может быть больше, чем идентификаторов в строке. В таком случае оставшиеся игнорируются.

Вы можете использовать format(), чтобы выполнить простое позиционное форматирование:

Python

print('Hello, {}'.format(name))

# Вывод: 'Hello, Bob'

print('Hello, {}'.format(name))

# Вывод: 'Hello, Bob'

Или, вы можете сослаться на свои подстановки переменных по имени, и использовать их в том порядке, в котором вам хочется. Это достаточно мощный способ, так как он позволяет повторно упорядочить порядок отображения без изменения переданных функции format() аргументов:

print( 'Hey {name}, there is a 0x{errno:x} error!'.format(name=name, errno=errno))

# Вывод: 'Hey Bob, there is a 0xbadc0ffee error!'

Это также демонстрирует, что синтаксис формата переменной int — это шестнадцатеричная измененная строка. Теперь вам нужно передать формат spec, внеся суффикс :x. Синтаксис формата строки стал более сильным, не усложняя при этом более простые варианты использования. Не лишним будет ознакомиться с этим мини-языком форматирования строк в документации Python.

С другими методами форматирования можно ознакомиться в **дополнительных материалах:**

[**https://python-scripts.com/string-formatting**](https://python-scripts.com/string-formatting)

[**https://pythoner.name/formatted-output**](https://pythoner.name/formatted-output)

**Исходный код:**

import openpyxl

#Загружаем файл электронной таблицы из рабочего каталога

excel\_document = openpyxl.load\_workbook("sample.xlsx")

#Выводим на печать тип полученного документа

print(type(excel\_document))

#Считываем названия листов

lst=excel\_document.get\_sheet\_names()

print(lst)

#Считываем Лист1 - лист 0 в Пайтоне

sheet = excel\_document[lst[0]]

#Выводим на печать значения ячеек

multiple\_cells = sheet["A1":"C3"]

for row in multiple\_cells:

for cell in row:

print(cell.value, end="\t")

print("\n")

sheet['B1'] = "Возраст"

#Создать новый лист

sheet2 = excel\_document.create\_sheet("Лист2")

sheet2['A1'] = "{0} очень любит {1}".format(sheet["A2"].value, sheet["C2"].value)

print(sheet2['A1'].value)

# Сохранить документ

excel\_document.save("sample.xlsx")

**Задание:** Создать экселевский файл, в который в цикле ввести несколько строк данных о командах со следующими полями(столбцами):  
- ФИО

- Название команды

- Контактный телефон

- Название проекта

Продолжать ввод по подтверждению пользователя (y/n)

**Модуль 4**

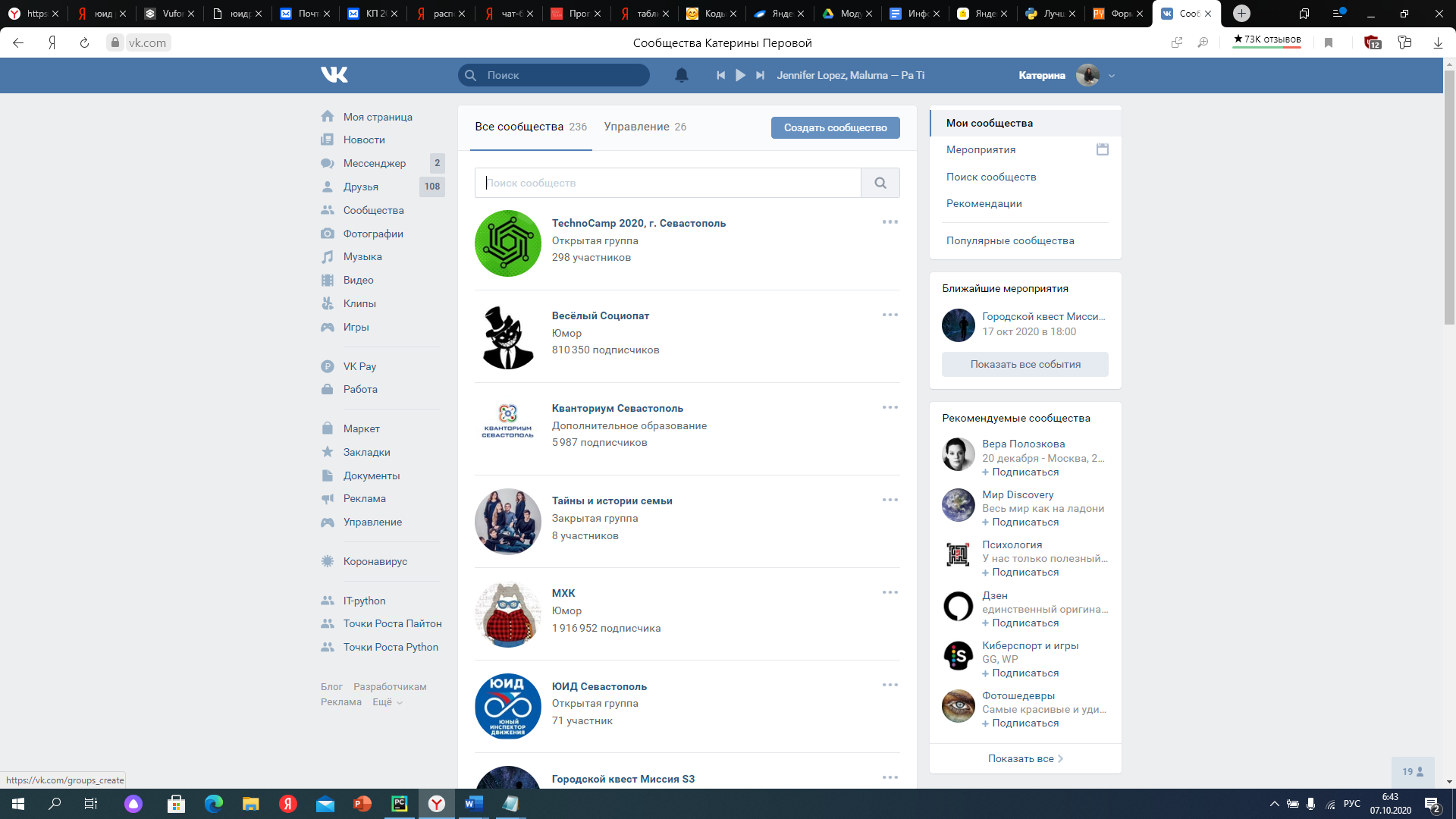
**Урок 2. Создание простого чат-бота в VK**

