МБОУ «Лицей №159»

**Тема проекта**

**«Исследование возможности использования чат-ботов для упрощения процесса записи домашнего задания»**

**Тип проекта**: практико-ориентированный

**Автор проекта:**

Борисов Виктор, 11Б класс

8-923-128-55-55, borisov\_vk@mail.ru

**ФИО научного руководителя:**

Омельченко Мария Михайловна ,

учитель информатики

МБОУ «Лицей №159»

8-913-923-31-89 ,

omelchenko1606@mail.ru

**Новосибирск, 2021 г**.

**Оглавление**

[**Вступление**](#_931lluly2l4n) **4**

[**Теоретическая часть**](#_hj8fs9ng9uad) **5**

[1.Определение](#_agx853gspxzo) 5

[2.История появления](#_hzg5z2r32uiu) 5

[3.Виды чат-ботов](#_ln6ul1vyau86) 6

[4.Возможности и сферы применения чат-ботов](#_knvh2zo2o02u) 7

[**Практическая часть**](#_ypw3atw1mfup) **8**

[Описание работы](#_x33sc537nm6e) 8

[**Используемые функции библиотеки vk\_api**](#_spsa601t9dmr) **9**

[1.Синхронизация бота с группой Вконтакте](#_h9fncdmuewzh) 9

[2.Получение основной информации о сообщении](#_hsw2ok9rwj2c) 10

[3.Отправка сообщений](#_h4e30g8yy2b2) 10

[**Обрабатываемые запросы**](#_7e8cqeyz371g) **11**

[1.Запись домашнего задания](#_17dp8vu) 11

[2.Отправка домашнего задания по предмету.](#_rm1jzki8b29p) 12

[3.Отправка домашнего задания по определённым дням недели.](#_rttq7duza8xo) 13

[4. Прошлое домашнее задание](#_d2o3gqke72y) 14

[**Сервер**](#_hdup6hmqqsbu) **15**

[**Статистические показатели**](#_2rwbox6z3qgx) **16**

[Заключение](#_26in1rg) **19**

[Список иллюстраций](#_lnxbz9) **20**

[**Список использованной литературы**](#_5aal71z79a6h) **21**

[**Приложение(исходный код)**](#_29jswn9imxh8) **22**

# Вступление

Люди всегда стремились упростить себе жизнь различными способами. Уже никто не будет отрицать, что большинство учащихся школ не носит дневники, а вместо этого используют телефон для более комфортного записывания домашнего задания. Сфотографировать одному ученику и отправить в общую беседу в социальных сетях гораздо удобнее, нежели каждому тратить время, записывая по отдельности. Но проблема в том, что за неделю накапливается множество домашних заданий и отсутствие систематизации данных создает большое количество неудобств каждый день.

Поэтому **цель проекта -** создание чат-бота, способного записывать и отправлять информацию о домашних заданиях учащимся.

**Задачи проекта**:

1. Создать программу для социальной сети Вконтакте, способную записывать и отправлять информацию о домашних заданиях.
2. Исследовать практичность применения программы в рамках 10-11 классов при помощи статистических данных, собранных во время периода эксплуатации.

# Теоретическая часть

## 1.Определение

**Чат-бот** — это виртуальный собеседник, программа, которая может решать типовые задачи: задавать вопросы и отвечать на них, искать информацию по запросу и выполнять простейшие поручения. Чат-боты можно классифицировать по платформе внедрения, технологии разработки, способу общения с пользователями и функциональности. Они могут быть простыми, как, например, чат-боты в Telegram, которые выполняют только одну задачу. И могут обладать искусственным интеллектом, как Siri.

Чат-боты можно настроить в мессенджерах и соцсетях (Telegram, Viber, Facebook, Вконтакте, WhatsApp и другие), на сайтах, в мобильных приложениях, системах управления (Siri), голосовых ассистентах (Алиса) и так далее.

