

# СОЗДАНИЕ ЧАТ – БОТА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Ашпетова Мария Владимировна

Информационных систем и технологий, информационных систем и технологий, Поволжский Государственный Университет Телекоммуникаций и Информатики, Самара, Россия

## **Аннотация**

Актуальность лабораторной работы: заключается в том, что язык программирования Python сейчас набирает обороты и становится все более популярным.

Цель лабораторной работы: создать чат - бота, написанный на языке программирования Python.

Задачи работы:

1. Ознакомиться с предметной областью;
2. Поэтапно расписать создание чат - бота.

Предмет исследования: социальная сеть ВКонтакте.

Объект исследования: чат - бот.

Основные результаты: чат - бот ВКонтакте по запросу будет выводить точное время по гринвичу, курс доллара к рублю и определять месторасположение.

## **Ключевые слова**

Чат – бот, языки программирования, Python, чат – бот в социальных сетях, время по гринвичу, лабораторная работа.

## **1. Введение**

Технологии развиваются в бешеном темпе. Искусственный интеллект – уже не новинка для современных людей. «Умные» дома, автомобили различные бытовые принадлежности, которые упрощают жизнь человека. Одной из таких нанотехнологий стала виртуальная машина-собеседник – чат бот.

Сначала разберёмся, что такое чат бот. И так чат бот – это виртуальная машина, которая запрограммирована для общения с одним или несколькими пользователями. Говоря простыми словами, это робот-собеседник.

Чат бот не только умеет вести беседу, но и станет хорошим помощником в решение многих задач, время на которые не всегда есть.

Понять, для чего нужен чат бот можно, уже исходя из определения. Это не только собеседник, но и хороший помощник, а для некоторых, особо занятых людей, даже незаменимый работник. Причём на бесплатной основе.

Но чат боты используются не только для общения. Часто их устанавливают на сайт, для привлечения клиентов и упрощения работы компании.

Вы можете задать вопрос в чате, но тогда он попросит оставить контактные данные (e-mail, номер телефона). Некоторые боты на сайтах запрограммированы для ответов на некоторые вопросы, например самые часто задаваемые.

#### *5 преимуществ ботов*

- Интерактивные (дают точные ответы на точные запросы);
- Не занимают памяти на телефоне;
- Не требуют дополнительной установки на смартфон или ПК, в отличие от стороннего приложения;
- Позволяют обмениваться данными с пользователем за долю секунды;
- Сокращают время ожидания клиента.

#### **2. Методы и методологии**

В данном примере показан чат – бот ВК, который по запросу показывают курс доллара к рублю, точное время по гринвичу, а также определяет локацию.

1. Импорт модуля vk\_api для доступа к API ВКонтакте

```
import vk_api
from vk_api.longpoll import VkLongPoll, VkEventType
```

2. Используем модули: random, datetime, bs4

Модуль random - для случайного целого числа при отправке сообщения пользователю(требования ВК);

Модуль datetime - для определения текущей даты и времени;

Модуль bs4 - для парсинга страницы google, содержащей информацию о текущем курсе доллара к рублю;

```
from random import randint
from datetime import datetime
from bs4 import BeautifulSoup
```

А также модули requests, который служит для получения данных из источника, и модуль json, необходимый для преобразования типа данных dict в json.

```
import requests
import json
```

### 3. Добавляем код для получения клавиатуры для бота

```
def get_keyboard():
    return json.dumps({'one_time': False,
                      'buttons': [
                          [{"action": {"type": "location"}}],
                          [{"action": {"type": "text", "label": "Курс доллара к рублю"}, "color": "negative"}],
                          [{"action": {"type": "text", "label": "Время по гринвичу"}, "color": "positive"}]
                      ])

def get_currency_price():
```

### 4. Указываем ссылку на нужную страницу

«DOLLAR\_RUB =

'https://www.google.com/search?sxsrf=ALeKk01NWm6viYijAo3HXYOEQUyDE  
DtFEw%3A1584716087546&source=hp&ei=N9l0XtDXHs716QTcuaXoAg&q=%  
D0%B4%D0%BE%D0%BB%D0%BB%D0%B0%D1%80+%D0%BA+%D1%80  
%D1%83%D0%B1%D0%BB%D1%8E&oq=%D0%B4%D0%BE%D0%BB%D0  
%BB%D0%B0%D1%80+&gs\_l=psy»