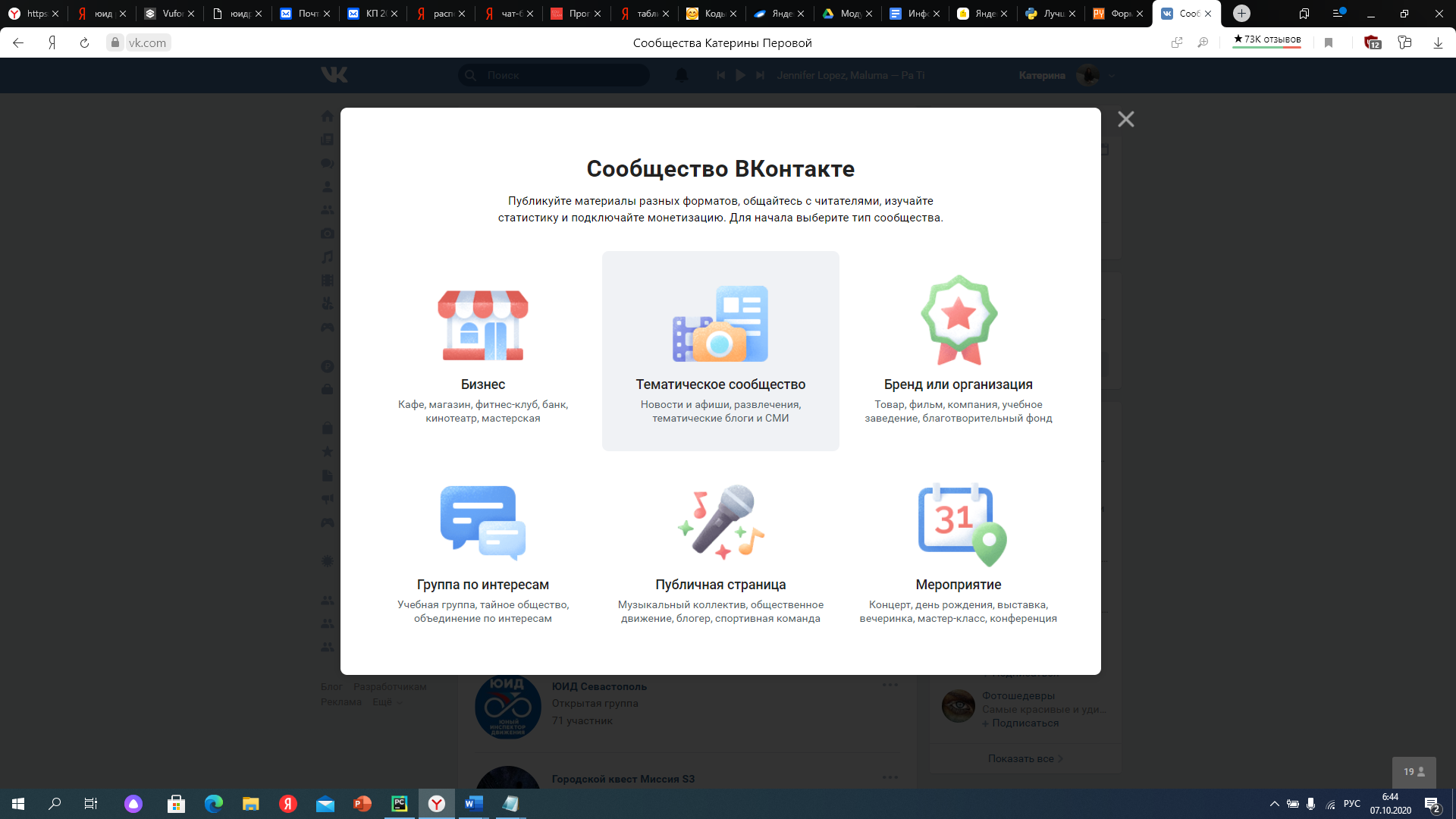
**Практическая проектная работа по применению полученных знаний**

Данная работа займет три последующих занятия и позволит вам подвести небольшой итог полученных знаний и применить их на практике. Для с работы с чат-ботами обязательно нужен мобильный телефон, потому что на него будут приходить СМС.

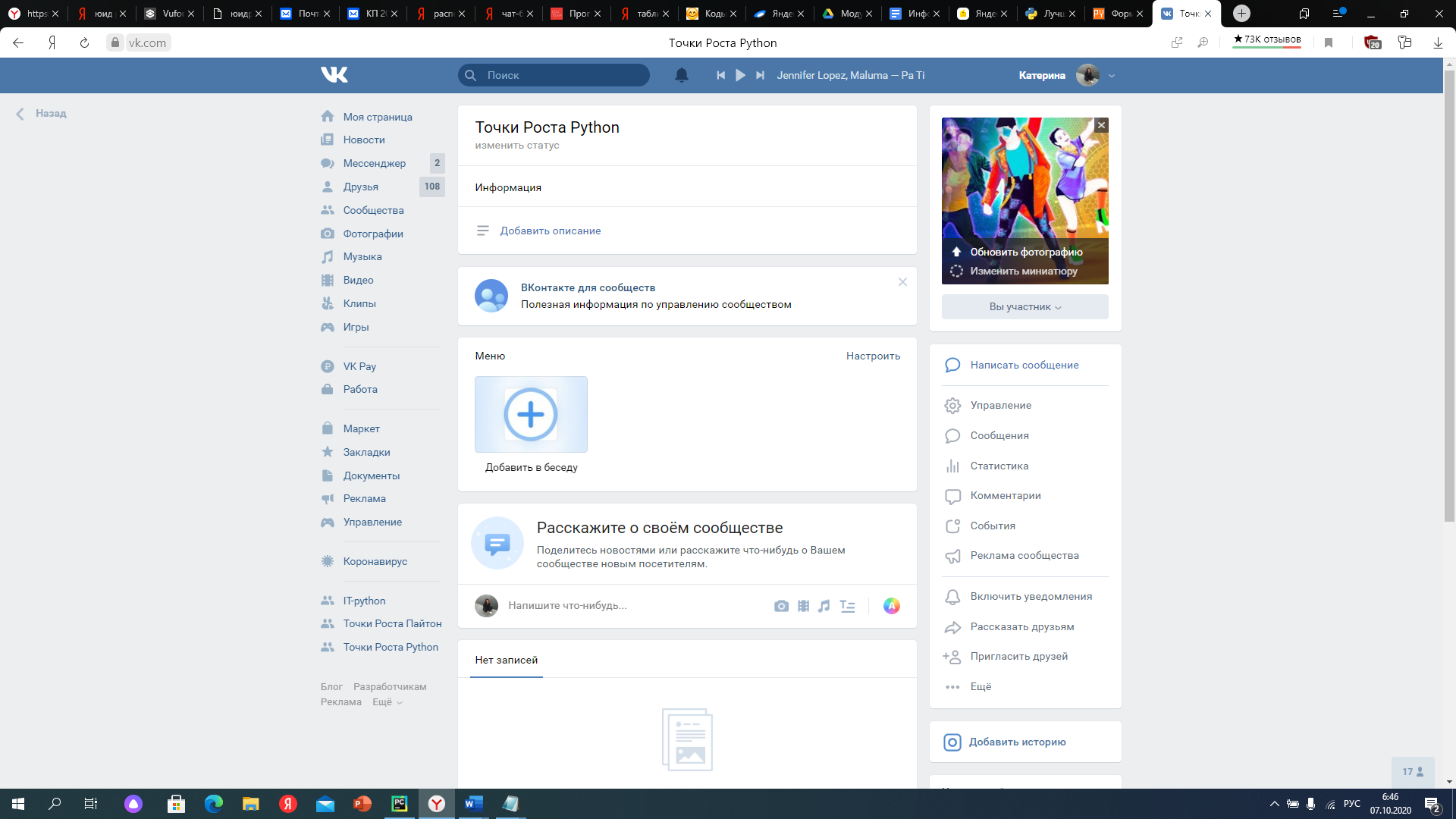
**Первый этап создания бота для социальной сети: Работа VКонтакте – создание группы, настройка сообщений и получение токена**