В зависимости от принципа работы, выделяют обучаемых и необучаемых чат-ботов. Необучаемые чат-боты работают по заранее заданному сценарию и предлагают готовые варианты ответа. Обучаемые чат-боты разработаны с помощью технологии искусственного интеллекта. Они могут задавать уточняющие вопросы, чтобы узнать цель поиска, и предлагают товар, исходя из ответов покупателя. Обучаемые чат-боты хранят уникальные поисковые запросы от разных пользователей и совершенствуются, отвечая все более точно.

## 2.История появления

В 1966 году Йозеф Вейценбаум разработал программу под названием ELIZA. Фактически, это был один из первых чат-ботов или программ для общения. Людям, которые общались с ней, казалось, что они разговаривают с психоаналитиком. Программа привлекла большое внимание, главным образом потому, что она стала первой, которая смогла «обмануть» человека.

Сам Вейценбаум четко заявил, что ELIZA абсолютно неразумна, поскольку она не совсем точно анализировала содержание высказываний и вопросы человека. И все потому, что вычислительной мощности, доступной в то время (1960-е годы), было недостаточно.

Сейчас же многие бренды используют чат-боты для оповещения клиентов об акциях, скидках, обновлениях, для принятия оплаты и для многих других целей. Например, для оплаты покупки в интернет магазине, вам необходимо будет отправить чат-боту определенное SMS.

## 3.Виды чат-ботов

По способу общения чат-боты бывают голосовыми и текстовыми. Текстовые боты распознают ключевые слова. Голосовые чат-боты взаимодействуют с пользователем с помощью устного общения, могут распознавать и воспроизводить человеческую речь. Голосовые боты можно использовать, когда требуется совершать или принимать тысячи типовых звонков в день. Они могут информировать, проводить опросы, уточнять детали заказов, обрабатывать входящие звонки и решать простые задачи, описанные в базе данных.

По функциональности можно выделить следующие виды чат-ботов:

* Чат-бот для продаж — консультирует покупателей и помогает им подобрать нужный товар, уведомляет о статусе заказа, рассказывает об акциях и скидках.
* Лид-бот — собирает данные посетителей сайта, предлагает записаться на демонстрацию продукта.
* Транзакционный бот — выполняют различные транзакции: размещение заказа, резервирование, денежные переводы.
* Бот-информатор — отвечает на запросы, предоставляет информацию о вариантах перелётов, ценах, и т. п.
* Чат-бот для поддержки — помогает в вопросах использования продукта или услуги.
* Бот-ассистент — интегрируется с другими платформами и помогает пользователю решать сразу несколько задач, такие как поиск в Google, установка напоминаний, отбор новостей. [5]

Наиболее часто чат-ботов используют в продажах, саппорте и маркетинге. Чат-боты справляются с рутинными операциями, которые можно свести к конкретному алгоритму, ищут и агрегируют данные, распространяют информацию. Всё это повышает производительность разных команд и улучшает клиентский сервис.

## 4.Возможности и сферы применения чат-ботов

Чат-боты:

* отвечают на простые вопросы;
* помогают найти нужные сведения на сайте;
* дают персональные рекомендации и консультируют;
* предлагают записаться на демо;
* собирают данные посетителей сайтов;
* собирают отзывы клиентов;
* бронируют билеты, столики в ресторане, номера в отеле;
* выполняют инвентаризацию и отслеживают заказы;
* обрабатывают денежные транзакции.

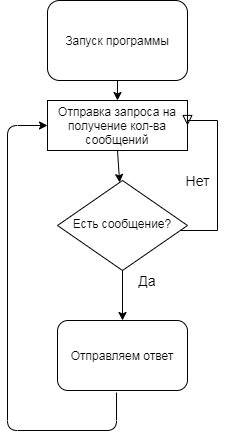
Сферы, в которых чаще всего используют чат-ботов:

* банки — онлайн-расчеты процентов по займам и кредитам;
* операторы связи — техническая поддержка;
* страхование — помощь в заполнении форм и заявок, консультация по страхованию;
* онлайн-торговля — консультирование по доставке, оплате, адресам точек выдачи;
* туризм — агрегатор предложений, рассылка горящих туров, бронирование;
* образование — рассылки, приглашение на вебинары;
* государственные услуги — сбор жалоб, быстрый доступ к публичным данным;
* HR — подбор подходящих резюме, автоматизация задач внутри компании.