### 5. Добавляем заголовки для передачи вместе с URL

```
headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149 Safari/537.36'}
```

### 6. Парсим (обрабатываем всю страницу целиком) всю страницу

```
full_page = requests.get(DOLLAR_RUB, headers=headers)
```

7. Разбираем через BeautifulSoup (это библиотека Python для извлечения данных из файлов HTML и XML)

```
soup = BeautifulSoup(full_page.content, 'html.parser')
```

8. Получаем нужное для нас значение и возвращаем его

```
convert = soup.findAll("span", {"class": "DF1fde", "class": "SwhCTb", "data-precision": 2})  
return convert[0].text
```

```
class BotEngine:  
    def __init__(self):
```

9. Устанавливаем токен и авторизуемся с помощью API ВКонтакте (как это сделать – подробно показано после примера реализации а «Приложении А»)

```
self.token = '467e5d0f5925cdb47277d6f9a71108da91e1e17bb89563f3dde033121e31f63ae2aff77f5d7418e96333e'  
self.vk = vk_api.VkApi(token=self.token)  
  
self.start()  
  
def write_msg(self, user_id, message, keyboard=None):  
    """ Отправка сообщения пользователю. Если не передать методу объект клавиатуры, то клавиатура исчезнет """  
    if keyboard is None:  
        keyboard = {}  
    self.vk.method('messages.send',  
                  {'user_id': user_id,  
                   'message': message,  
                   'random_id': randint(0, 2147483646),  
                   'keyboard': keyboard})  
  
def start(self):  
    """ Запуск бота """  
    longpoll = VkLongPoll(self.vk)
```

10. Подключаемся к серверу ВК с помощью LongPoolAPI

11. Прописываем основной цикл

```
for event in longpoll.listen():
```

12. Условия:

Если пришло новое сообщение и оно имеет метку для бота:

- получаем текст сообщения

```
if event.type == VkEventType.MESSAGE_NEW and event.to_me:  
    request = event.text  
  
if request.lower() == 'начать':  
    self.write_msg(event.user_id, message='Приветствую!', keyboard=get_keyboard())  
elif request.lower() == 'курс доллара к рублю':  
    self.write_msg(event.user_id, f'1 Доллар США равно: {get_currency_price()} Российский рубль')  
elif request.lower() == 'время по гринвичу':  
    self.write_msg(event.user_id, f'Текущее время по гринвичу: {datetime.utcnow().strftime("%H:%M")}')
```

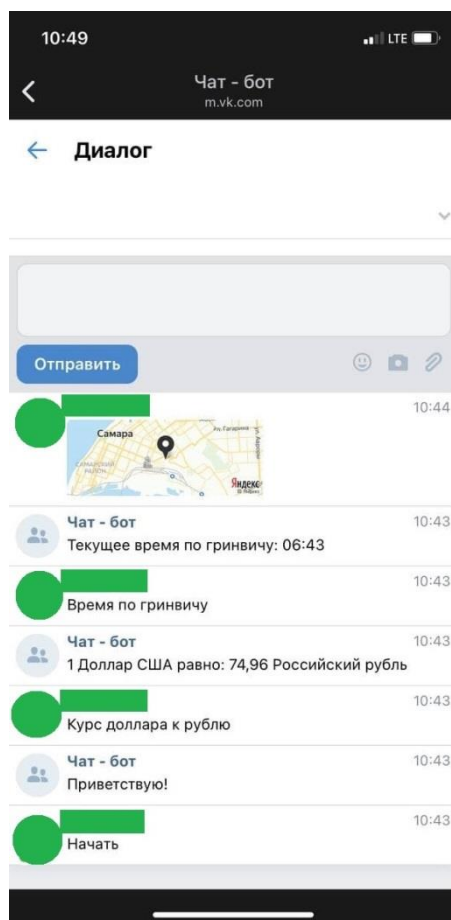
13. Создание экземпляра класса

```
app = BotEngine()
```