1. Создать сообщество VK, тему выбрать самостоятельно

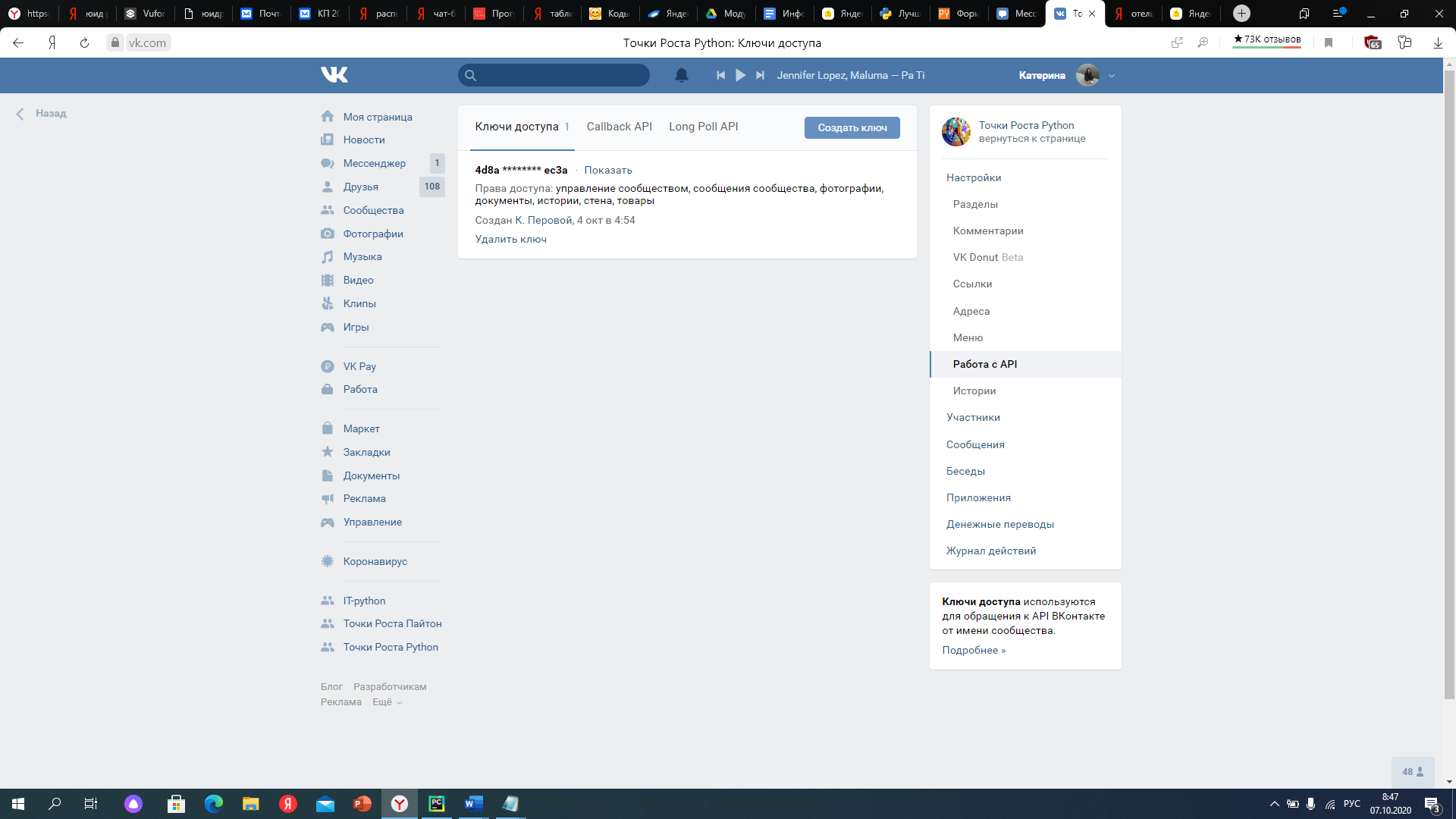




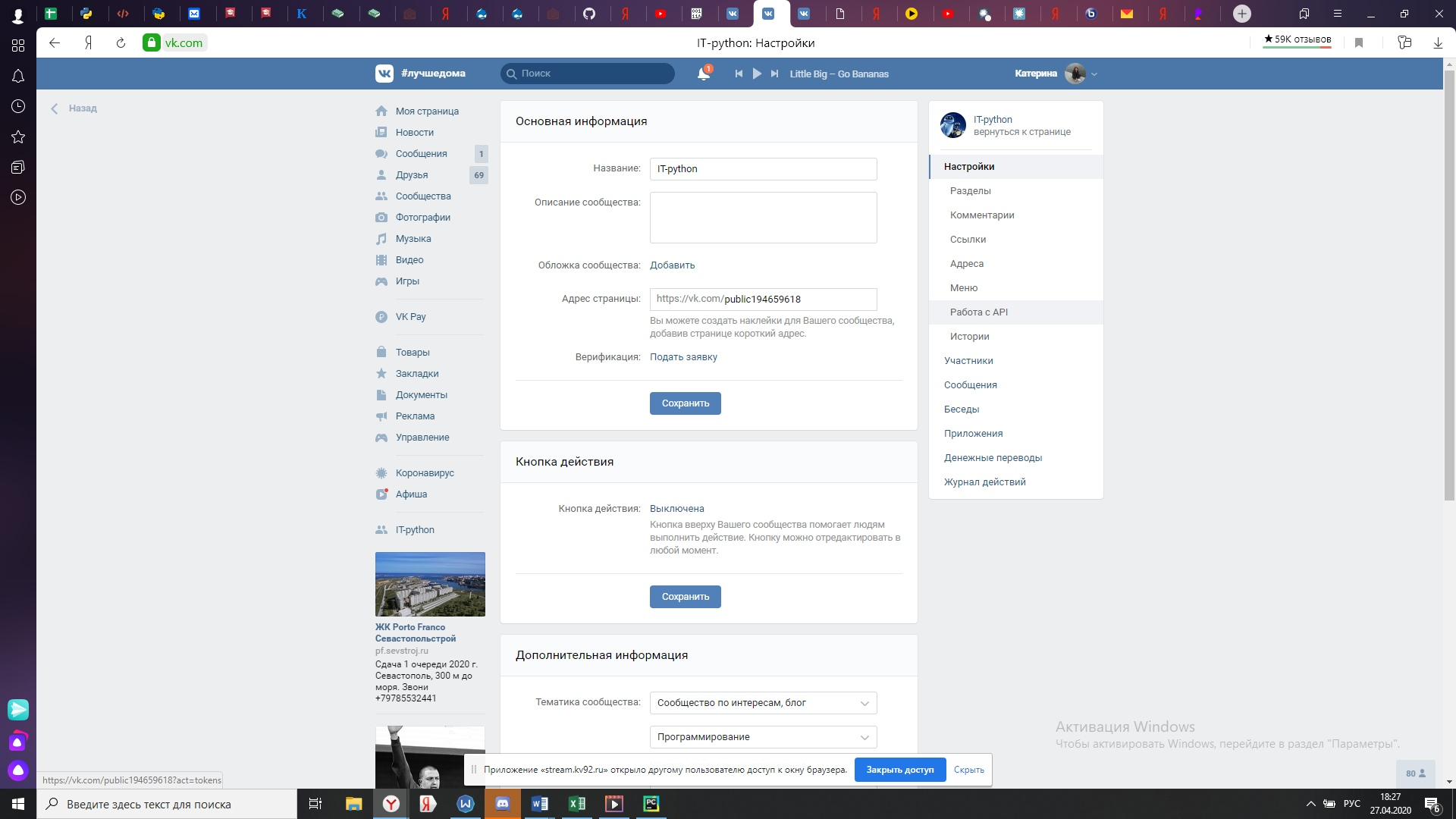
1. Придумать красивое название (поле «Сайт» можно не заполнять), затем заполнить картинку аватара сообщества.



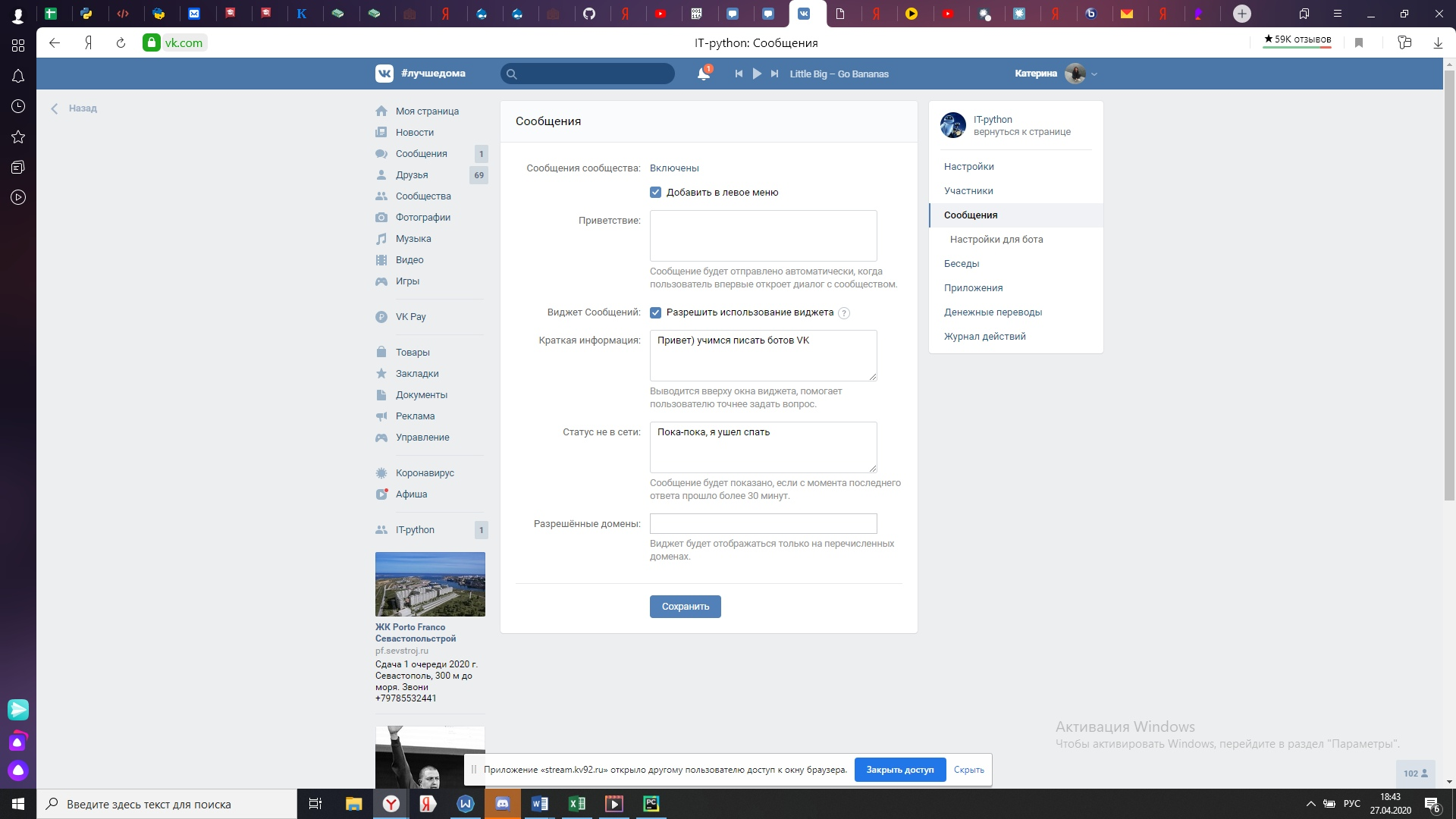
1. Зайти в пункт меню Управление - находим пункт "Работа с API"
2. Нажимаем кнопку «Создать ключ». В появившемся окошке разрешаем все, ставим все галочки. Подтверждаем номер телефона и сохраняем куда-нибудь ключ доступа. Теперь переходим во вкладку Callback API. Ставим свое название сервера и выбираем последнюю версию API.