# Практическая часть

## Описание работы

При разработке своего чат-бота я использовал следующую последовательность действий:

1. Отправляем запрос на получение количества сообщений.
2. Есть непрочитанное сообщение:

2.1. Анализируем сообщение, характеризуем запрос.

2.2. Отправляем соответствующий ответ.

2.3. Пункт 1.

3. Нет непрочитанных сообщений:

3.1.Пункт 1.

Вышеуказанный алгоритм в фрагменте основного кода программы на Python 3.7:

While true :

try:

#основной код. Если есть ответ на запрос, то variable\_for\_exception=1

if variable\_for\_exception == 0:

#Отправка сообщения о неизвестном запросе

variable\_for\_exception=0

# Используемые функции библиотеки vk\_api

Чтобы получить все функции Вконтакте, необходимо использовать API. В Python есть подходящая библиотека vk\_api.

## 1.Синхронизация бота с группой Вконтакте

Подключение к библиотеке. Указанный алгоритм в фрагменте основного кода программы на Python 3.7:

import vk\_api

token = ‘’ #Ключ доступа к боту

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

vk.\_auth\_token()

## 2.Получение основной информации о сообщении.

Указанный алгоритм в фрагменте основного кода программы на Python 3.7

messages = vk.method("messages.getConversations",

{"filter": "unanswered"})

Переменная messages содержит массив с информацией о сообщении.

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

*id* - id пользователя, отправившего сообщение

*id\_message = messages["items"][0]["last\_message"]["id"]*

*id\_message-* id сообщения относительно бота. Соответственно, первому сообщению, написанному боту с момента его создания, будет присвоено id=1

msg\_txt= messages["items"][0]["last\_message"]["text"].lower().split()

msg\_txt *-* текст сообщения последнего сообщения в формате списка из слов в нижнем регистре.

## 3.Отправка сообщений

Указанный алгоритм в фрагменте основного кода программы на Python 3.7

vk.method("messages.send",{"peer\_id":id,"message":text,

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

*id* - id пользователя, получающего сообщение.

text - текст отправляемого сообщения.

forward\_messages - идентификаторы пересылаемых сообщений, перечисленные через запятую. Перечисленные сообщения отправителя будут отображаться в теле письма у получателя. Не более **100** значений на верхнем уровне, максимальный уровень вложенности: **45**, максимальное количество пересылаемых сообщений **500**

# Обрабатываемые запросы

## 1.Запись домашнего задания

Каждому школьному предмету выделяется отдельная строка в файле hometask.txt, куда будут записываться id сообщений, чтобы при запросе можно было переслать сообщение

subjects={"общество":0,"физика":1,"алгебра":2,"геометрия":3,"литра":4,"русский":5,"география":6,"биология":7, "англ1":8, "англ2":9, "информатика":10, "история":11, "химия":12, "обж":13, "литра(род)":14}

Для записи домашнего задания пользователю необходимо ввести название школьного предмета и после, в этом же сообщении, написать домашнее задание. Т.к бот пересылает сообщения, можно прикреплять фото, видео, аудио, документы, другие пересланные сообщения.

