Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №3

по дисциплине «Создание чат-ботов для социальной сети Telegram»

Выполнили:

студенты группы 22ВВП1

Демин М. С.

Беляев Д. И.

Захаров А. С.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2025

**Название**

Создание чат-ботов для социальной сети Telegram

**Цель работы**

Изучить библиотеки, позволяющие создавать чат-боты в Телеграм. Изучить Telegram API.

**Задание**

Протестировать чаты-боты данные из-задания.

**Ход работы**

Создание телеграм бота «Квест».

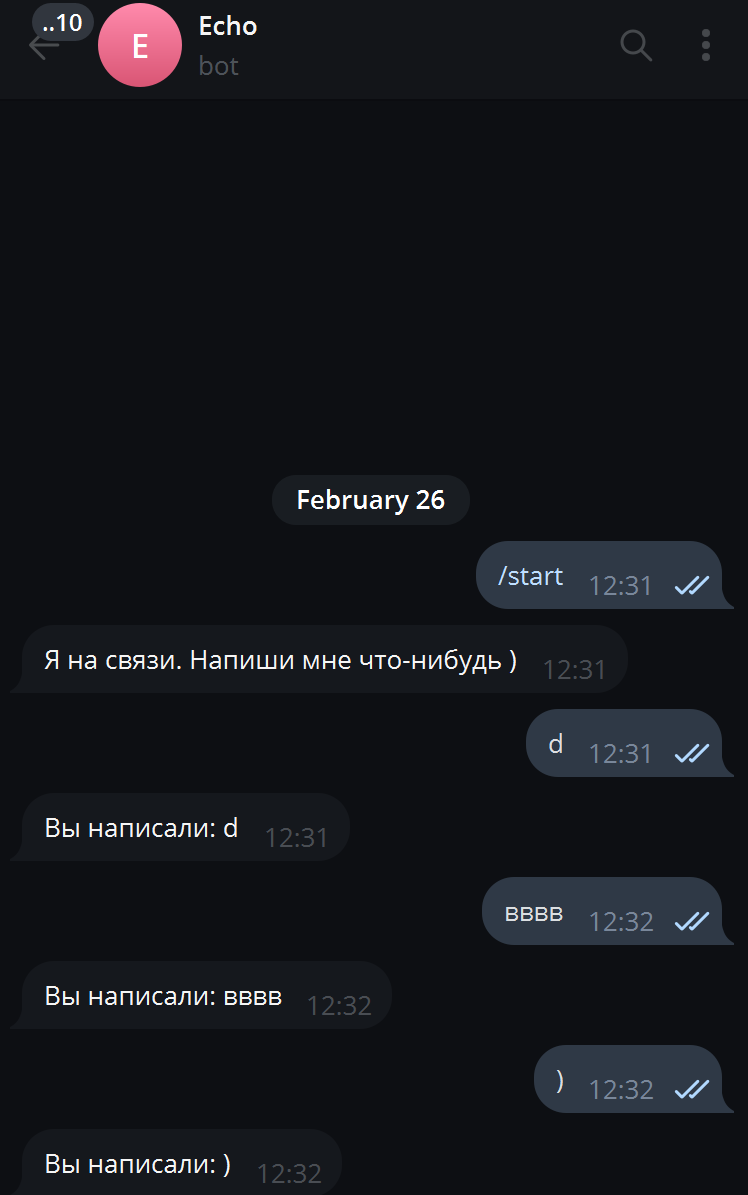
Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Листинг «Квест»

from telebot import TeleBot, types  
from random import randint  
from dotenv import load\_dotenv  
import os  
  
pictures = {  
 0: "https://cdn.ryfma.com/images/ozJtmxqwo9PBpAMWC/posts/post\_full\_1601737968052\_podzemele\_1-jpg.jpg",  
 1: "https://i.pinimg.com/736x/99/fa/fe/99fafead23c7da4b258e28a0c4e4baef.jpg",  
 2: "https://avatars.dzeninfra.ru/get-zen\_doc/1852570/pub\_5d9e314b43863f00b00484ec\_5d9e33f278125e00add2095c/scale\_1200",  
 3: "https://avatars.mds.yandex.net/i?id=22eea756c612e810255a8b6cedcce920049ad407-5246350-images-thumbs&n=13",  
 4: "https://c.wallhere.com/photos/57/44/stairs\_door\_window\_lights\_digital\_art\_digital\_lighting-92330.jpg!d",  
 5: "https://thumbs.dreamstime.com/b/вопрос-о-вери-38320736.jpg",  
 6: "https://avatars.mds.yandex.net/i?id=1f01ec801ed4d46d42631eae977cb859\_l-5910394-images-thumbs&n=13",  
 7: "https://avatars.mds.yandex.net/i?id=fc2e64c75fcc03556739197998f5b8d1\_l-5236988-images-thumbs&n=13",  
 8: "https://avatars.mds.yandex.net/i?id=fc2e64c75fcc03556739197998f5b8d1\_l-5236988-images-thumbs&n=13"  
}  
  