14. Запускаем бот

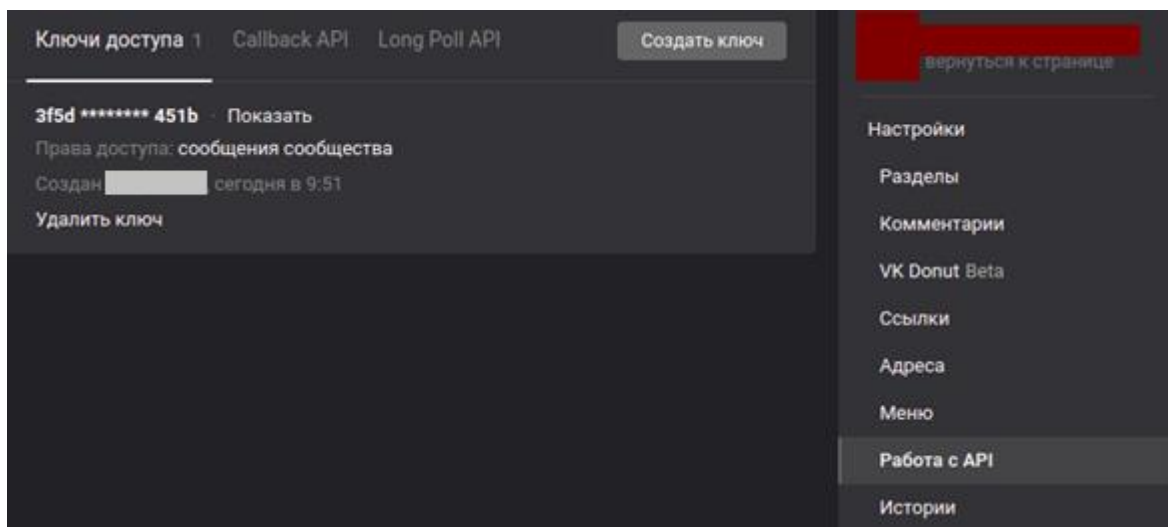
app.start()