Соответствующий фрагмент программы:

def file\_change(subject\_name, hmtsk,files\_name):#Функция записи домашнего задания в файл

print("Запись в файл")

hometask\_file = open(files\_name, "r", encoding="utf-8")

hometask\_lines = hometask\_file.readlines()

hometask\_file.close()

hometask\_file = open(files\_name, "w", encoding="utf-8")

old\_hmtsk = hometask\_lines[subjects[subject\_name]]

hometask\_lines[subjects[subject\_name]] = hmtsk + " \n"

for i in range(len(hometask\_lines)):

hometask\_file.write(hometask\_lines[i])

hometask\_file.close()

return old\_hmtsk.strip()

Зацикленная часть программы:

if msg\_txt[0] in subjects:

file\_change(msg\_txt[0], file\_change(msg\_txt[0], str(id\_message), 'hometask.txt'), 'hmtsk\_last.txt')

#Сначала программа переносит старые данные в файл hmtsk\_last.txt, затем записывает новые в hometask.txt.

vk.method("messages.send",{"peer\_id": id, "message": "дз получено", "random\_id": random.randint(1, 2147483647)})

variable\_for\_exception = 1

print("Дз записано")

## 2.Отправка домашнего задания по предмету.

Если пользователю нужно получить информацию по одному предмету, то ему необходимо написать боту: «дз + название школьного предмета». Например, «дз алгебра».

Соответствующий фрагмент программы:

def files\_send(subject\_name,files\_name):#Функция получения id сообщений по названию предмета

print("Из файла достаем id")

hometask\_file = open(files\_name, "r",encoding="utf-8")

hometask\_lines = hometask\_file.readlines()

hometask = hometask\_lines[subjects[subject\_name]]

hometask\_file.close()

return hometask

Соответствующая зацикленная часть программы:

if msg\_txt[0]=="дз" and msg\_txt[1] in subjects:

vk.method("messages.send",{"peer\_id": id, "message": "Распишитесь", "random\_id": random.randint(1, 2147483647),"forward\_messages":files\_send(msg\_txt[1],'hometask.txt')})

variable\_for\_exception = 1

print("Дз отправлено")

## 3.Отправка домашнего задания по определённым дням недели.

day\_lesson = {"пн":["алгебра", "литра", "физика", "обж"],

"вт" : ["геометрия", "англ1", "англ2", "общество", "физика"],

"ср": ["геометрия", "англ1", "англ2", "русский", "информатика", "литра"],

"чт": ["информатика", "история", "литра(род)"],

"пт": ["англ1", "англ2", "алгебра", "биология", "география"],

"сб": ["физика", "алгебра", "химия"] }

day\_lesson - словарь, ключи которого - дни недели в сокращенной форме и соответствующие им значения - списки уроков в эти дни.

Для того, чтобы получить домашнее задание на определенный день недели, пользователь должен ввести первые две согласные буквы нужного дня, например: пн .

Функция, отправляющая домашнее задание по тому, какой день был в нее введен:

def day\_send(day):

print("Отправка домашнего задания по дням")

message\_id = ""

for i in range(len(day\_lesson[day])):

message\_id=files\_send(day\_lesson[day][i],'hometask.txt')

vk.method("messages.send",{"peer\_id": id, "message": "", "random\_id": random.randint(1, 2147483647),"forward\_messages":message\_id})

Вследствие работы программы, бот отправит N-ое количество сообщений, в зависимости от запрашиваемого дня недели. Каждое сообщение представляет собой пересланное сообщение, id которого, было ранее записано в файл hometask.txt.

## 4. Прошлое домашнее задание

Благодаря функции ПДЗ (прошлое домашнее задание) пользователь может получить ДЗ, которое было обновлено, т.е прошлое домашнее задание. Это будет полезно ученикам, имеющим долги по домашним работам.

Зацикленный фрагмент программы:

if msg\_txt[0]=="пр" and msg\_txt[1] in subjects:

variable\_for\_exception = 1

vk.method("messages.send",{"peer\_id": id, "message": "Распишитесь", "random\_id": random.randint(1, 2147483647),"forward\_messages":files\_send(msg\_txt[1],'hmtsk\_last.txt')})

ПДЗ работает аналогично ЗДЗ (Запись домашнего задания, пункт №1), однако разница в том, что при отправке, id пересылаемого сообщения берется из файла hmtsk\_last.txt.

