

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА  
ЭКОНОМИКИ»

**Московский институт электроники и математики им. А.Н. Тихонова**

Матанов Кирилл Алексеевич и Никитин Владимир Олегович, БИВ202

# Бот для перевода сообщений из программы Zulip в социальную сеть Вконтакте

## *Отчёт*

по направлению 09.03.01 Информатика и вычислительная техника студента  
образовательной программы бакалавриата «Информатика и вычислительная  
техника»

Никитин Владимир Олегович,  
Матанов Кирилл Алексеевич.

Руководитель  
Романова Ирина Ивановна

Москва 2021 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**ЗАДАНИЕ**  
**на курсовую работу бакалавра**

студентам группы БИВ202 Матанову Кириллу Алексеевичу  
и Никитину Владимиру Олеговичу

1. Тема работы

Разработка бота для перевода сообщений из программы Zulip в социальную сеть  
Вконтакте

2. Требования к работе

2.1 Общие требования:

Бот для синхронизации сообщений между платформами Zulip и Вконтакте.

2.2 Требования к процессу разработки:

К процессу разработки бота предъявляются следующие требования:

- Обеспечить передачу данных между ботами двух платформ;
- Обеспечить наличия диалога с пользователем;

2.3 Требования к программно-аппаратному обеспечению

Любое устройство с доступом к платформам Zulip и Вконтакте.

## **Аннотация**

Объектом разработки курсовой работы является разработка программы, синхронизирующей сообщения между двумя платформами: Вконтакте и Zulip. Целью – создания кода, реализующего приём, обработку и передачу сообщений обеих платформ. Итоговая программа может быть использована людьми, работающими в компаниях, которые используют Zulip.

Работа содержит 34 иллюстраций, 10 источников литературы. Отчет содержит 39 страниц.

## **Annotation**

The object of the development of the course work is to develop a program that synchronizes messages between the two platforms: Facebook and Zulip. Purpose – to create the code that implements the reception, processing and transmission of messages both platforms. The final program can be used by people working in companies that use Zulip.

The work contains 34 illustrations, 10 sources of literature. The report contains 39 pages.

## Оглавление

Введение .....	5
1. Описание аналогов.....	5
2. Описание ПО.....	6
3. Описание функционала.....	8
4. Результаты тестирования ПО.....	11
5. Руководство пользователя.....	14
6. Заключение.....	25
Список источников.....	26
Производительность ПО и потребляемые ресурсы.....	27
Приложение. Код программы.....	27

## Введение

Задачей данной курсовой работы является создание связи между двумя платформами. Данный механизм позволит облегчить работу людям, которые вынуждены пользоваться различными платформами, что занимает у них много времени на то, чтобы проверять наличие сообщений в разных приложениях, теперь им нужно будет следить только за сообщениями в одном из приложений. Для выполнения использовался язык программирования Python, который был изучен в ходе выполнения.

Так как курсовая работа подразумевает только написание программного кода, без использования специфического оборудования, кроме компьютера, что позволило выполнять ее дистанционно в условиях эпидемиологической ситуации. На данный момент существует большое количество разнообразных ботов, которые выполняют различные функции, но такого, который синхронизирует сообщения в диалогах социальной сети Вконтакте с каналом в Zulip, ещё нет.

Для выполнения работы потребовалось:

1. Узнать, как создаются боты;
2. Понять, как связать между собой ботов для социальной сети Вконтакте и программы Zulip;
3. Разобраться в основах работы ботов для социальной сети Вконтакте;
4. Разобраться в основах работы ботов для программы Zulip;
5. Понять, как разрабатываются боты;
6. Изучить язык программирования Python.

## Описание аналогов

В современном мире людям приходится работать в нескольких системах, например, в программе Zulip и социальной сети Вконтакте. Такая работа является не очень удобной и эффективной, так как пользователю постоянно приходится переключаться между системами и сталкиваться с тем, что информация повторяется. Для решения этой проблемы можно использовать бота, который синхронизирует сообщения между этими платформами. Но для начала нужно понять, что такое бот.

Бот — специальная программа, выполняющая автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

Прямых аналогов боту для пересылки сообщений из программы Zulip в социальную сеть Вконтакте не существует, но есть боты, которые по функционалу похожи на него: они переводят посты группы из социальной сети Вконтакте в такие платформы, как Telegram и Discord. Для работы такого бота необходимо знание API всех платформ, куда пересылаются посты.

А также существуют другие виды ботов: Биржевые боты – программы, которые анализируют ситуацию на рынке и моментально принимают решения. Веб-боты – программы, которые анализируют данные сайтов и распределяют их по базам данных, например, такие боты собирают email-адреса пользователей для спам-рассылки. Боты поисковых систем – программы, которые помогают работать поисковой системе: в большинстве случаев просматривают адреса сайтов и присваивают им индексы, к которым обращается впоследствии поисковик после запроса. Чат-боты – программы, которые отвечают на сообщения человека заданным заранее текстом. Существуют также боты для чатов, которые пересылают сообщения одного человека другому. Также боты используются во вредоносных целях: человеку на компьютер попадает вирус, который впоследствии использует вычислительные ресурсы этого компьютера в своих целях, тем самым делая его менее работоспособным. Бота для перевода сообщений из программы Zulip в социальную сеть Вконтакте можно отнести к типу ботов для чатов, так как он будет пересылать

сообщения людей из одной платформы в другую.

У такого бота есть ряд плюсов:

1. Пользователю не придется устанавливать множество различных приложений для общения с другими людьми, что позволяет сохранить место на памяти устройства;
2. Все действия происходят в привычном для пользователя интерфейсе, поэтому нет необходимости привыкать к новому внешнему виду приложения;
3. Доступность приложений для общения практически на всех платформах устройств, что увеличивает полезность и необходимость бота такого типа.

Однако у ботов есть существенный недостаток: они нуждаются в постоянной поддержке со стороны разработчика. Так как в современном мире все приложения постоянно обновляются и модифицируются, вследствие чего бот должен подстраиваться под эти изменения.

Аналогов разработанному боту, связывающих Zulip и Вконтакте, не существует, но имеются программы, которые синхронизируют между собой другие платформы:

1. Discord – Telegram;
2. VK – Telegram;
3. Discord – VK;
4. WhatsApp – Telegram.

Первым минусом всех этих ботов является то, что их используют больше в развлекательных целях, например, Discord – программа, используемая для связи в играх, Вконтакте и Telegram – приложения, созданные для неформального общения людей и распространения информации, WhatsApp – обычный мессенджер. В то время



как Zulip – приложение, которое широко используют при работе в компаниях или просто при работе над проектами, в которых участвует большое количество людей. Именно это выделяет бота, разработанного в ходе курсовой работы.

## **Описание ПО**

### **Setup.exe:**

Данный файл создаёт необходимые для работы бота файлы и скачивает нужные библиотеки. Работа файла начинается с того, что он скачивает 4 библиотеки: tk, requests, vk\_api, zulip. После чего происходит скачивание файлов: MainBot.exe, Zulip\_getter.py, VK\_getter.py с github. Далее создаётся папка Data с расположенными в ней файлами: id\_chat.txt, private\_inform.txt, Settings.txt. В этих файлах создаётся основа для сохранения в них информации. Сохранение данных происходит именно в файлы, потому что каждая из частей программы использует эти данные. После этого с помощью библиотеки tkinter поочерёдно создаются 3 окна с полями для ввода необходимой информации, которая после этого записывается в выше перечисленные текстовые файлы.

### **MainBot.exe:**

Работа файла начинается с запуска в новых процессах файлов Zulip\_getter.py, VK\_getter.py. После чего происходит выполнение функции file\_read которая считывает информацию из файла private\_inform.txt. Далее происходит подключение к ботам в Zulip и Вконтакте с помощью только что считанной информации. Потом создаётся socket, происходит подключение локального хоста и порта и запускается режим прослушивания. Данный метод позволяет синхронно использовать функции получения сообщений из обеих платформ, тем самым время пересылки сообщения сводится к минимальному. Далее начинается цикл, который повторяется до тех пор, пока не будет получено сообщение /stop. В данном цикле происходит приём данных и их обработка.

### **Zulip\_getter.py:**

Работа программы начинается с такой же функции `file_read`. Потом происходит подключение к боту в Zulip. Происходит получение сообщения, подключение к серверу, созданному в программе `MainBot.exe` и отправка полученного сообщения на сервер.

### **VK\_getter.py:**

Работа программы начинается с такой же функции `file_read`. Потом происходит подключение к боту в Вконтакте. Происходит получение сообщения, подключение к серверу, созданному в программе `MainBot.exe` и отправка полученного сообщения на сервер.

## **Описание функционала**

Бот осуществляет синхронизацию сообщений между беседой Вконтакте и чатом Zulip. Существует список команд, которые влияют на работу бота:

`/help` – выводит список всех команд

`/start` - запускает бота

`/stop` - выключает бота

`/include email true(false)` - добавление в сообщение email пользователя, отправившего сообщение из Zulip

`/include display_recipient true(false)` - добавление в сообщение название канала Zulip, из которого пришло сообщение

`/include subject true(false)` - добавление в сообщение темы канала Zulip, из которого пришло сообщение

`/include id true(false)` – добавление в сообщение id пользователя Вконтакте

`/status` - отображение статуса настроек

## Результаты тестирования ПО

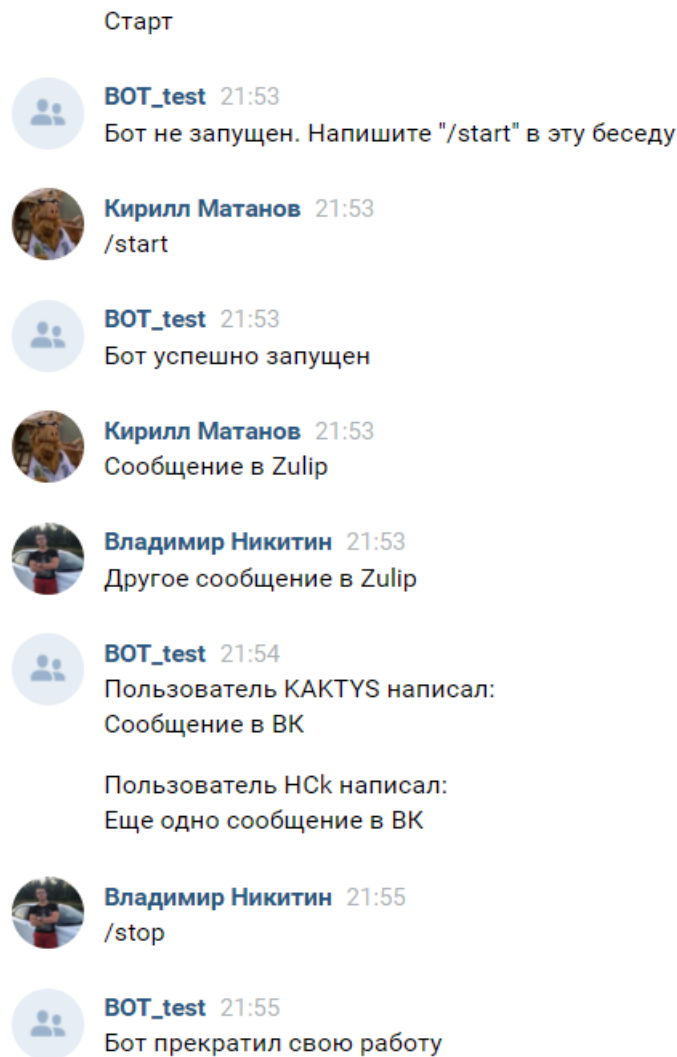


Рис. 29. Пример пересылки в Вконтакте.