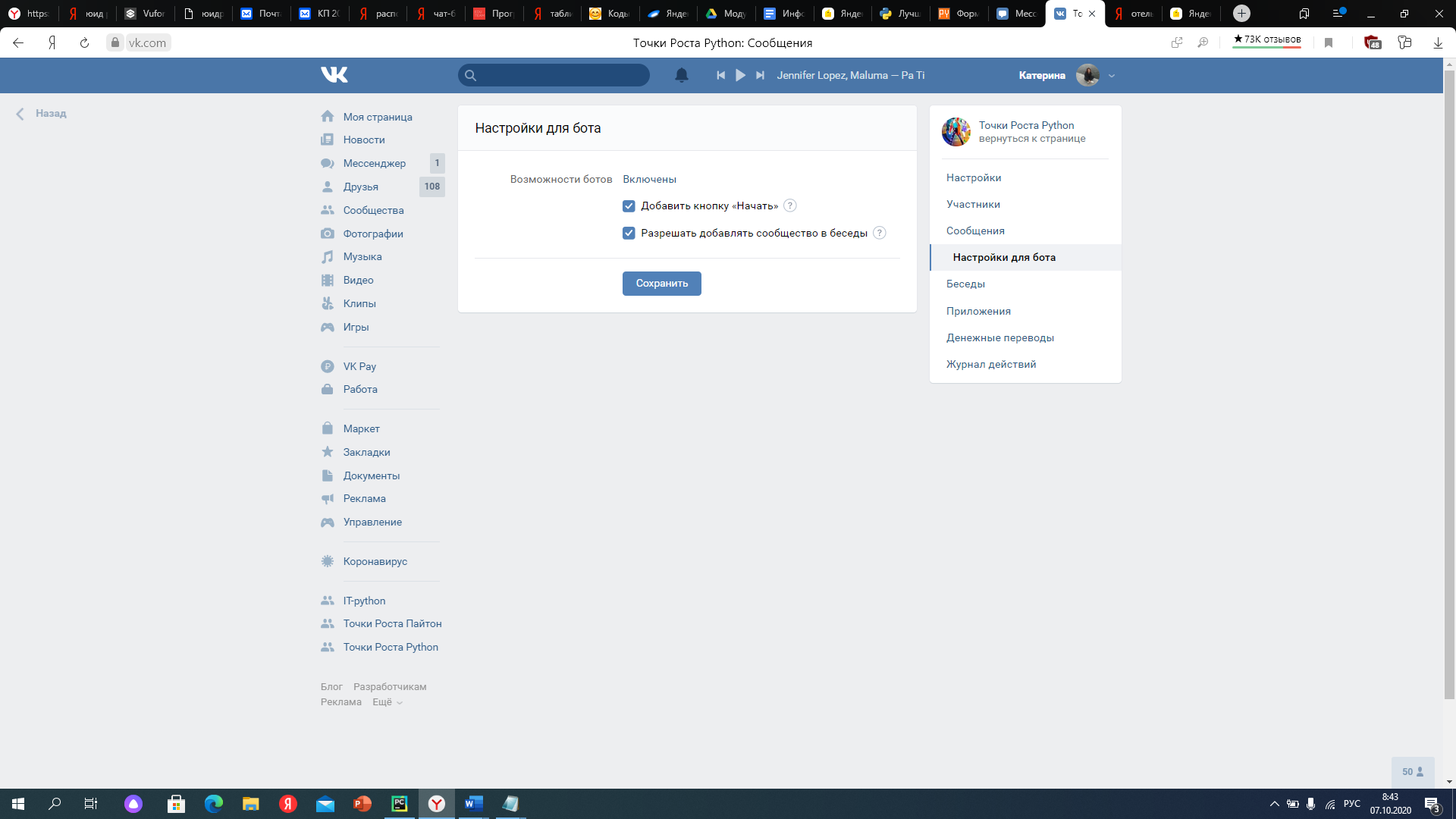


1. Пункт Сообщения - Сообщения от сообщества ставим: Включены и добавляем в левое меню.