# Сервер

Очевидно, для постоянной, непрерывной работы программы необходима вычислительная машина. В нашем случае есть два варианта сервера:

1. Персональный компьютер
2. Облачная платформа

В первом случае вычислительные ресурсы не ограничены, однако, ваш персональный компьютер всегда должен быть включен . Этот вариант подразумевает большие затраты на электроэнергию, да и износ компьютера никто не отменял.

Во втором же случае сразу встал материальный вопрос. Но некоторые сайты, например pythonanywhere.com, предлагают хоть и бесплатную, но ограниченную вычислительную мощность . Однако время показало, что этого достаточно для стабильной работы программы.

# Статистические показатели

Благодаря встроенным функциям Вконтакте можно получить 2 следующие диаграммы о работе программы на протяжении месяца (в период с 1 по 27 марта 2020 года).

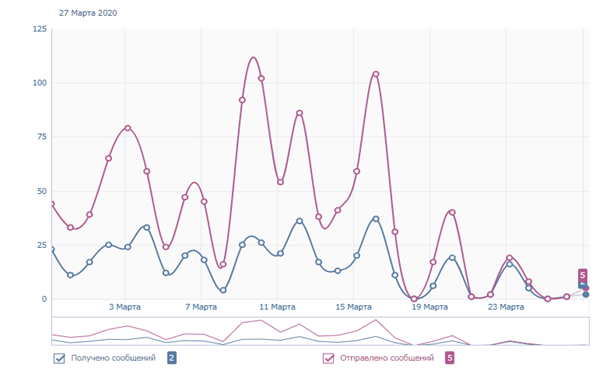


Рисунок 2; График, отображающий количество отправленных/полученных сообщений в течение дня за месяц

Красным цветом отмечены полученные сообщения, а синим - отправленные.

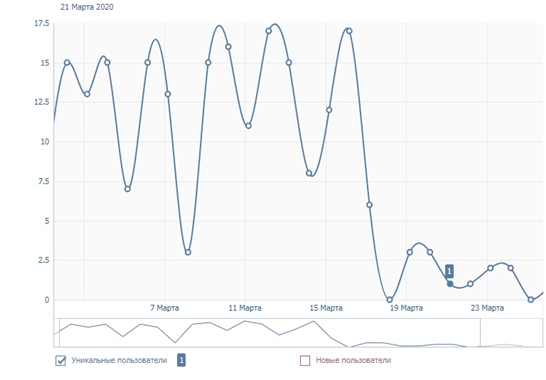


Рисунок 3; График, отображающий количество пользователей, написавших сообщение в течение дня за месяц

Сразу же можно заметить, что количество запросов резко снижается. Это объясняется началом весенних каникул.

В классе всего 21 человек, получается 18/21\*100%=85,7% учащихся пользуются чат-ботом, а значит программа актуальна и пользуется спросом

Также, добавив в программу переменные, считывающие количество использований определенных функций, удалось составить среднее ежемесячное соотношение популярности всех обрабатываемых запросов.