states = {}  
inventories = {}  
awaiting\_answers = {}  
  
dotenv\_path = os.path.join(os.path.dirname(\_\_file\_\_), '.env')  
if os.path.exists(dotenv\_path):  
 load\_dotenv(dotenv\_path)  
  
# load\_dotenv()  
# BOT\_TOKEN = os.getenv('BOT\_TOKEN')  
BOT\_TOKEN = ‘token’   
bot = TeleBot(BOT\_TOKEN)  
  
  
@bot.message\_handler(commands=["start"])  
def start\_game(message):  
 user = message.chat.id  
 states[user] = 0  
 inventories[user] = []  
 bot.send\_message(user, "Добро пожаловать в игру!")  
 process\_state(user, states[user], inventories[user])  
  
  
@bot.callback\_query\_handler(func=lambda call: True)  
def user\_answer(call):  
 user = call.message.chat.id  
 process\_answer(user, call.data)  
  
  
def process\_state(user, state, inventory):  
 kb = types.InlineKeyboardMarkup()  
 bot.send\_photo(user, pictures[state])  
  
 if state == 0:  
 kb.add(types.InlineKeyboardButton(text="пойти направо", callback\_data="1"))  
 kb.add(types.InlineKeyboardButton(text="пойти налево", callback\_data="2"))  
 bot.send\_message(user, "Вы оказались в темном подземелье, перед вами два прохода.", reply\_markup=kb)  
 elif state == 1:  
 kb.add(types.InlineKeyboardButton(text="осмотреть сундук", callback\_data="3"))  
 kb.add(types.InlineKeyboardButton(text="идти дальше", callback\_data="4"))  
 bot.send\_message(user, "Вы нашли старый сундук. Открыть или идти дальше?", reply\_markup=kb)  
 elif state == 2:  
 kb.add(types.InlineKeyboardButton(text="идти вперед", callback\_data="5"))  
 bot.send\_message(user, "Вы прошли через дверь и оказались в большой комнате.", reply\_markup=kb)  
 elif state == 3:  
 if randint(1, 10) > 3:  
 inventories[user].append("ключ")  
 bot.send\_message(user, "Вы нашли ключ в сундуке!")  
 else:  
 bot.send\_message(user, "Сундук оказался пустым.")  
 states[user] = 4  
 process\_state(user, states[user], inventories[user])  
 elif state == 4:  
 kb.add(types.InlineKeyboardButton(text="спуститься вниз", callback\_data="6"))  
 kb.add(types.InlineKeyboardButton(text="взять факел", callback\_data="8"))  
 bot.send\_message(user, "Вы попали в коридор с лестницей вниз. Здесь лежит факел.", reply\_markup=kb)  
 elif state == 5:  
 kb.add(types.InlineKeyboardButton(text="1 кг железа", callback\_data="wrong"))  
 kb.add(types.InlineKeyboardButton(text="1 кг пуха", callback\_data="wrong"))  
 kb.add(types.InlineKeyboardButton(text="Они весят одинаково", callback\_data="correct"))  
 bot.send\_message(user,  
 "Перед вами массивная дверь. Чтобы пройти, вам нужно решить загадку: Что тяжелее: 1 кг железа или 1 кг пуха?",  
 reply\_markup=kb)  
 elif state == 6:  
 if "факел" in inventory:  
 bot.send\_message(user, "Вы осветили путь факелом и прошли дальше.")  
 states[user] = 7  
 else:  
 bot.send\_message(user, "Здесь слишком темно, без факела идти опасно.")  
 elif state == 7:  
 if "ключ" in inventory:  
 bot.send\_message(user, "Вы открыли дверь ключом и выбрались наружу. Победа!")  
 states[user] = 8  
 else:  
 bot.send\_message(user, "Дверь заперта, но вы можете попробовать выбить её (шанс 50%).")  
 if randint(1, 10) > 5:  
 bot.send\_message(user, "Вы с силой толкнули дверь, и она поддалась! Вы выбрались наружу. Победа!")  
 states[user] = 8  
 else:  
 bot.send\_message(user, "Вы не смогли выбить дверь. Попробуйте ещё раз.")  
 elif state == 8:  
 kb.add(types.InlineKeyboardButton(text="Начать заново", callback\_data="restart"))  
 bot.send\_message(user, "Поздравляем, вы прошли игру! Хотите сыграть снова?", reply\_markup=kb)  
  