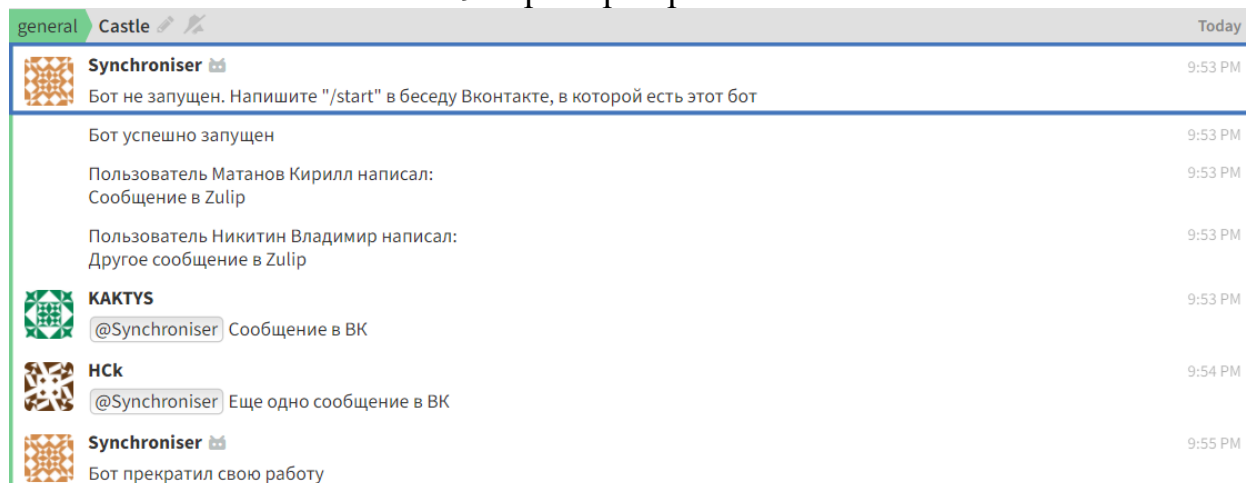


Рис. 30. Пример пересылки в Zulip.

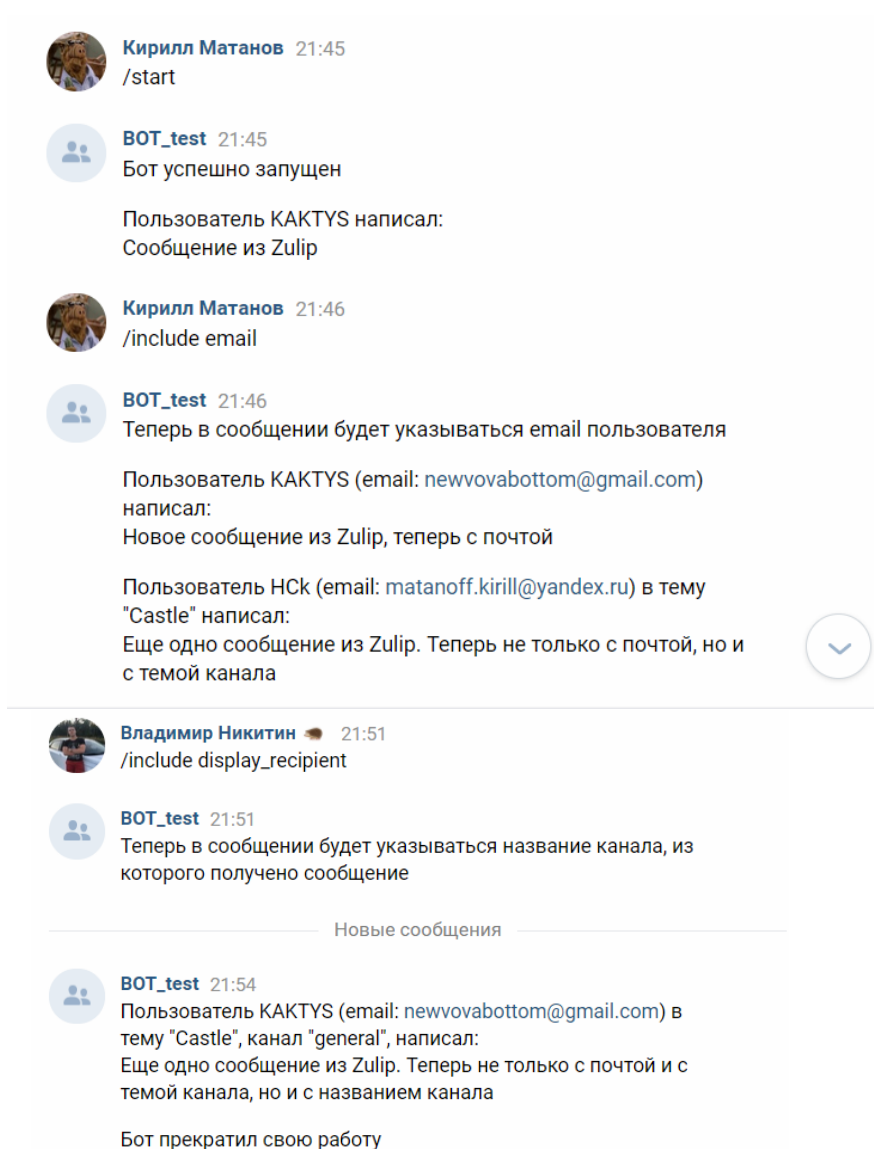


Рис. 31. Пример работы команд в Вконтакте.

general	Castle	Today
	<b>Synchroniser</b> Бот успешно запущен	9:45 PM
	<b>KAKTYS</b> @Synchroniser Сообщение из Zulip	9:45 PM
	@Synchroniser Новое сообщение из Zulip, теперь с почтой	9:47 PM
	@Synchroniser /include subject	9:48 PM
	<b>Synchroniser</b> Теперь в сообщении будет указываться тема, из которой получено сообщение	9:49 PM
	<b>HCK</b> @Synchroniser Еще одно сообщение из Zulip. Теперь не только с почтой, но и с темой канала	9:50 PM
	<b>KAKTYS</b> @Synchroniser Еще одно сообщение из Zulip. Теперь не только с почтой и с темой канала, но и с названием канала	9:53 PM
	@Synchroniser /stop	9:54 PM
	<b>Synchroniser</b> Бот прекратил свою работу	9:55 PM

Рис.32. Пример работы команд в Zulip.

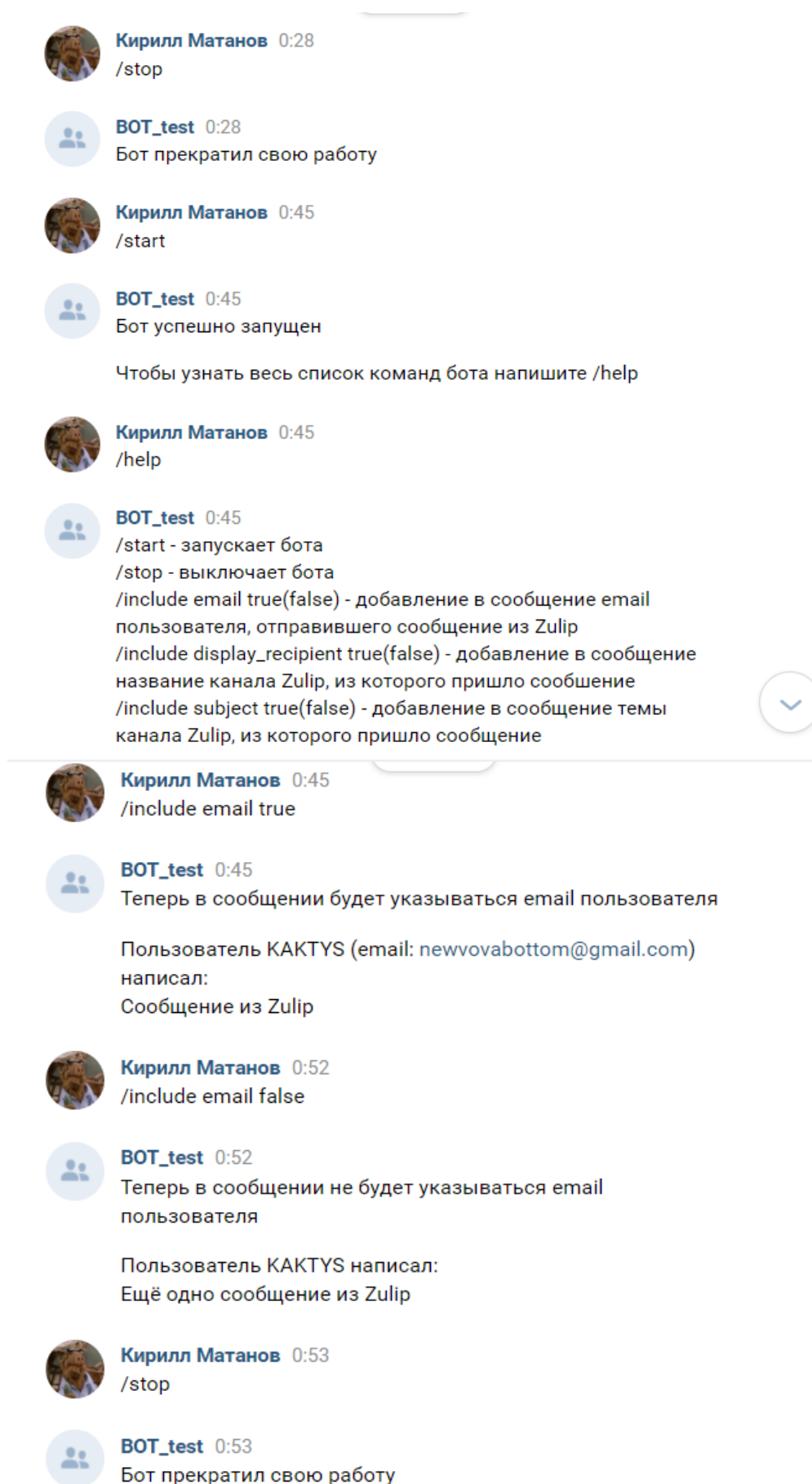


Рис.33. Пример работы команд в Вконтакте.