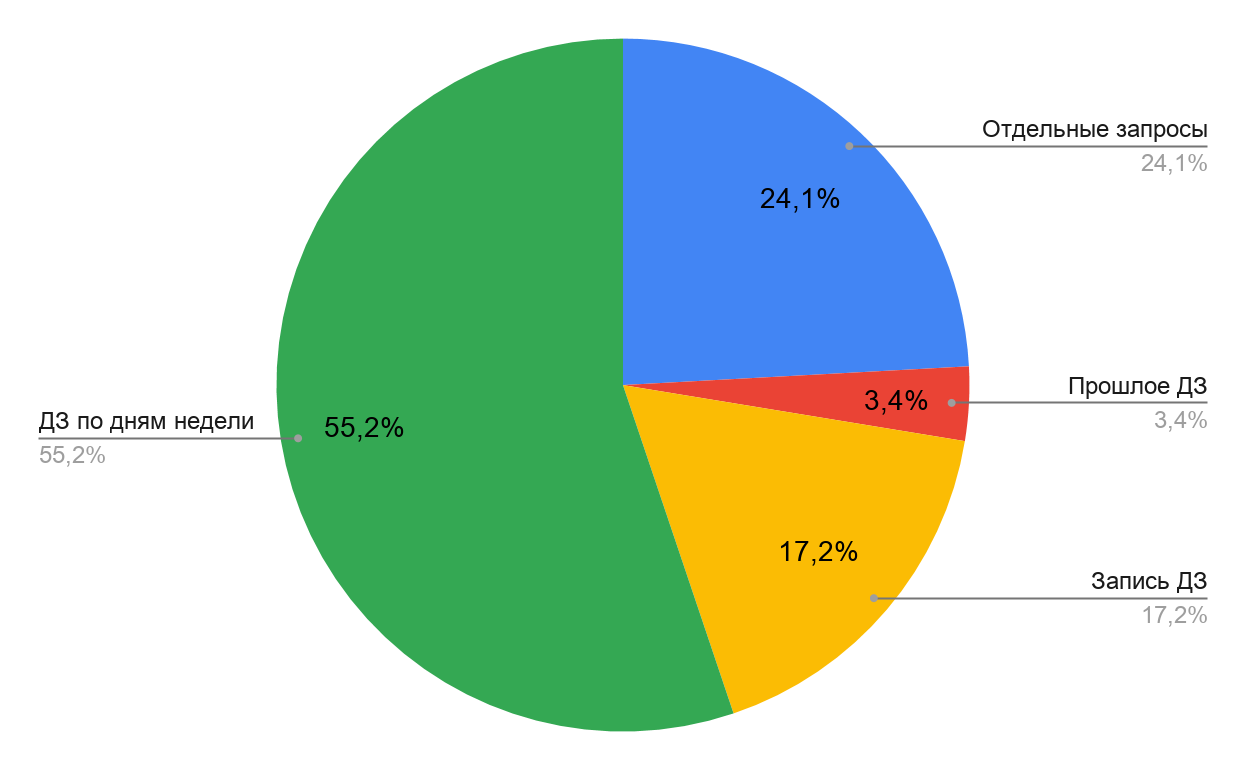


Рисунок 4; Диаграмма, отображающая популярность каждого из запросов.

Исходя из диаграммы можно сделать вывод: «ДЗ по дням недели » является наиболее популярным запросом. Это вызвано тем, что, введя всего лишь 2 символа (первые две согласные дня недели), пользователь получает всё домашнее задание на день недели (минимальные затраты энергии - максимальный результат). А «Прошлое ДЗ» , в свою очередь, является наименее популярным запросом, т.к является самым узконаправленным.

Также я провел опрос среди моих одноклассников в школьной беседе в социальной сети Вконтакте. Ниже представлены результаты ответа на вопрос:

«Часто ли вы пользуетесь ботом?»