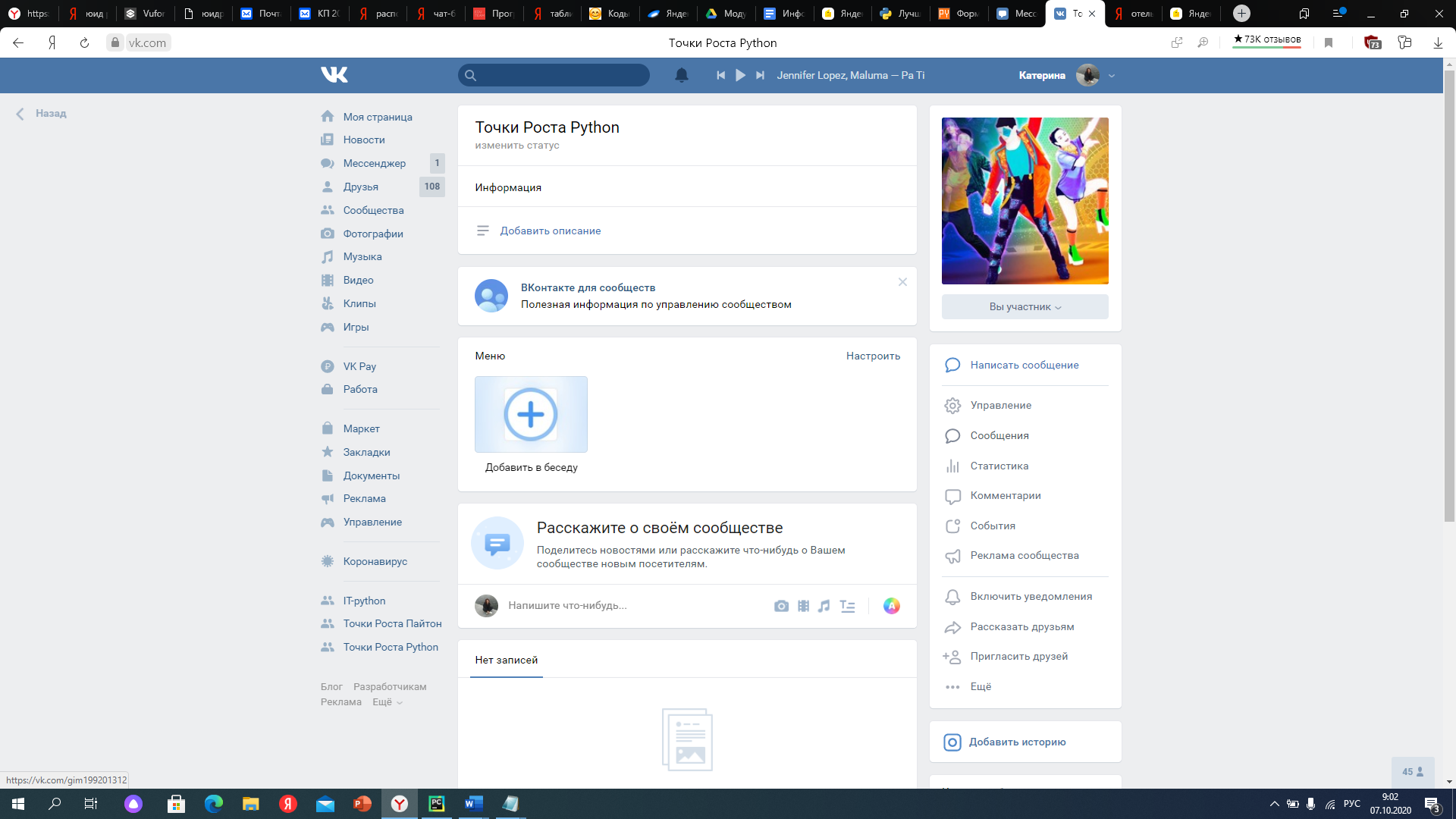
1. Заходим в Меню: Управление – Сообщения - Настройки для бота, ставим Включено и галочки в полях.





**Второй этап работы с кодом Python-PyCharm (или привычный вам редактор кода)**

1. Создать новый файл python
2. Установить библиотеку: vk\_api (pip install vk\_api)
3. Скопировать код программы
4. Вставить свой токен, полученный в VK
5. Запустить проверку работы чат-бота на исходном коде, чат бот должен ответить на сообщение «Привет» в течение секунды после запуска программы. Написать боту нужно в Сообщения созданного вами сообщества:



1. **Выполнить задание:** изменить слова и ответы (обучить бота своим словам и ответам)

**Исходный код:**

# -\*- coding: utf-8 -\*-

import vk\_api

import time

import random

import json

token = "ТУТ ДОЛЖЕН БЫТЬ ВАШ ТОКЕН"

vk = vk\_api.VkApi(token=token)

vk.\_auth\_token()

while True:

try:

messages = vk.method("messages.getConversations", {"offset": 0, "count": 20, "filter": "unanswered"})

if messages["count"] >= 1:

id = messages["items"][0]["last\_message"]["from\_id"]

body = messages["items"][0]["last\_message"]["text"]

if body.lower() == "привет":

vk.method("messages.send",

{"peer\_id": id, "message": "Добрый день! Как дела?", "random\_id": random.randint(1, 2147483647)})

elif body.lower() == "хорошо":

vk.method("messages.send",

{"peer\_id": id, "message": "Очень рад, вот тебе мешок золота!", "random\_id": random.randint(1, 2147483647)})

elif body.lower() == "норм":

vk.method("messages.send",

{"peer\_id": id, "message": "Желаю тебе прекрасного времени суток",

"random\_id": random.randint(1, 2147483647)})

elif body.lower() == "люблю":

vk.method("messages.send",

{"peer\_id": id, "message": "https://justdancenow.com/",

"random\_id": random.randint(1, 2147483647)})

elif body.lower() == "не люблю":

vk.method("messages.send",

{"peer\_id": id, "message": "Ненавижу манную кашу с комочками!",

"random\_id": random.randint(1, 2147483647)})

elif body.lower() == "важно":

vk.method("messages.send",

{"peer\_id": id, "message": "Сиди дома!",

"random\_id": random.randint(1, 2147483647)})

elif body.lower() == "неважно":

vk.method("messages.send",

{"peer\_id": id, "message": "Сверхъестественное - отличный сериал)",

"random\_id": random.randint(1, 2147483647)})

elif body.lower() == "плохо":

vk.method("messages.send",

{"peer\_id": id, "message": "Не грусти, хочешь песенку спою?",

"random\_id": random.randint(1, 2147483647)})

else:

vk.method("messages.send",

{"peer\_id": id, "message": "Не понял тебя!", "random\_id": random.randint(1, 2147483647)})

except Exception as E:

time.sleep(1)

**Модуль 4**

**Урок 3. Клавиатура для чат-бота в VK**

**Использования объекта json для создания интересного вывода нужной информации.**

На этом занятии мы продолжаем работать с нашим ботом и добавим в его ответы небольшое меню, в VK она называется «клавиатура» keyboard. Для создания клавиатуры нам понадобится объект json. Данный объект, по сути, выглядит, как словарь, который содержит кортежи и списки.

Тем не менее, JSON (англ. JavaScript Object Notation, обычно произносится как /ˈdʒeɪsən/ JAY-sən ) — это текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается независимым от языка и может использоваться практически с любым языком программирования.

Для Python существует встроенная библиотека для создания и обработки данных в формате JSON.

Структура клавиатуры:

1. Поле **"one\_time"** может принимать значения: False – если клавиатуру нужно повторять до тех пор, пока ее не заменят пустым или другим объектом/ True - если клавиатуру нужно вывести один раз
2. Поле **"buttons"**: сами кнопки клавиатуры в нужно формате.