## Руководство пользователя

Подготовка к работе с ботом:

### 1. Создание новой организации:

1.1 Перейти по ссылке <https://zulip.com/new/>

1.2 Заполнить поле и нажать «Создать организацию»:

Рис. 1. Окно создания новой организации Zulip.

1.3 Перейдя по ссылке, отправленной на почту, заполнить все поля и нажать зарегистрироваться

Рис. 2. Окно регистрации организации.

- 1.4 В созданной организации нажать на шестерёнку в правом верхнем углу и выбрать пункт settings:

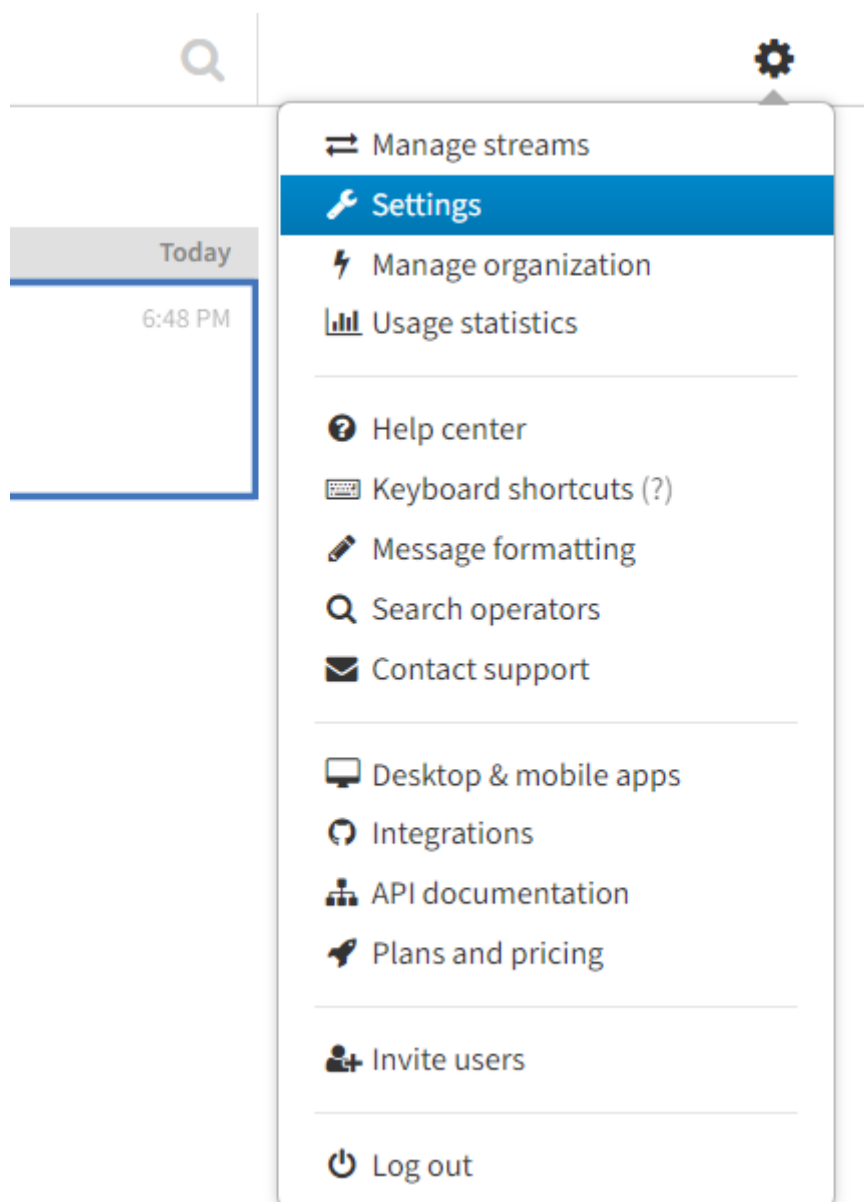


Рис. 3. Список настроек организации.

- 1.5 В открывшемся окне перейти на вкладку Your bots и заполнить все необходимые поля, нажать Create bot:

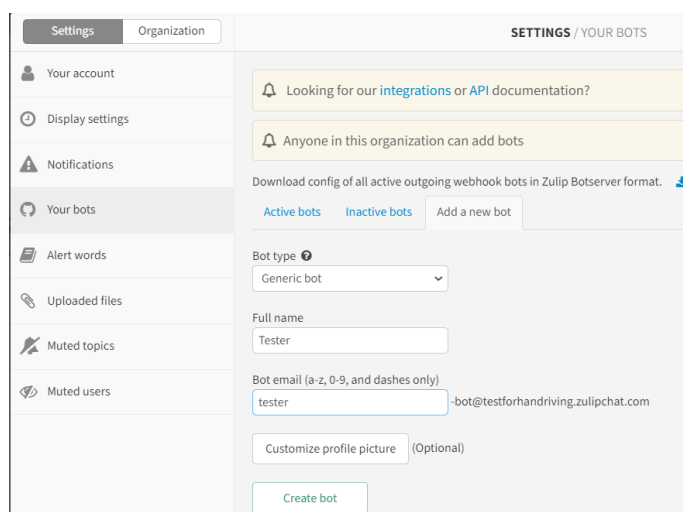


Рис. 4. Окно создания бота.

- 1.6 Нажать на кнопку скачать около имени бота:

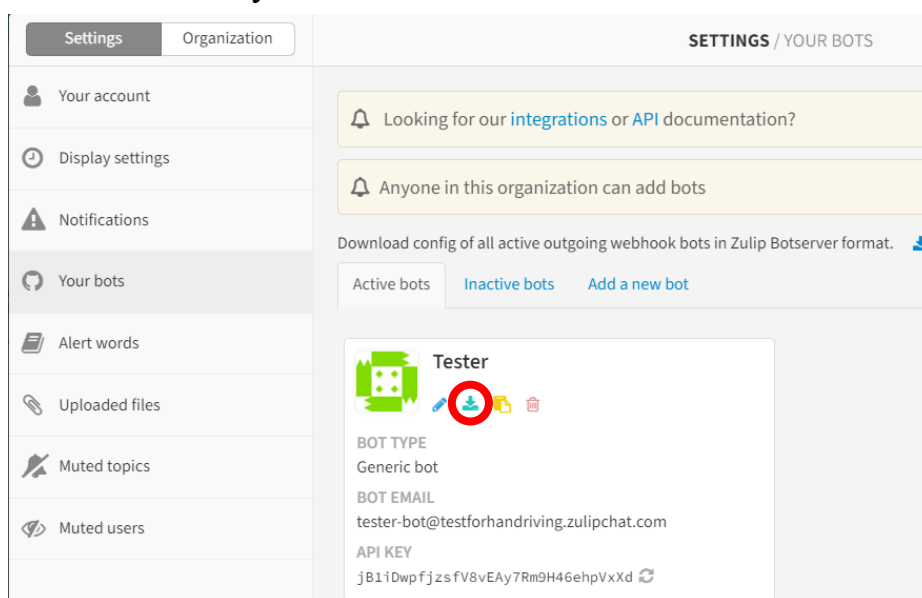


Рис. 5. Окно с информацией о созданных ботах.

2. Создание группы Вконтакте и получение всех необходимых данных:

- 2.1 Открыть вкладку сообщества и нажать «Создать сообщество»

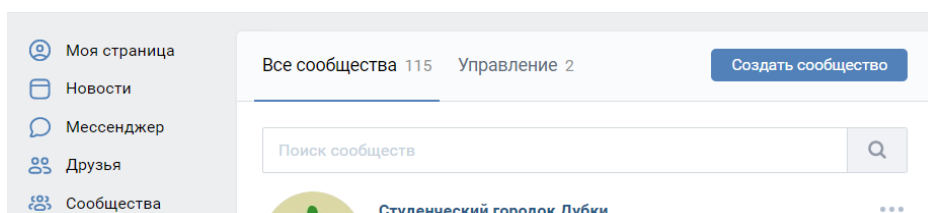


Рис. 6. Вкладка «сообщества».



## 2.2 Выбрать тип сообщества

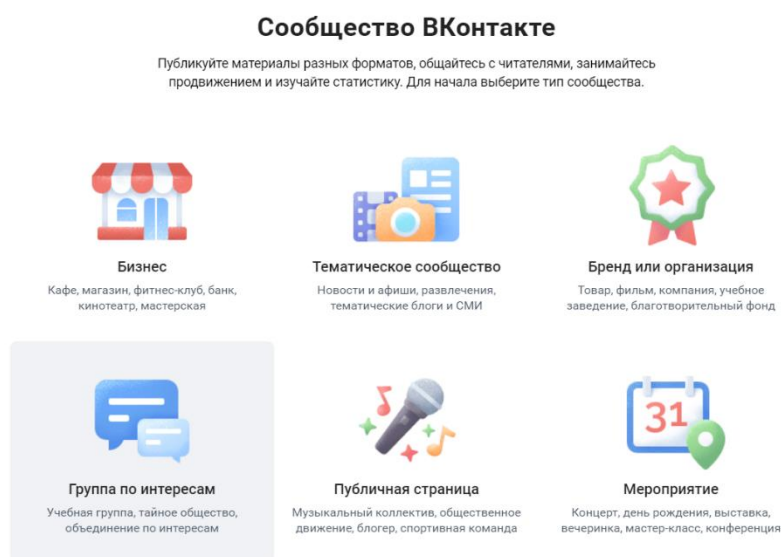


Рис. 7. Окно выбора типа сообщества.

## 2.3 Заполнить информацию о группе и нажать «Создать сообщество»

Создание сообщества

**Группа по интересам**  
Общайтесь и делитесь контентом с одноклассниками, коллегами и единомышленниками

Название:

Тематика:

Тип группы:

Сайт (если есть):

Адрес: [Указать адрес](#)

[Отмена](#) [Создать сообщество](#)

Рис. 8. Окно создания сообщества.