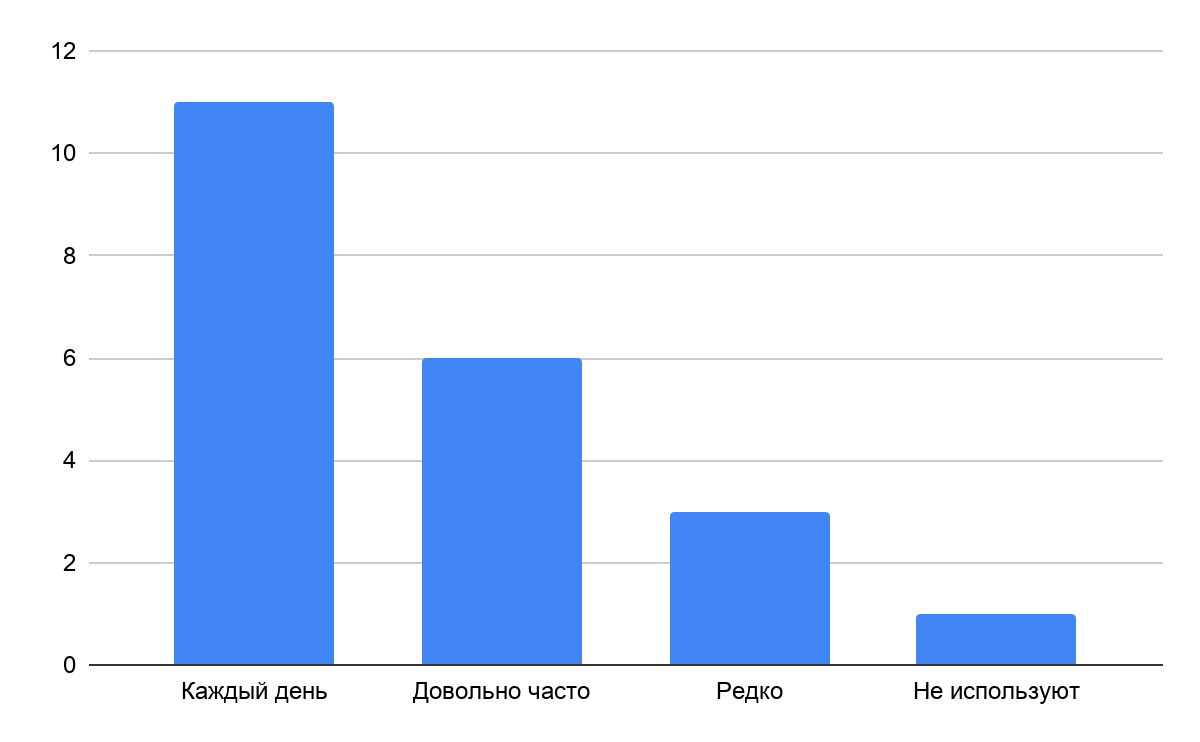


Рисунок 5; Диаграмма, показывающая частоту использования программы

Полностью не использует бота только один человек. В то время как каждый день программой пользуются больше половины учащихся.

# Заключение

Подводя итоги моей работы, можно сделать следующие выводы:

1. Программа полезна для преобладающего большинства опрошенных.
2. Программа удобна, так как имеет постоянный спрос.

# Список иллюстраций

Рисунок 1; Принцип работы программы[.](about:blank) 2

Рисунок 2; График, отображающий количество отправленных/полученных сообщений в течении дня за месяц12

Рисунок 3; График, отображающий количество пользователей, написавших сообщение в течении дня за месяц12

Рисунок 4; Диаграмма, отображающая популярность каждого из запросов.13

Рисунок 5; Диаграмма, показывающая частоту использования программы13

# Список использованных источников

1. Описание методовAPI <https://vk.com/dev/methods>.
2. API для чат-ботов https://vk.com/dev/bots\_docs.
3. Знакомство с API ВКонтакте https://vk.com/dev/first\_guide
4. Подробная информация о Python 3 <https://pythonworld.ru/samouchitel-python>
5. Подробная информация о применении чат-ботов https://www.carrotquest.io/blog/chatbot-types/

# Приложение(исходный код)

import vk\_api

import time

import random

print("Bot Vkontakte v2.0 start working... ")

token = ""

subjects = {"общество":0,"физика":1,"алгебра":2,"геометрия":3,"литра":4,"русский":5,"география":6,"биология":7,"англ1":8,"англ2":9,"информатика":10,"история":11,"химия":12,"обж":13,"литра(род)":14}

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

vk.\_auth\_token()

days=['пн','вт','ср','чт','пт','сб']

#days=[['пн','пон','понедельник'],['вт','вторник'],['ср','среда'],['чт','четверг'],['пт','пят','пятница'],['сб','суббота','субота']]

instruction=" Инструкция. Основные функции:\n 1.ДЗ на день недели (первые 2 согласные дня), примеры запроса: пн ; вт.\n 2.Дз по конкретному предмету, пример: дз алгебра. \n 3.Запись дз, пример: общество стр10 №1 (можно прикреплять фотографии)\n 4.Прошлое дз по предмету: пр алгебра .\n Предметы : общество, физика, алгебра, геометрия, литра, русский, география, биология, англ1, англ2, информатика, история, химия, обж, литра(род)."