Для того, чтобы убрать меню – нужно отправить в сообщениях пустой объект: {"buttons":[],"one\_time":true}

**Исходный код:**

import vk\_api  
import time  
import random  
import json  
  
token = **"ТУТ ДОЛЖЕН БЫТЬ ВАШ ТОКЕН"**

#Создание и авторизация объекта Vk\_Api

vk = vk\_api.VkApi(token = token)  
vk.\_auth\_token()

#Создание json-объекта клавиатуры

keyboard = {  
**"one\_time"** : False,  
 **"buttons"**: [  
 [{  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**1**\"**}"**,  
 **"label"**: **"Не люблю"** },  
 **"color"**: **"negative"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**2**\"**}"**,  
 **"label"**: **"Люблю"** },  
 **"color"**: **"positive"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**3**\"**}"**,  
 **"label"**: **"Важно"** },  
 **"color"**: **"primary"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**4**\"**}"**,  
 **"label"**: **"Неважно"** },  
 **"color"**: **"secondary"** }  
 ]  
 ]  
}  
#Проверка форматирования клавиатуры  
  
keyboard = json.dumps(keyboard, ensure\_ascii=False).encode(**'utf-8'**)  
keyboard = str(keyboard.decode(**'utf-8'**))

while True:  
 try:

#Проверяем получение сообщений, если сообщений больше одного – считываем последнее.   
 messages = vk.method(**"messages.getConversations"**, {**"offset"**: 0, **"count"**: 20, **"filter"**: **"unanswered"**})  
 if messages[**"count"**] >= 1:  
 id = messages[**"items"**][0][**"last\_message"**][**"from\_id"**]  
 body = messages[**"items"**][0][**"last\_message"**][**"text"**]  
 if body.lower() == **"кнопки"** or oldmessage == **"кнопки"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Выбери кнопку"**, **"keyboard"**: keyboard, **"random\_id"**: random.randint(1, 2147483647)})  
 if body.lower() == **"люблю"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"https://justdancenow.com/"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"не люблю"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Ненавижу манную кашу с комочками!"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"важно"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Сиди дома!"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"неважно"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Сверхъестественное - отличный сериал)"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"привет"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Привет! Как дела?"**, **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"хорошо"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Очень рад, вот тебе мешок золота!"**, **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"норм"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Желаю тебе прекрасного времени суток"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
  
 elif body.lower() == **"плохо"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Не грусти, хочешь песенку спою?"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 else:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Не понял тебя!"**, **"random\_id"**: random.randint(1, 2147483647)})  
 oldmessage = body.lower()  
 except Exception as E:  
 time.sleep(1)

Не стоит использовать в занятиях листинг исходников представленных в курсе, они использованы лишь для того чтобы вы лучше понимали что происходит. Разумеется, они все рабочие и можно использовать их по частям.

**Задание: добавить к боту еще одну клавиатуру, которая будет** появляться по запросу пользователя (выбрать на своем усмотрение). **Разбить кнопки клавиатуры на 2-3 уровня (регулируем квадратными и фигурными скобками).**

**Дополнительные материалы по Vk\_api:**

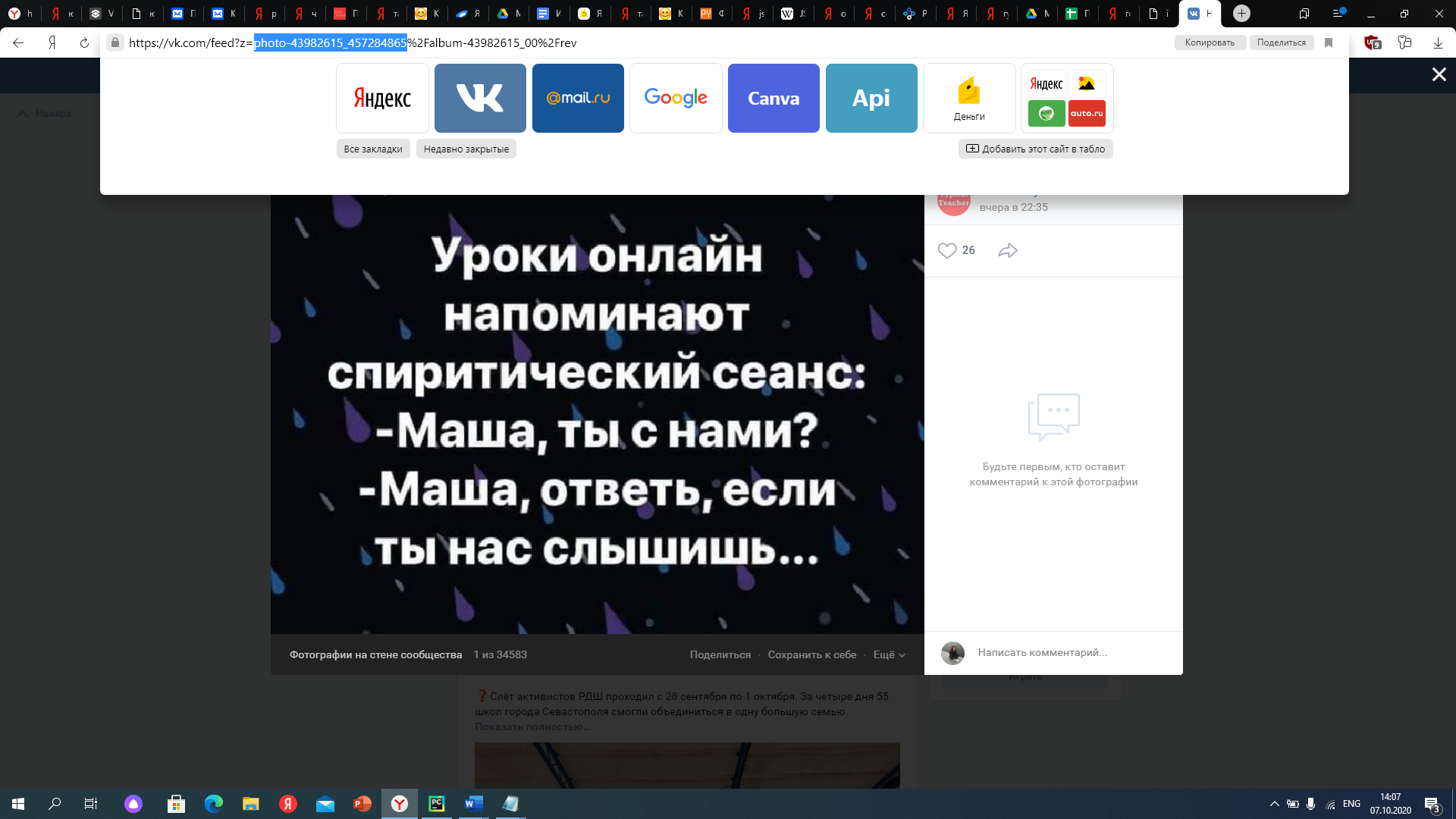
[**https://vk.com/dev/bots\_docs\_3**](https://vk.com/dev/bots_docs_3)

**Модуль 4**

**Урок 4. Медиа-сообщения и иконки для чат-бота в VK**

**Использования дополнительных возможностей для работы с чат-ботом.**

Для работы с медийными сообщениями в контакте вам понадобиться найти уникальный идентификатор медиа-файла в адресной строке вашего браузера.