## 2.4 Зайти в меню «управление»

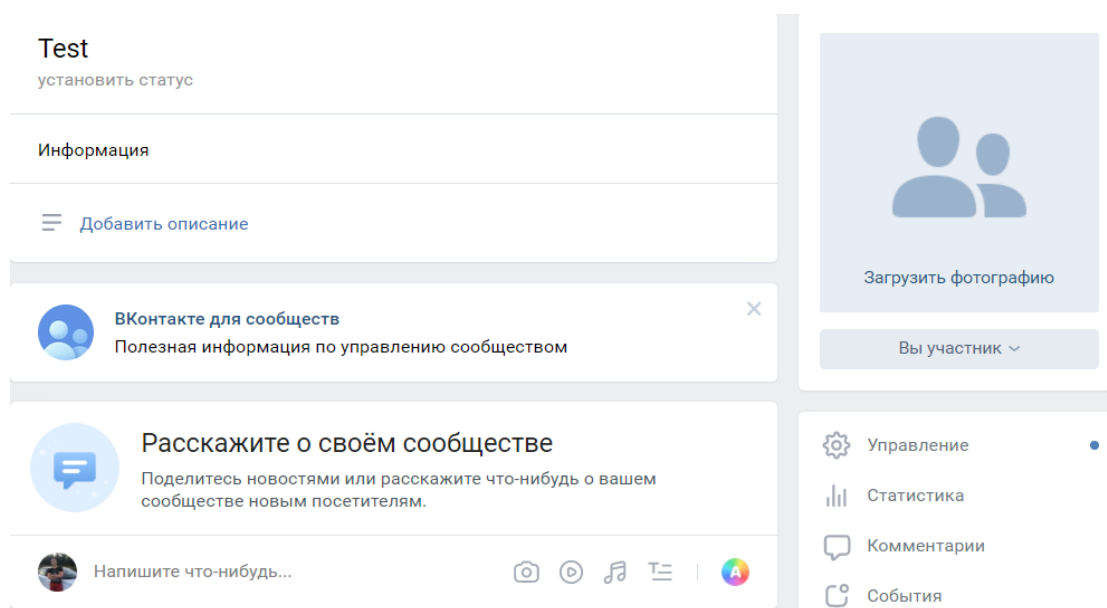


Рис. 9. Главная страница сообщества.

## 2.5 Перейти в раздел «Работа с API»

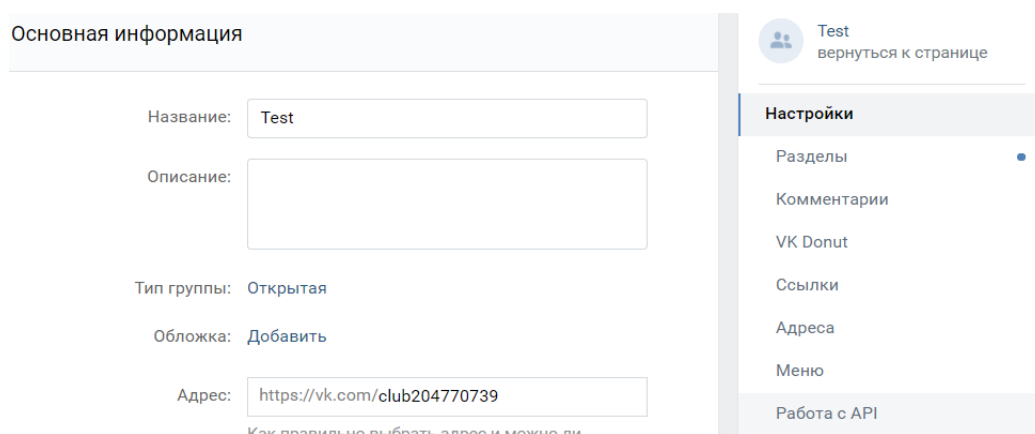
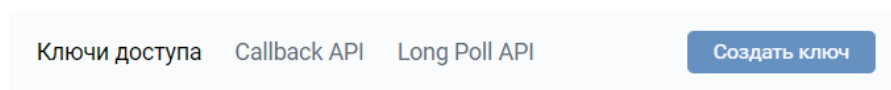


Рис. 10. Раздел настроек сообщества «Работа с API».

## 2.6 Нажать «Создать ключ»



Здесь будут выводиться полученные ключи.

Рис. 11. Создание ключа доступа.

2.7 Выбрать необходимые права и нажать «Создать» и подтвердить создание с помощью телефона в следующем диалоговом окне

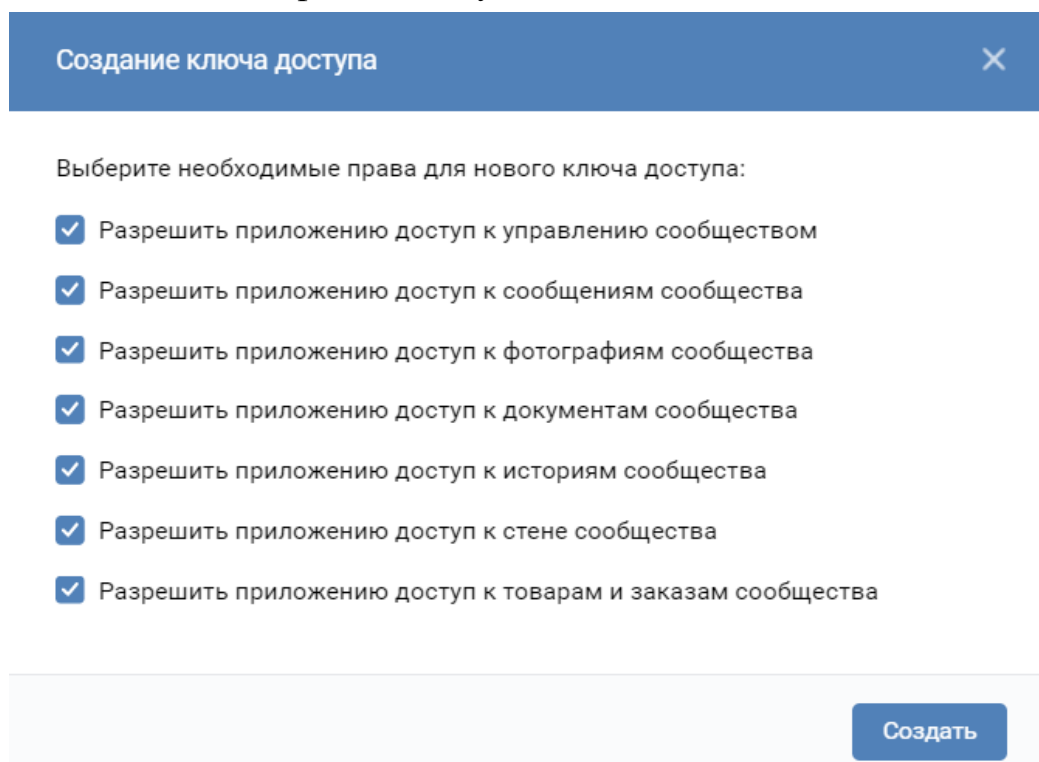


Рис. 12. Окно выбора прав ключа доступа.

2.8 Созданный ключ необходимо сохранить, чтобы использовать дальше при запуске бота.

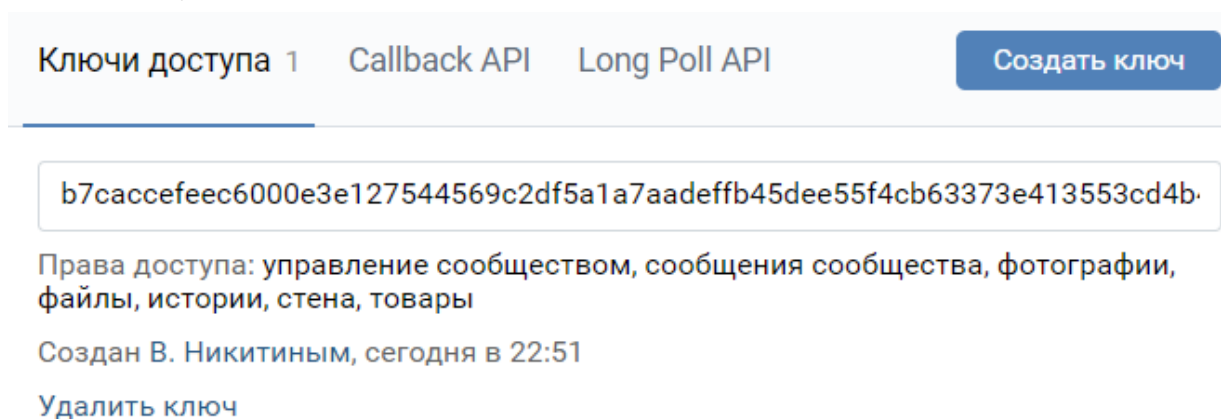


Рис. 13. Поле с созданным ключом доступа.

2.9 Перейти на главную страницу группы и скопировать её id из адреса страницы только цифры:

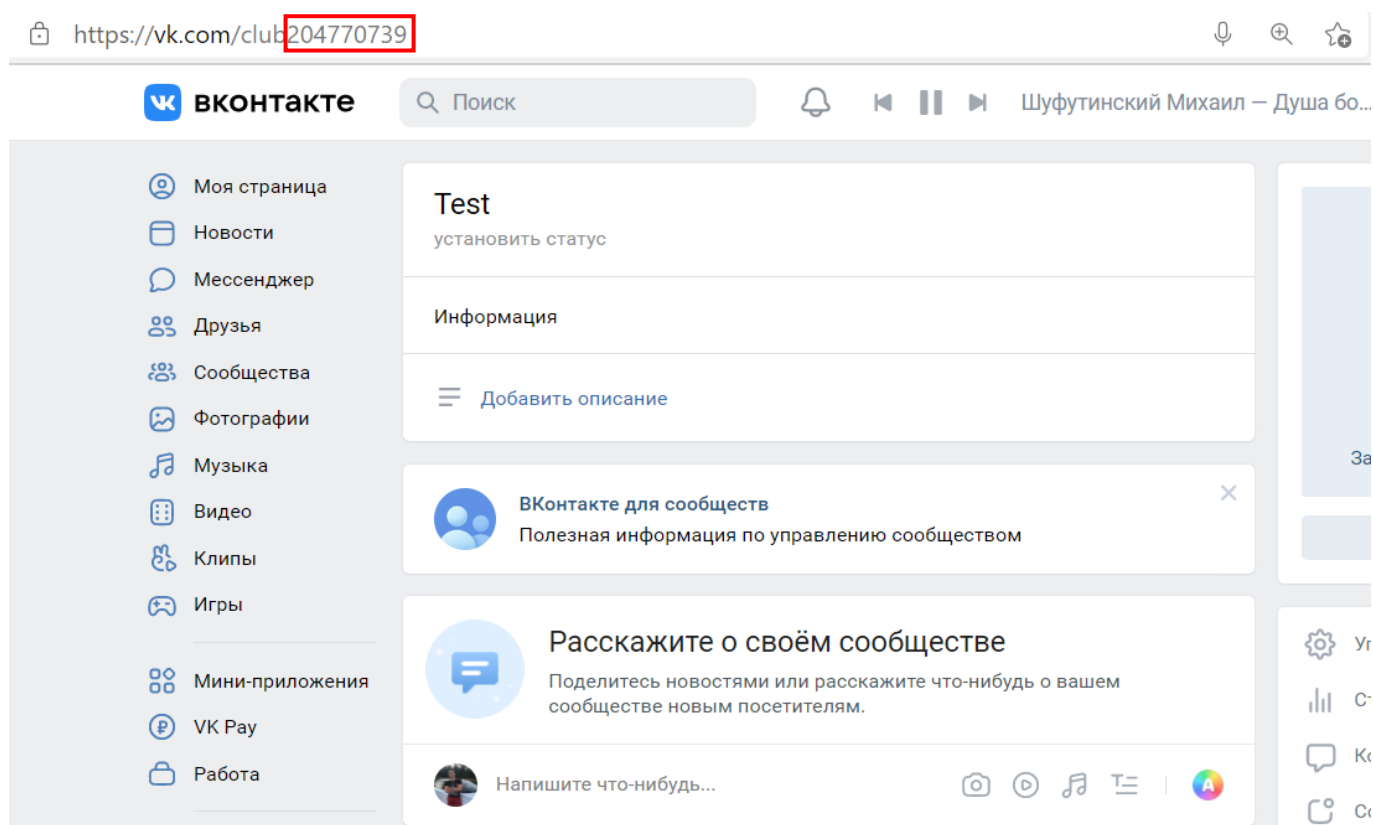


Рис. 14. Главная страница сообщества.