zapr= "Запрос не опознан. Для получения инструкции введите \"инструкция\" (писать без ковычек)"

request\_error=0

spec = "Специальные возможности. Получение домашки на определенный день недели, примеры запросов: пн ; вт"

day\_lesson = {"пн": ["алгебра", "литра", "физика", "обж"], "вт": ["геометрия","англ1","англ2","общество","физика"],"ср": ["геометрия","англ1","англ2","русский","информатика","литра"],"чт": ["информатика","история","литра(род)"],"пт": ["англ1","англ2","алгебра","биология","география"],"сб": ["физика","алгебра","химия"]}

def files\_send(subject\_name,files\_name):

print("Из файла достаем id")

hometask\_file = open(files\_name, "r",encoding="utf-8")

hometask\_lines = hometask\_file.readlines()

hometask = hometask\_lines[subjects[subject\_name]]

hometask\_file.close()

return hometask

def file\_change(subject\_name, hmtsk,files\_name):

print("Запись в файл")

hometask\_file = open(files\_name, "r", encoding="utf-8")

hometask\_lines = hometask\_file.readlines()

hometask\_file.close()

hometask\_file = open(files\_name, "w", encoding="utf-8")

old\_hmtsk = hometask\_lines[subjects[subject\_name]]

hometask\_lines[subjects[subject\_name]] = hmtsk + " \n"

for i in range(len(hometask\_lines)):

hometask\_file.write(hometask\_lines[i])

hometask\_file.close()

return old\_hmtsk.strip()

def day\_send(day):# функция домашки по дням

print("Отправка домашки по дням")

message = ""

for i in range(len(day\_lesson[day])):

message=files\_send(day\_lesson[day][i],'hometask.txt')

vk.method("messages.send",{"peer\_id": id, "message": "", "random\_id": random.randint(1, 2147483647),"forward\_messages":message})

def join1(p):

k = " "

for i in range(len(p)):

k +=p[i]

k+= ' '

return k

def message\_in(msg\_txt):

request\_error = 0

if msg\_txt[0] in subjects:

file\_change(msg\_txt[0],file\_change(msg\_txt[0],str(id\_message),'hometask.txt'),'hmtsk\_last.txt')

vk.method("messages.send",{"peer\_id": id, "message": "дз получено", "random\_id": random.randint(1, 2147483647)})

request\_error= 1

if len(msg\_txt)>1:

print('3')

if msg\_txt[0]=="пр" and msg\_txt[1] in subjects:

request\_error = 1

vk.method("messages.send",{"peer\_id": id, "message": "Распишитесь", "random\_id": random.randint(1, 2147483647),"forward\_messages":files\_send(msg\_txt[1],'hmtsk\_last.txt')})

if msg\_txt[0]=="дз" and msg\_txt[1] in subjects:

vk.method("messages.send",{"peer\_id": id, "message": "Распишитесь", "random\_id": random.randint(1, 2147483647),"forward\_messages":files\_send(msg\_txt[1],'hometask.txt')})

request\_error = 1

print("Дз отправлено")

if msg\_txt[0] in days:

request\_error = 1

day\_send(msg\_txt[0])

print(msg\_txt[0])

return request\_error

while True:

try:

request\_error = 0

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

if messages["count"] >= 1:

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

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

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

k=body.lower().split()

request\_error = message\_in(body.lower().split())

if body.lower() == "инструкция":

vk.method("messages.send", {"peer\_id": id, "message": instruction, "random\_id": random.randint(1, 2147483647)})

request\_error=1

if request\_error == 0:

vk.method("messages.send", {"peer\_id": id, "message": zapr, "random\_id": random.randint(1, 2147483647)})

except Exception as E:

time.sleep(1)