### 3. Результаты

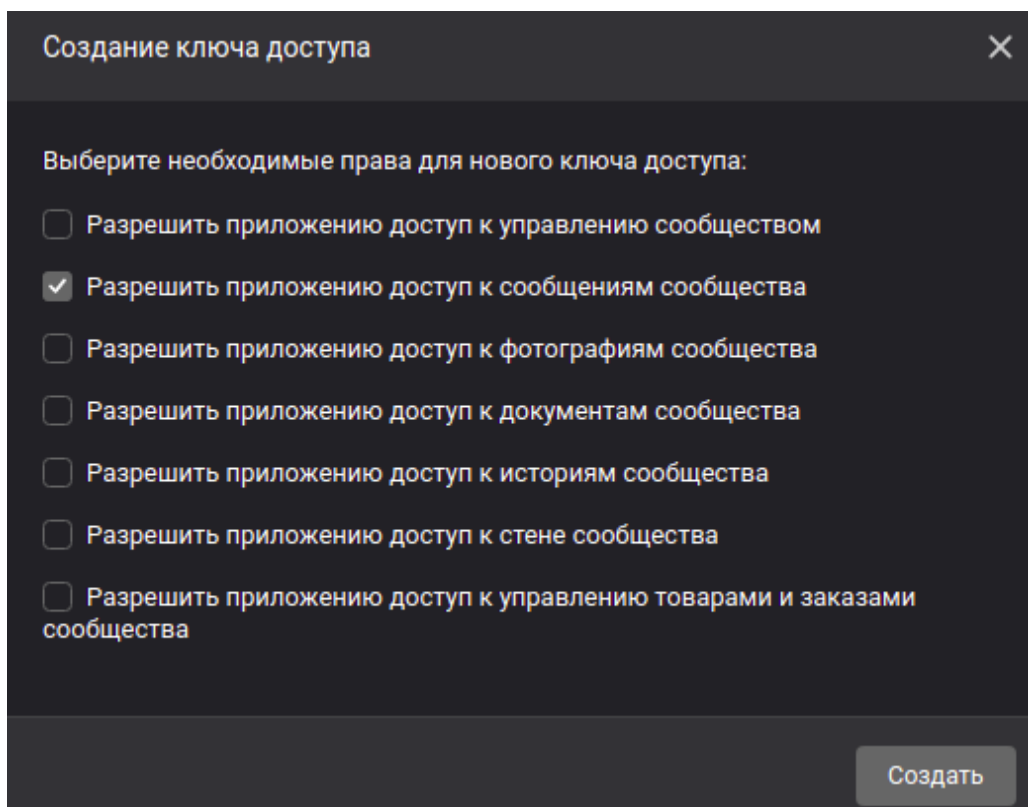


## Приложение А

1. Заходим на страничку ВК;
2. Создаём сообщество;
3. Заходим в его настройки. Выбираем вкладку “Работа с API”;

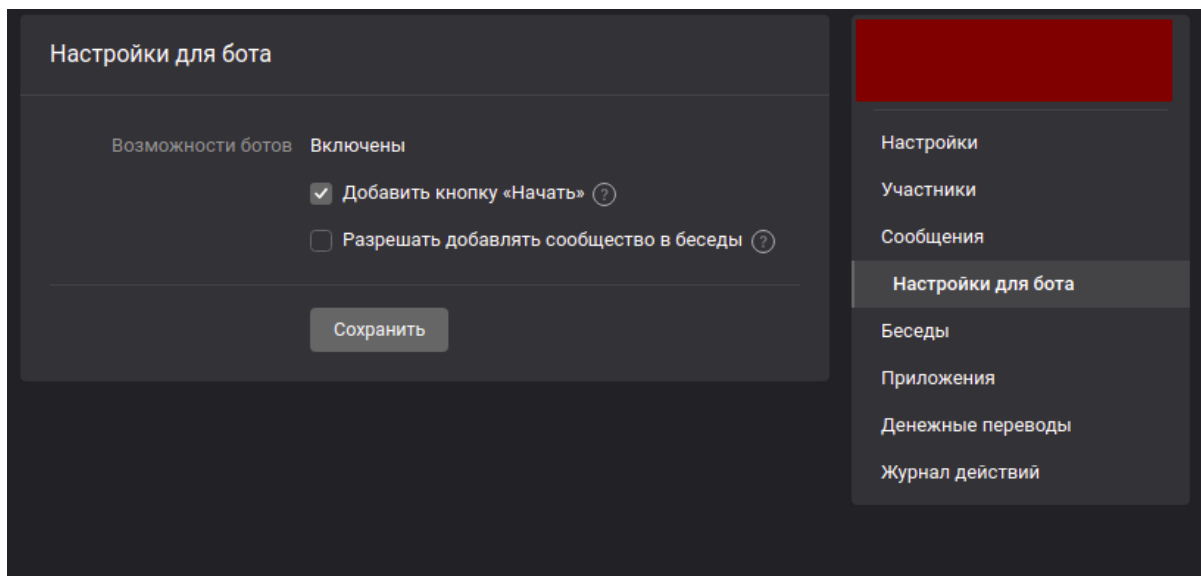


4. Создаём ключ (он же токен). В разрешениях указываем “Доступ к сообщениям сообщества”;



5. Переходим в раздел “Сообщения” и включаем сообщения сообщества;

6. Заходим во вкладку “Настройки для бота”: включаем возможности для ботов и отмечаем галочкой доступность кнопки “Начать”;