## Процесс установки и запуска:

1. Перед началом работы необходимо скачать и установить Python 3.9.0. Это можно сделать по ссылке <https://www.python.org/downloads/>

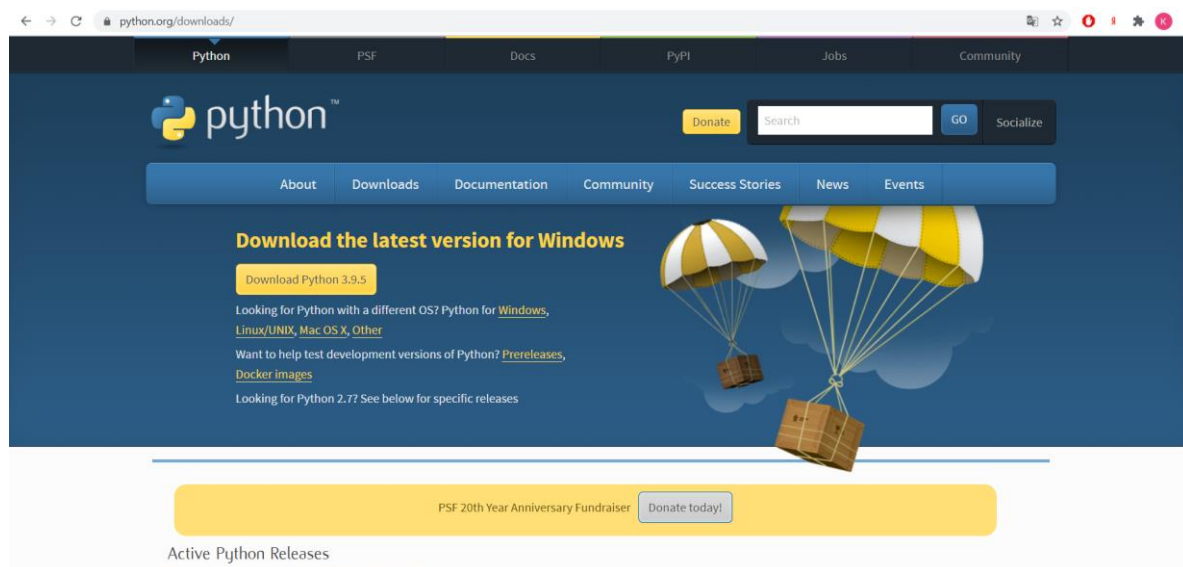


Рис. 15. Страница загрузки Python.

2. Для того чтобы скачать все необходимые файлы нужно скачать файл Setup.py.



Рис. 16. Файл Setup.py.

3. Для начала установки необходимо открыть командную строку и вписать команду `cd <путь до папки с файлом Setup.py>`

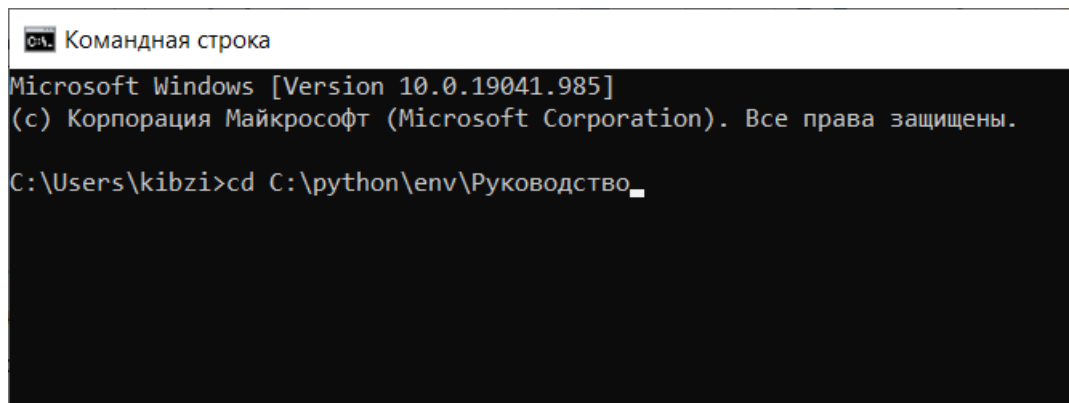


Рис. 17. Окно командной строки с командой для перехода в нужную директорию.

4. Для запуска файла Setup.py необходимо написать команду `python Setup.py`

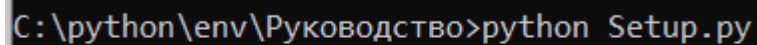
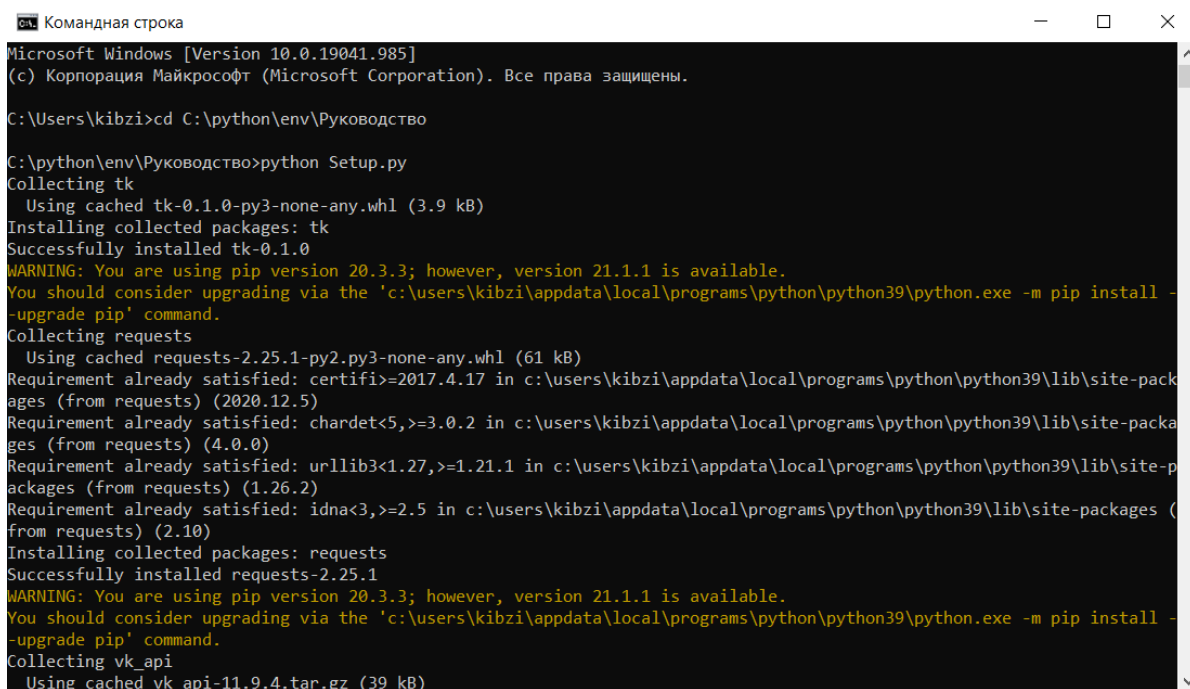
A screenshot of a Windows Command Prompt window. The text inside shows 'C:\python\env\Руководство>python Setup.py' with a cursor at the end.

Рис. 18. Команда для запуска файла Setup.py

5. Если все предыдущие шаги выполнены верно, то в командной строке появится подобная информация.



```
Командная строка
Microsoft Windows [Version 10.0.19041.985]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\kibzi>cd C:\python\env\Руководство

C:\python\env\Руководство>python Setup.py
Collecting tk
  Using cached tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0
WARNING: You are using pip version 20.3.3; however, version 21.1.1 is available.
You should consider upgrading via the 'c:\users\kibzi\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.
Collecting requests
  Using cached requests-2.25.1-py2.py3-none-any.whl (61 kB)
Requirement already satisfied: certifi<=2017.4.17 in c:\users\kibzi\appdata\local\programs\python\python39\lib\site-packages (from requests) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\kibzi\appdata\local\programs\python\python39\lib\site-packages (from requests) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\kibzi\appdata\local\programs\python\python39\lib\site-packages (from requests) (1.26.2)
Requirement already satisfied: idna<3,>=2.5 in c:\users\kibzi\appdata\local\programs\python\python39\lib\site-packages (from requests) (2.10)
Installing collected packages: requests
Successfully installed requests-2.25.1
WARNING: You are using pip version 20.3.3; however, version 21.1.1 is available.
You should consider upgrading via the 'c:\users\kibzi\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.
Collecting vk_api
  Using cached vk_api-11.9.4.tar.gz (39 kB)
```

Рис. 19. Пример вывода работы файла Setup.py.

6. Далее поочерёдно появятся 3 окна, в которые необходимо ввести ключ доступа сообщества с ботом Вконтакте, id этого сообщества, и путь до файла zuliprc на вашем компьютере.

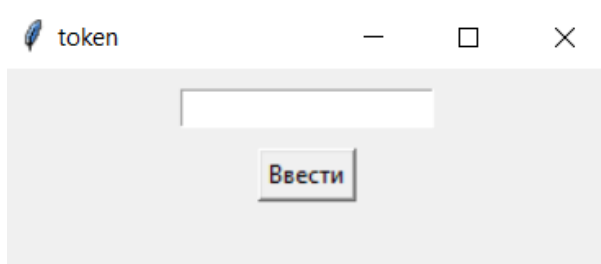


Рис. 20.