Также в данном занятии вам понадобится ссылка на таблицу работы с эмоджи: <https://kody-smajlov-vkontakte.ru/>

Все остальные понятия уже должны быть вам знакомы. Если какой-то этап покажется сложным и бот перестанет работать – попробуйте вернуться к более старой проверенной версии кода и затем постепенно усложнять, проверяя работу бота на каждом этапе.

import vk\_api  
import time  
import random  
import json  
  
token = **"Ваш токен"**vk = vk\_api.VkApi(token = token)  
  
vk.\_auth\_token()

#Стартовое меню  
keyboard0 = {  
 **"one\_time"**: False,  
 **"buttons"**: [  
 [{  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**1**\"**}"**,  
 **"label"**: **"Регистрация"** },  
 **"color"**: **"primary"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**2**\"**}"**,  
 **"label"**: **"Новости"** },  
 **"color"**: **"secondary"** }  
 ]  
 ]  
}  
  
#Меню регистрации  
  
keyboard = {  
**"one\_time"**: False,  
 **"buttons"**: [  
 [{  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**1**\"**}"**,  
 **"label"**: **"ФИО"** },  
 **"color"**: **"negative"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**2**\"**}"**,  
 **"label"**: **"Телефон"** },  
 **"color"**: **"positive"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**3**\"**}"**,  
 **"label"**: **"Команда"** },  
 **"color"**: **"primary"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**4**\"**}"**,  
 **"label"**: **"Проект"** },  
 **"color"**: **"secondary"** }  
 ]  
 ]  
}

#Меню новостей и информации  
  
keyboard1 = {  
**"one\_time"**: False,  
 **"buttons"**: [  
 [ {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**1**\"**}"**,  
 **"label"**: **"&#128198; Новости "** },  
 **"color"**: **"negative"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**2**\"**}"**,  
 **"label"**: **"&#127917; Мероприятия"** },  
 **"color"**: **"positive"** }],  
 [{  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**3**\"**}"**,  
 **"label"**: **"&#128182; Наш сайт"** },  
 **"color"**: **"primary"** },  
 {  
 **"action"**: {  
 **"type"**: **"text"**,  
 **"payload"**: **"{**\"**button**\"**:** \"**4**\"**}"**,  
 **"label"**: **"&#128182; Информация"** },  
 **"color"**: **"secondary"** }  
 ]  
 ]  
}  
#Формирование корректных строк для передачи в функцию message.  
keyboard = json.dumps(keyboard, ensure\_ascii=False).encode(**'utf-8'**)  
keyboard = str(keyboard.decode(**'utf-8'**))  
  
keyboard0 = json.dumps(keyboard0, ensure\_ascii=False).encode(**'utf-8'**)  
keyboard0 = str(keyboard0.decode(**'utf-8'**))  
  
keyboard1 = json.dumps(keyboard1, ensure\_ascii=False).encode(**'utf-8'**)  
keyboard1 = str(keyboard1.decode(**'utf-8'**))  
  
  
lst = [**"регистрация"**,**"фио"**, **"телефон"**, **"команда"**, **"проект"**]  
oldmessage = **''**name=**'незнакомец'**while True:  
 try:  
 messages = vk.method(**"messages.getConversations"**, {**"offset"**: 0, **"count"**: 20, **"filter"**: **"unanswered"**})  
 if messages[**"count"**] >= 1:  
 id = messages[**"items"**][0][**"last\_message"**][**"from\_id"**]  
 body = messages[**"items"**][0][**"last\_message"**][**"text"**]  
 if body.lower() == **"регистрация"** or (oldmessage in lst):  
 *#Блок работы с меню регистрации* if body.lower() == **"фио"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Введи свои фамилию, имя и отчество"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"телефон"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Введи свой контактный телефон"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"команда"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Введи название своей команды:"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"проект"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Введи название вашего проекта"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
  
 *#Блок записи данных пользователя* elif oldmessage == **"фио"**:  
 name = oldmessage = body  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Приятно познакомиться, "** + name,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 continue *#завершение работы текущего цикла* elif oldmessage == **"команда"**:  
 team = oldmessage = body  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Твоя команда: "** + team,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 continue  
 elif oldmessage == **"проект"**:  
 project = oldmessage = body  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Ваш проект: "** + project,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 continue  
 elif oldmessage == **"телефон"**:  
 phone = oldmessage = body  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Введен телефон: "** + phone,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 continue  
 else: vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Выбери данные для ввода в меню или задай другой вопрос:"**, **"keyboard"**: keyboard,  
 **"random\_id"**: random.randint(1, 2147483647)})  
  
 *# Блок работы с меню Новостей* elif body.lower() == **"новости"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Какие именно - выбери:"**, **"keyboard"**: keyboard1, **"random\_id"**: random.randint(1, 2147483647)})  
 elif body == **"💶 Информация"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Посмотри видео-обзор: "**, **"attachment"**: **"video-15755094\_456274907"**, **"random\_id"**: random.randint(1, 2147483647)})  
 elif body == **"📆 Новости"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Посмотри новости на стене"**, **"attachment"**: **"wall-154074615\_9246"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body == **"🎭 Мероприятия"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Вот афиша событий"**, **"attachment"**: **"photo-154074615\_457256003"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body == **"💶 Наш сайт"**:  
 vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: **"Ссылка на сайт"**, **"attachment"**: **"https://www.eduteam.info/"**,  
 **"random\_id"**: random.randint(1, 2147483647)})  
 elif body.lower() == **"привет"**:  
 str = **"Привет, "** + name + **"!"** vk.method(**"messages.send"**,  
 {**"peer\_id"**: id, **"message"**: str, **"keyboard"** : keyboard0, **"random\_id"**: random.randint(1, 2147483647)})  
 else:  
 vk.method(**"messages.send"**, {**"peer\_id"**: id, **"message"**: **"Не понял тебя!"**, **"keyboard"** : keyboard0, **"random\_id"**: random.randint(1, 2147483647)})  
 print(body, **" +oldнепонял+"**, oldmessage) #Печать сообщения в терминал для отладки ошибок

oldmessage = body.lower()  
 except Exception as E:  
 time.sleep(1)