## Приложение Б

```
# Модуль random для случайного целого числа при отправке сообщения пользователю(требования ВК)
# Модуль datetime для определения текущей даты и времени
# Модуль bs4 для парсинга страницы google, содержащей информацию о текущем курсе доллара к рублю
from random import randint
from datetime import datetime
from bs4 import BeautifulSoup

# Модуль requests служит для получения данных из источника
# Модуль json нужен для преобразования типа данных dict в json
import requests
import json

# Получение клавиатуры для бота
def get_keyboard():
    return json.dumps({'one_time': False,
                      'buttons': [
                          [{"action": {"type": "location"}}],
                          [{"action": {"type": "text", "label": "Курс доллара к рублю"}, "color": "negative"}],
                          [{"action": {"type": "text", "label": "Время по гринвичу"}, "color": "positive"}]
                      })

def get_currency_price():
    # Ссылка на нужную страницу
    DOLLAR_RUB = 'https://www.google.com/search?sxsrf=ALekK01Nwm6viYijAo3HXyOEQyUEDtFEw%3A1584716087546&source=hp&ei=N910XtDXHs716QTcuaXoAg&q=%D0%B
    # Заголовки для передачи вместе с URL
    headers = {
        'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149 Safari/537.36'
    }

    # Парсим всю страницу
    full_page = requests.get(DOLLAR_RUB, headers=headers)

    # Разбираем через BeautifulSoup
    soup = BeautifulSoup(full_page.content, 'html.parser')

    # Получаем нужное для нас значение и возвращаем его
    convert = soup.findAll("span", {"class": "DF1fde", "class": "SwHCTb", "data-precision": 2})
    return convert[0].text

class BotEngine:
    def __init__(self):
        # установка токена и авторизация с помощью API ВКонтакте
        self.token = '467e5d0f5925cdb47277d6f9a71108da91e1e17bb89563f3dde033121e31f63ae2aff77f5d7418e96333e'
        self.vk = vk_api.VkApi(token=self.token)

        self.start()

    def start(self):
        """ Запуск бота """
        longpoll = VkLongPoll(self.vk) # Подключаемся к серверу ВК с помощью LongPollAPI
        # Основной цикл
        for event in longpoll.listen():
            # Если пришло новое сообщение и оно имеет метку для бота
            if event.type == VkEventType.MESSAGE_NEW and event.to_me:
                request = event.text # получаем текст сообщения
                if request.lower() == 'начать':
                    self.write_msg(event.user_id, message='Приветствую!', keyboard=get_keyboard())
                elif request.lower() == 'курс доллара к рублю':
                    self.write_msg(event.user_id, f'1 Доллар США равно: {get_currency_price()} Российский рубль')
                elif request.lower() == 'время по гринвичу':
                    self.write_msg(event.user_id, f'Текущее время по гринвичу: {datetime.utcnow().strftime("%H:%M")}')
```

```
app = BotEngine() # Создание экземпляра класса
app.start() # Запуск бота
```



## Список использованных источников

- 1) Interkassa [Электронный ресурс] : статья «Все о чат-ботах: преимущества, типы и схема работы» - Загл. с титул. Экрана. Режим доступа: <https://www.interkassa.com/blog/vse-o-chat-botah-preimushchestva-tipy-i-shema-raboty/>.
- 2) Carrot quest [Электронный ресурс] : статья «Что такое чат – боты и какие они бывают» - Загл. с титул. Экрана. Режим доступа: <https://www.carrotquest.io/blog/chatbot-types/>.
- 3) Яндекс Дзен [Электронный ресурс] : статья «Что такое чат – боты и зачем нужны» - Загл. с титул. Экрана. Режим доступа: <https://zen.yandex.ru/media/id/5c17ac61acbbe800aa5d991f/chto-takoe-chatboty-i-zachem-oni-nujny-5cb6f94a362abf00b3fсеbed>.
- 4) VC.RU [Электронный ресурс] : статья «Как устроены чат - боты» - Загл. с титул. Экрана. Режим доступа: <https://vc.ru/services/142965-kak-ustroeny-chat-boty-i-zachem-oni-nuzhny>.
- 5) WhatsApp [Электронный ресурс] : статья «Что такое боты в ВК и как создать самому» - Загл. с титул. Экрана. Режим доступа: <https://wotsap24.ru/faq/chto-takoe-boty-v-vatsap-i-kak-sozdat-samomu>.