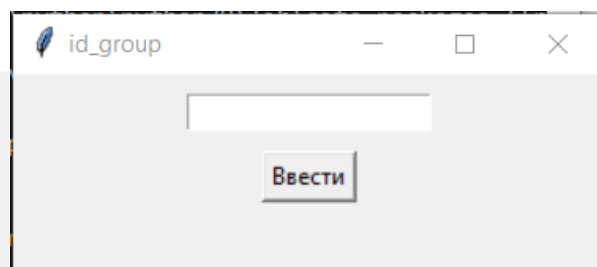


Рис. 21

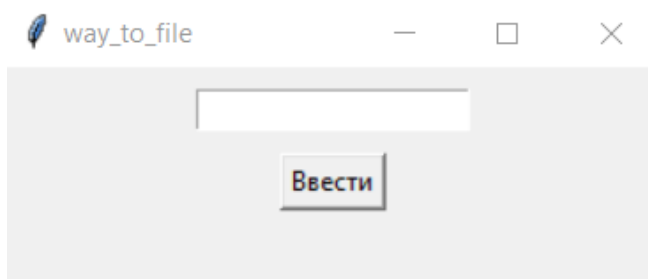
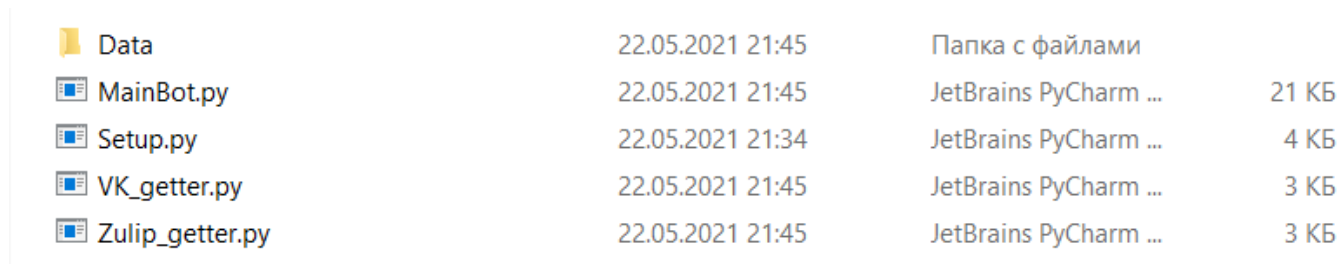


Рис. 22. Окна для ввода данных.

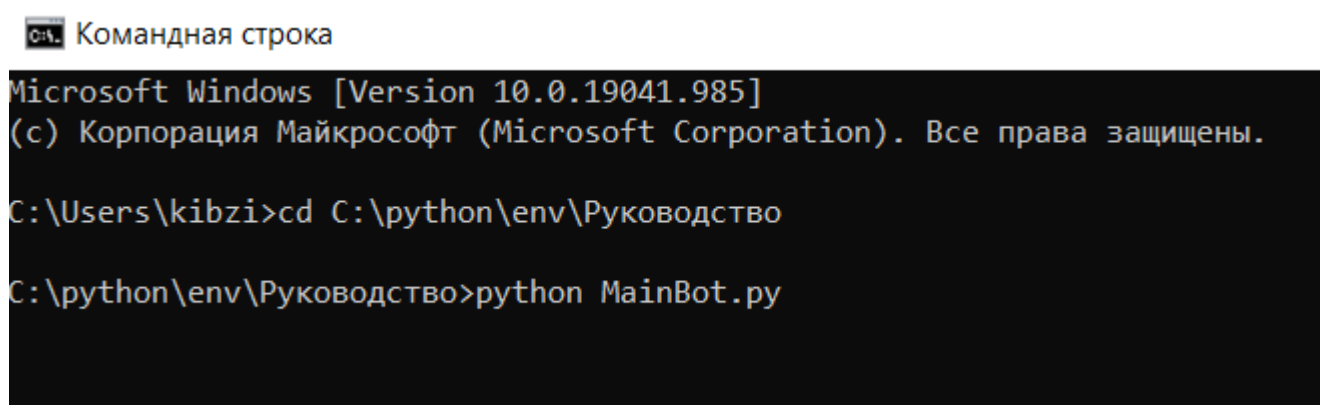
7. После выше перечисленных действий в директории из которой был запущен файл Setup появится папка Data, содержащая 3 текстовых файла: id\_chat.txt, private\_inform.txt, Settings.txt, и будут скачаны 3 файла: MainBot.exe, VK\_getter.py, Zulip\_getter.py.



Data	22.05.2021 21:45	Папка с файлами	
MainBot.py	22.05.2021 21:45	JetBrains PyCharm ...	21 КБ
Setup.py	22.05.2021 21:34	JetBrains PyCharm ...	4 КБ
VK_getter.py	22.05.2021 21:45	JetBrains PyCharm ...	3 КБ
Zulip_getter.py	22.05.2021 21:45	JetBrains PyCharm ...	3 КБ

Рис. 23. Созданные программой файлы.

8. Для начала работы бота необходимо запустить файл MainBot.exe, для этого, находясь в папке с файлом MainBot.py, пропишите команду python MainBot.py.



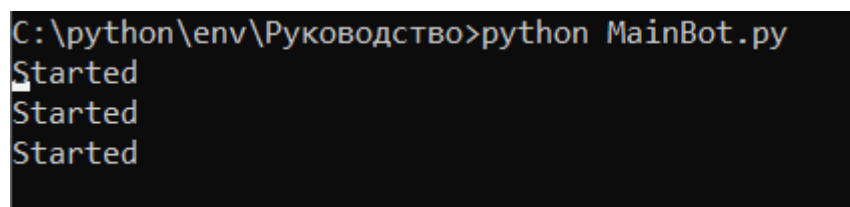
```
Microsoft Windows [Version 10.0.19041.985]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\kibzi>cd C:\python\env\Руководство

C:\python\env\Руководство>python MainBot.py
```

Рис. 24. Команда для запуска MainBot.py.

9. После запуска в командной строке появится три надписи Started, что сигнализирует о том, что все корректно запущено.



```
C:\python\env\Руководство>python MainBot.py
Started
Started
Started
```

Рис. 25. Вывод программы при корректном запуске.

10. После запуска файла в беседу Вконтакте необходимо написать /start. Для того, чтобы узнать полный список команд, напишите либо в беседу Вконтакте, либо в канал Zulip сообщение /help (Чтобы обратиться к боту в Zulip, напишите @\*\*<Имя бота>\*\*).

Чтобы узнать весь список команд бота напишите /help



**Владимир Никитин** 22:13

/help



**BOT\_test** 22:13

/start - запускает бота

/stop - выключает бота

/include email true(false) - добавление в сообщение email пользователя, отправившего сообщение из Zulip

/include display\_recipient true(false) - добавление в сообщение название канала Zulip, из которого пришло сообщение

/include subject true(false) - добавление в сообщение темы канала Zulip, из которого пришло сообщение

/include id true(false) - добавление в сообщение id пользователя Вконтакте

/status - отображение статуса настроек

Рис. 26. Пример работы команды /help.

11. Для завершения работы бота напиши либо в беседу Вконтакте, либо в канал



Zulip сообщение /stop.

Рис. 27. Завершение работы бота.

12. Для возобновления работы бота, повторите пункты 7-9.

13. Для повторной установки бота необходимо перейти в другую папку, либо удалить все файлы.

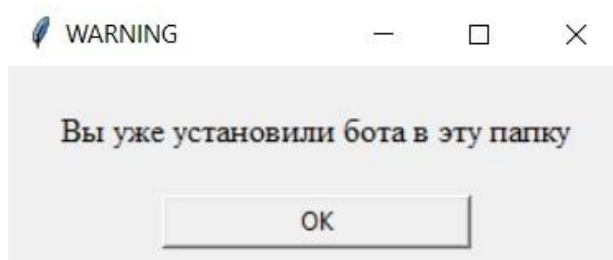


Рис. 28. Окно-уведомление о том, что бот уже установлен.



## Заключение

Результатом выполнения курсовой работы является программа, обладающая более широким функционалом, чем та, которая была заявлена в техническом задании. В начале разработки бота уровень знаний и понимания в данной сфере был достаточно низким, вследствие чего возникало много трудностей, для решения которых приходилось углубляться в различные методы их решения. Одной из таких проблем являлась большая задержка при пересылке сообщения из одной платформы в другую. Для решения этой проблемы пришлось попробовать множество вариаций написания кода. Первым вариантом был бесконечный цикл, выход из которого осуществлялся по команде /stop, но это решение оказалось неудачным, в силу особенностей API Вконтакте, вследствие чего сообщение из Zulip не пересылалось до тех пор, пока не придёт сообщение из Вконтакте. Далее была попытка использования потоков, что привело к полной неработоспособности программы. В итоге, желаемый результат был достигнут использованием сокетов. Время пересылки сообщения удалось сократить до 0.3 секунды.

Для удобства установки программы был разработан специальный файл, при запуске которого устанавливаются все необходимые библиотеки и скачиваются все необходимые файлы, а также появляются удобные диалоговые окна, в которые пользователь должен вводить информацию необходимую для запуска бота, а именно токен группы Вконтакте, её id и путь до файла zuliprc. А для удобства использования были добавлены различные команды, дополняющие текст сообщения информацией о пользователе, отправившем его.

Также результатом выполнения курсовой работы является изучения языка программирования Python, который широко используется во многих сферах и пригодится при дальнейшем обучении. Были получены базовые навыки использования API Вконтакте и Zulip, потоков и сокетов.

## Список использованных источников

1. Zulip Api documentation  
URL: <https://zulip.com/api/>
2. Vk Api documentation  
URL: <https://vk.com/dev/manuals>
3. Python PEP  
URL: <https://www.python.org/dev/peps/pep-0008/>
4. Статья Хабр о создание бота  
URL: <https://habr.com/ru/post/427691/>
5. Making chat bots the Zulip  
URL: <https://code.likeagirl.io/myra-a-smart-bot-14d5a0ec755c>
6. Пишем бота пересылки сообщений из VK в Telegram на Python  
URL: [nikovit.ru/blog/pishem-bota-peresytki-soobshcheniy-iz-vk-v-telegram-na-python/](http://nikovit.ru/blog/pishem-bota-peresytki-soobshcheniy-iz-vk-v-telegram-na-python/)
7. Поток и многопоточность / для начинающих  
URL: <https://pythonru.com/uroki/potoki-i-mnogopotchnost-dlja-nachinajushhih>
8. Сокеты в Python для начинающих  
URL: <https://habr.com/ru/post/149077/>
9. Обучение Python GUI (уроки по Tkinter)  
URL: <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter>
10. Файлы в python, ввод-вывод  
URL: <https://pythonru.com/osnovy/fajly-v-python-vvod-vyvod>

## Производительность ПО и потребляемые ресурсы


Имя	Состояние	ЦП	Память	Диск	Сеть
>  Обработчик команд Windows ...		0%	57,5 МБ	0 МБ/с	0,1 Мбит/с

Рис.34. Потребляемые ресурсы.

Задержка сообщения из Zulip в Вконтакте: 0.27 сек.

Задержка сообщения из Вконтакте в Zulip: 0.21 сек.

## Код программы

### Zulip\_getter.py

```
import socket
import zulip
from typing import Any, Dict
import time

def file_read():
    """Функция file_read получает информацию из файла private_inform"""

    with open('Data/private_inform.txt') as f:
        line1 = f.readline()
        arrline1 = line1.split()
        main_token_from_file = arrline1[1]
        line2 = f.readline()
        arrline2 = line2.split()
        id_group_from_file = int(arrline2[1])
        line3 = f.readline()
        arrline3 = line3.split()
        way_to_config_from_file = arrline3[1]
        return main_token_from_file, id_group_from_file, way_to_config_from_file

def zulip_getter():
    """Функция zulip_getter получает сообщение из каналов приложения Zulip, возвращая строку вида
    "Пользователь <имя пользователя> написал:
    <сообщение пользователя>"
    Для работы функции не требуются входные данные"""

    with open("Data/Settings.txt", "r") as f:
        line1 = f.readline()
        arrline1 = line1.split()
        user_email_bool = arrline1[1]
```

```

        line2 = f.readline()
        arrline2 = line2.split()
        subject_bool = arrline2[1]
        line3 = f.readline()
        arrline3 = line3.split()
        display_recipient_bool = arrline3[1]

request: Dict[str, Any] = {
    'anchor': 'newest',
    'num_before': 1,
    'num_after': 0
}

# получение информации о сообщении
res = zulip_client.get_messages(request)
message = res['messages'][0]['content']
user_name = res['messages'][0]['sender_full_name']
user_email = res['messages'][0]['sender_email']
subject = res['messages'][0]['subject']
display_recipient = res['messages'][0]['display_recipient']
s = message.find('</span>')
e = message.find('</p>')
message = message[s + 8:e].replace('<br>', '')
# формирование текста сообщения для пересылки
if message[0] == ' ':
    message = message[1:len(message)]
if message[0] == '/':
    message_from_zulip = "ZULIP156324"+message
    return message_from_zulip
message_from_zulip = ('ZULIP156324Пользователь {}'.format(user_name))
if user_email_bool == 'True':
    message_from_zulip += f' (email: {user_email})'

if subject_bool == 'True':
    message_from_zulip += f' в тему "{subject}"'
if display_recipient_bool == 'True':
    message_from_zulip += f', канал "{display_recipient}",'
message_from_zulip += f' написал:\n{message}'
return message_from_zulip

print("Started")
# подключение к боту Вконтакте
main_token, id_group, way_to_config = file_read()

zulip_client = zulip.Client(config_file="{}".format(way_to_config))

old_message_from_zulip = zulip_getter()

# основной цикл программы
while True:
    time.sleep(2)
    message_from_zulip = zulip_getter()
    if message_from_zulip != old_message_from_zulip:
        with socket.create_connection(("127.0.0.1", 10002)) as sock:
            sock.sendall(message_from_zulip.encode("utf8"))
        old_message_from_zulip = message_from_zulip

```

## Vk\_getter.py

```
import socket
import vk_api
from vk_api.bot_longpoll import VkBotLongPoll, VkBotEventType

def file_read():
    """Функция file_read получает информацию из файла private_inform"""

    with open('Data/private_inform.txt') as f:
        line1 = f.readline()
        arrline1 = line1.split()
        main_token_from_file = arrline1[1]
        line2 = f.readline()
        arrline2 = line2.split()
        id_group_from_file = int(arrline2[1])
        line3 = f.readline()
        arrline3 = line3.split()
        way_to_config_from_file = arrline3[1]
    return main_token_from_file, id_group_from_file, way_to_config_from_file

def vk_getter():
    """Функция vk_getter получает сообщение из бесед социальной сети Вконтакте,
    в которых состоит бот, возвращая строку вида
    "Пользователь <имя пользователя> написал:
    <сообщение пользователя>"
    Для работы функции не требуются входные данные"""

    for event in longpoll.listen():
        with open("Data/Settings.txt", "r") as f:
            f.readline()
            f.readline()
            f.readline()

            line4 = f.readline()
            arrline4 = line4.split()
            vk_id = arrline4[1]

            if event.type == VkBotEventType.MESSAGE_NEW and event.from_chat:
                id_chat = event.chat_id
                with open("Data/id_chat.txt", "w") as f:
                    f.write(str(id_chat))
                user_id = event.message.get('from_id') # получение айди пользователя, отправившего сообщение
                message_vk = event.message.get('text') # получение текста сообщения
                user = vk_session.method("users.get", {"user_ids": user_id})
                # формирование текста сообщения для пересылки
                if message_vk[0] == '/':
                    message_from_vk = "VK156324"+message_vk
                    return message_from_vk
                if vk_id == 'False':
                    message_from_vk = ("VK156324Пользователь {} {} написал:\n{}".format(user[0]['last_name'],
                                                                                       user[0]['first_name'], message_vk))
                else:
                    message_from_vk = ("VK156324Пользователь {} {} (id: {}) написал:\n{}".format(user[0]['last_name'],
                                                                                               user[0]['first_name'],
                                                                                               user_id, message_vk))
            return message_from_vk

print("Started")
# подключение к боту Вконтакте
main_token, id_group, way_to_config = file_read()

vk_session = vk_api.VkApi(token=main_token)

longpoll = VkBotLongPoll(vk_session, id_group)
```

```

old_message_from_vk = None

# основной цикл программы
while True:
    message_from_vk = vk_getter()
    if message_from_vk != old_message_from_vk:
        with socket.create_connection(("127.0.0.1", 10002)) as sock:
            sock.sendall(message_from_vk.encode("utf8"))
        old_message_from_vk = message_from_vk

```

## MainBot.py

```

import socket
import vk_api
import zulip
from vk_api.bot_longpoll import VkBotLongPoll
from subprocess import Popen

def file_read():
    """Функция file_read получает информацию из файла private_inform"""

    with open('Data/private_inform.txt') as f:
        line1 = f.readline()
        arrline1 = line1.split()
        main_token_from_file = arrline1[1]
        line2 = f.readline()
        arrline2 = line2.split()
        id_group_from_file = int(arrline2[1])
        line3 = f.readline()
        arrline3 = line3.split()
        way_to_config_from_file = arrline3[1]
        return main_token_from_file, id_group_from_file, way_to_config_from_file

def zulip_sender(message):
    """Функция zulip_sender осуществляет отправку сообщений в чат Zulip.
    Для работы функции ей необходимо передать само сообщение."""

    request = {
        "type": "stream",
        "to": "general",
        "topic": "Castle",
        "content": message,

```

```

    }
    return zulip_client.send_message(request)

def vk_sender(id_chat, message):
    """Функция vk_sender осуществляет отправку сообщений в беседу ВК.
    Для работы функции ей необходимо передать id беседы и само сообщение."""

    vk_session.method('messages.send', {'chat_id': id_chat, 'message': message, 'random_id': 0})

print("Started")

with open("Data/Settings.txt", "w") as f:
    f.write("user_email_bool: False\n")
    f.write("subject_bool: False\n")
    f.write("display_recipient_bool: False\n")
    f.write("id_vk_bool: False\n")

main_token, id_group, way_to_config = file_read()
# запуск файлов Zulip_getter.py и VK_getter.py
Popen('python Zulip_getter.py')
Popen('python VK_getter.py')

# подключение чатов zulip и Вконтакте
zulip_client = zulip.Client(config_file="{}".format(way_to_config))
vk_session = vk_api.VkApi(token=main_token)
longpoll = VkBotLongPoll(vk_session, id_group)

user_email_bool = False
subject_bool = False

display_recipient_bool = False
id_vk_bool = False
stop = True
start = False
old_text_zulip = None
old_text_vk = None

help_message = """
/start - запускает бота\n \
/stop - выключает бота\n \
/include email true(false) - добавление в сообщение email пользователя, \
" отправившего сообщение из Zulip\n" \
/include display_recipient true(false) - добавление в сообщение название канала Zulip, \
" из которого пришло сообщение\n" \
/include subject true(false) - добавление в сообщение темы канала Zulip, \
" из которого пришло сообщение\n" \
/include id true(false) - добавление в сообщение id пользователя Вконтакте\n" \
/status - отображение статуса настроек\n
"""

# создание сокета
with socket.socket() as sock:
    sock.bind(("127.0.0.1", 10002))
    sock.listen()

# тело основного цикла с обработкой полученных сообщений
while stop:
    conn, addr = sock.accept()
    with conn:
        # получение данных из программ Zulip_getter.py и VK_getter.py
        data = conn.recv(1024)
        data = data.decode("utf8")

```

```

# выделение текста из всего сообщения
length = len(data)
position = data.find('\n')
if not('\n' in data):
    position = data.find('4')
text = data[position+1:length]

# чтение настроек из файла
with open("Data/Settings.txt", "r") as f:
    line1 = f.readline()
    arrline1 = line1.split()
    user_email_bool = arrline1[1]
    line2 = f.readline()
    arrline2 = line2.split()
    subject_bool = arrline2[1]
    line3 = f.readline()
    arrline3 = line3.split()
    display_recipient_bool = arrline3[1]
    line4 = f.readline()
    arrline4 = line4.split()
    id_vk_bool = arrline4[1]

# чтение id из файла
with open("Data/id_chat.txt", "r") as f:
    id_chat = f.readline()

# обработка сообщения из Zulip
if "ZULIP156324" in data and (old_text_zulip is None or not(text in old_text_zulip)):
    length_data = len(data)
    message_from_zulip = data[11:length_data]
    l2 = len(message_from_zulip)

    pos = message_from_zulip.find("\n")
    old_text_zulip = message_from_zulip[pos+1:l2]
    message_from_zulip = data[11:length_data]
    if start is True and message_from_zulip[0] != '/':
        vk_sender(id_chat, message_from_zulip)
    elif start is False:
        zulip_sender(
            'Бот не запущен. Напишите "/start" в беседу Вконтакте, в которой есть этот бот')
    elif start is True and ('/help' in message_from_zulip):
        zulip_sender(help_message)
    elif start is True and ('/status' in message_from_zulip):
        with open("Data/Settings.txt", "r") as f:
            status = f.read()
        zulip_sender(status)
    elif start is True and ('/stop' in message_from_zulip):
        stop = False
        vk_sender(id_chat, 'Бот прекратил свою работу')
        zulip_sender('Бот прекратил свою работу')
    elif start is True and ('/include email true' in message_from_zulip) \
        and user_email_bool == 'False':
        zulip_sender('Теперь в сообщении будет указываться email пользователя')
        vk_sender(id_chat, 'Теперь в сообщении будет указываться email пользователя')
        with open("Data/Settings.txt", "r") as f:
            all_text = f.read()
        with open("Data/Settings.txt", "w") as f:
            all_text = all_text.replace("user_email_bool: False", "user_email_bool: True")
            f.write(all_text)
    elif start is True and ('/include subject true' in message_from_zulip) \
        and subject_bool == 'False':
        zulip_sender('Теперь в сообщении будет указываться тема, из которой получено сообщение')

```



```

vk_sender(id_chat, 'Теперь в сообщении будет указываться тема, из которой получено сообщение')
with open("Data/Settings.txt", "r") as f:
    all_text = f.read()
with open("Data/Settings.txt", "w") as f:
    all_text = all_text.replace("subject_bool: False", "subject_bool: True")
f.write(all_text)
elif start is True and ('/include display_recipient true' in message_from_zulip) \
    and display_recipient_bool == 'False':
    zulip_sender(
        'Теперь в сообщении будет указываться название канала, из которого получено сообщение')
    vk_sender(id_chat,
        'Теперь в сообщении будет указываться название канала, из которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("display_recipient_bool: False", "display_recipient_bool: True")
    f.write(all_text)
elif start is True and ('/include id true' in message_from_zulip) \
    and id_vk_bool == 'False':
    zulip_sender(
        'Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    vk_sender(id_chat,
        'Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("id_vk_bool: False", "id_vk_bool: True")
    f.write(all_text)

elif start is True and ('/include email false' in message_from_zulip) \
    and user_email_bool == 'True':
    zulip_sender('Теперь в сообщении не будет указываться email пользователя')
    vk_sender(id_chat, 'Теперь в сообщении не будет указываться email пользователя')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("user_email_bool: True", "user_email_bool: False")
    f.write(all_text)
elif start is True and ('/include subject false' in message_from_zulip) \
    and subject_bool == 'True':
    zulip_sender('Теперь в сообщении не будет указываться тема, из которой получено сообщение')
    vk_sender(id_chat, 'Теперь в сообщении не будет указываться тема, из которой получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("subject_bool: True", "subject_bool: False")
    f.write(all_text)
elif start is True and ('/include display_recipient false' in message_from_zulip) \
    and display_recipient_bool == 'True':
    zulip_sender(
        'Теперь в сообщении не будет указываться название канала, из которого получено сообщение')
    vk_sender(id_chat,
        'Теперь в сообщении не будет указываться название канала, из которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("display_recipient_bool: True", "display_recipient_bool: False")
    f.write(all_text)

```

```

elif start is True and ('/include id false' in message_from_zulip) \
    and id_vk_bool == 'True':
    zulip_sender(
        'Теперь в сообщении не будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    vk_sender(id_chat,
        'Теперь в сообщении не будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("id_vk_bool: True", "id_vk_bool: False")
        f.write(all_text)

# обработка сообщения из Вконтакте
if "VK156324" in data and (old_text_vk is None or not(text in old_text_vk)):
    length_data = len(data)
    message_from_vk = data[8:length_data]
    l2 = len(message_from_vk)
    pos = message_from_vk.find("\n")
    old_text_vk = message_from_vk[pos+1:l2]
    if '/start' in message_from_vk and start is False:
        vk_sender(id_chat, 'Бот успешно запущен')
        vk_sender(id_chat, "Чтобы узнать весь список команд бота напишите /help")
        zulip_sender('Бот успешно запущен')
        zulip_sender("Чтобы узнать весь список команд бота напишите /help")
        start = True
    elif start is True and ('/stop' in message_from_vk):
        stop = False
        vk_sender(id_chat, 'Бот прекратил свою работу')

    zulip_sender('Бот прекратил свою работу')
elif start is True and '/' != message_from_vk[0]:
    zulip_sender(message_from_vk)
elif start is False:
    vk_sender(id_chat, 'Бот не запущен. Напишите "/start" в эту беседу')
elif start is True and ('/help' in message_from_vk):
    vk_sender(id_chat, help_message)
elif start is True and ('/status' in message_from_vk):
    with open("Data/Settings.txt", "r") as f:
        status = f.read()
    vk_sender(id_chat, status)
elif start is True and ('/include email true' in message_from_vk) \
    and user_email_bool == 'False':
    vk_sender(id_chat, 'Теперь в сообщении будет указываться email пользователя')
    zulip_sender('Теперь в сообщении будет указываться email пользователя')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("user_email_bool: False", "user_email_bool: True")
        f.write(all_text)
elif start is True and ('/include subject true' in message_from_vk) \
    and subject_bool == 'False':
    vk_sender(id_chat,
        'Теперь в сообщении будет указываться тема, из которой получено сообщение')
    zulip_sender('Теперь в сообщении будет указываться тема, из которой получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("subject_bool: False", "subject_bool: True")
        f.write(all_text)

```

```

elif start is True and ('/include display_recipient true' in message_from_vk) \
    and display_recipient_bool == 'False':
    vk_sender(id_chat,
        'Теперь в сообщении будет указываться название канала, из которого получено сообщение')
    zulip_sender('Теперь в сообщении будет указываться название канала, из которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("display_recipient_bool: False", "display_recipient_bool: True")
        f.write(all_text)
elif start is True and ('/include id true' in message_from_vk) \
    and id_vk_bool == 'False':
    vk_sender(id_chat, 'Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    zulip_sender('Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("id_vk_bool: False", "id_vk_bool: True")
        f.write(all_text)

elif start is True and ('/include email false' in message_from_vk) \
    and user_email_bool == 'True':
    vk_sender(id_chat, 'Теперь в сообщении не будет указываться email пользователя')
    zulip_sender('Теперь в сообщении не будет указываться email пользователя')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("user_email_bool: True", "user_email_bool: False")
        f.write(all_text)

elif start is True and ('/include subject false' in message_from_vk) \
    and subject_bool == 'True':
    vk_sender(id_chat,
        'Теперь в сообщении не будет указываться тема, из которой получено сообщение')
    zulip_sender('Теперь в сообщении не будет указываться тема, из которой получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("subject_bool: True", "subject_bool: False")
        f.write(all_text)
elif start is True and ('/include display_recipient false' in message_from_vk) \
    and display_recipient_bool == 'True':
    vk_sender(id_chat,
        'Теперь в сообщении не будет указываться название канала, из которого получено сообщение')
    zulip_sender(
        'Теперь в сообщении не будет указываться название канала, из которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()
    with open("Data/Settings.txt", "w") as f:
        all_text = all_text.replace("display_recipient_bool: True", "display_recipient_bool: False")
        f.write(all_text)
elif start is True and ('/include id false' in message_from_vk) \
    and id_vk_bool == 'True':
    vk_sender(id_chat, 'Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    zulip_sender('Теперь в сообщении будет указываться id пользователя Вконтакте,'
        ' от которого получено сообщение')
    with open("Data/Settings.txt", "r") as f:
        all_text = f.read()

```

```
with open("Data/Settings.txt", "w") as f:
    all_text = all_text.replace("id_vk_bool: True", "id_vk_bool: False")
    f.write(all_text)
```

Setup.py

```

import os

# загрузка всех необходимых для работы программы библиотек
os.system("pip install tk")
os.system("pip install requests")
os.system("pip install vk_api")
os.system("pip install zulip")

import requests
from tkinter import *

def save_file(url, dir):
    """Функция скачивания файлов расширение .py. На вход подается ссылка на скачивание и директория."""
    r = requests.get(url, allow_redirects=True)
    file_name = url.split("/")[-1]
    dir = dir + '/' + file_name
    with open(dir, "wb") as f:
        f.write(r.content)

def make_file(name, text):
    """Функция создания текстовых файлов с информацией, необходимой для работы ботов. На вход подается имя файла и его содержимое."""
    dir = os.getcwd() + "/Data/" + name
    with open(dir, "w") as f:
        f.write(text)

def get_token():
    """Функция получения токена группы Вконтакте через поле ввода в окне."""
    global token
    token = entry.get()
    root.destroy()

def get_id_group():
    """Функция получения id группы Вконтакте через поле ввода в окне."""
    global id_group
    id_group = entry.get()
    root.destroy()

def get_way_to_file():
    """Функция получения токена пути к файлу zuliprc через поле ввода в окне."""
    global way_to_file
    way_to_file = entry.get()
    root.destroy()

def enter_data():
    """Функция ввода данных, полученных от пользователя в файл private_inform.txt."""
    global token, id_group, way_to_file
    with open(os.getcwd() + "/Data/private_inform.txt", "r") as f:
        line = f.readline()
        line1 = line[:len(line)-1]
        line = f.readline()
        line2 = line[:len(line)-1]
        line = f.readline()
        line3 = line[:len(line)]
    with open(os.getcwd()+"/Data/private_inform.txt", "w") as f:

```

```

f.write(line1 + token + "\n" + line2 + id_group + "\n" + line3 + way_to_file)

# блок для проверки наличия папки Data, нужный для вывода ошибки о том, что бот уже установлен в этой папке
try:
    os.mkdir('Data')
except FileExistsError:
    root = Tk()
    root.title("WARNING")
    w = root.winfo_screenwidth()
    h = root.winfo_screenheight()
    w = w // 2 - 150
    h = h // 2 - 50
    root.geometry('300x100+{}+{}'.format(w, h))
    Label(text="Вы уже установили бота в эту папку", font=("Times New Roman", 12), width=50, height=3).pack()
    button = Button(text="OK", width=20)

    def button_exit(event):
        root.destroy()
        exit()
    button.bind("<Button-1>", button_exit)
    button.pack()
    root.mainloop()

# ссылки для скачивания файлов .py
link1 = "https://raw.githubusercontent.com/HCK-git/Vk-to-Zulip/main/MainBot.py"
link2 = "https://raw.githubusercontent.com/HCK-git/Vk-to-Zulip/main/VK_getter.py"
link3 = "https://raw.githubusercontent.com/HCK-git/Vk-to-Zulip/main/Zulip_getter.py"
directory = os.getcwd()
token = "None"

id_group = "None"
way_to_file = "None"

save_file(link1, directory)
save_file(link2, directory)
save_file(link3, directory)

# создание текстовых файлов в папке Data
make_file("Settings.txt", "user_email_bool: False\nsubject_bool: False\ndisplay_recipient_bool: False\n"
        "id_vk_bool: False")
make_file("private_inform.txt", "token: \nid_group: \nway_to_config: ")
make_file("id_chat.txt", " ")

# создание окна для получение токена группы Вконтакте
root = Tk()
root.title("token")
w = root.winfo_screenwidth()
h = root.winfo_screenheight()
w = w // 2 - 150
h = h // 2 - 50
root.geometry('300x100+{}+{}'.format(w, h))
entry = Entry()
entry.pack(pady=10)
Button(text="Ввести", command=get_token).pack()

root.mainloop()

# создание окна для получение id группы Вконтакте
root = Tk()
root.title("id_group")

```

```

root.geometry('300x100+{}+{}'.format(w, h))
entry = Entry()
entry.pack(pady=10)
Button(text="Ввести", command=get_id_group).pack()

root.mainloop()

# создание окна для получение пути к файлу zuliprc
root = Tk()
root.title("way_to_file")
root.geometry('300x100+{}+{}'.format(w, h))
entry = Entry()
entry.pack(pady=10)
Button(text="Ввести", command=get_way_to_file).pack()

root.mainloop()

enter_data()

```