  
def process\_answer(user, answer):  
 if answer == "restart":  
 start\_game(types.Message(chat=types.Chat(id=user)))  
 return  
  
 if states[user] == 0:  
 if answer == "1":  
 states[user] = 1  
 else:  
 states[user] = 2  
 elif states[user] == 1:  
 if answer == "3":  
 states[user] = 3  
 else:  
 states[user] = 4  
 elif states[user] == 2:  
 states[user] = 5  
 elif states[user] == 4:  
 if answer == "6":  
 states[user] = 6  
 elif answer == "8":  
 states[user] = 8  
 elif states[user] == 5:  
 if answer == "correct":  
 bot.send\_message(user, "Верно! Дверь открывается.")  
 states[user] = 7  
 else:  
 bot.send\_message(user, "Неправильно. Попробуйте еще раз.")  
 process\_state(user, states[user], inventories[user])  
  
  
bot.polling(none\_stop=True)

Создание телеграм бота «Эхо»



Листинг «Эхо»

import telebot  
# Создаем экземпляр бота  
bot = telebot.TeleBot(‘token’)  
# Функция, обрабатывающая команду /start  
@bot.message\_handler(commands=["start"])  
def start(m, res=False):  
 bot.send\_message(m.chat.id, 'Я на связи. Напиши мне что-нибудь )')  
# Получение сообщений от юзера  
@bot.message\_handler(content\_types=["text"])  
def handle\_text(message):  
 bot.send\_message(message.chat.id, 'Вы написали: ' + message.text)  
# Запускаем бота  
bot.polling(none\_stop=True, interval=0)

Создание телеграм бота «WIKIPEDIA-БОТ».

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Листинг «Вики»

import telebot, wikipedia, re  
# Создаем экземпляр бота  
bot = telebot.TeleBot(‘token')  
# Устанавливаем русский язык в Wikipedia  
wikipedia.set\_lang("ru")  
# Чистим текст статьи в Wikipedia и ограничиваем его тысячей символов  
def getwiki(s):  
 try:  
 ny = wikipedia.page(s)  
 # Получаем первую тысячу символов  
 wikitext=ny.content[:1000]  
 # Разделяем по точкам  
 wikimas=wikitext.split('.')  
 # Отбрасываем всЕ после последней точки  
 wikimas = wikimas[:-1]  
 # Создаем пустую переменную для текста  
 wikitext2 = ''  
 # Проходимся по строкам, где нет знаков «равно» (то есть все, кроме заголовков)  
 for x in wikimas:  
 if not('==' in x):  
 # Если в строке осталось больше трех символов, добавляем ее к нашей переменной и возвращаем утерянные при разделении строк точки на место  
 if(len((x.strip()))>3):  
 wikitext2=wikitext2+x+'.'  
 else:  
 break  
 # Теперь при помощи регулярных выражений убираем разметку  
 wikitext2=re.sub('\([^()]\*\)', '', wikitext2)  
 wikitext2=re.sub('\([^()]\*\)', '', wikitext2)  
 wikitext2=re.sub('\{[^\{\}]\*\}', '', wikitext2)  
 # Возвращаем текстовую строку  
 return wikitext2  
 # Обрабатываем исключение, которое мог вернуть модуль wikipedia при запросе  
 except Exception as e:  
 return 'В энциклопедии нет информации об этом'  
# Функция, обрабатывающая команду /start  
@bot.message\_handler(commands=["start"])  
def start(m, res=False):  
 bot.send\_message(m.chat.id, 'Отправьте мне любое слово, и я найду его значение на Wikipedia')  
# Получение сообщений от юзера  
@bot.message\_handler(content\_types=["text"])  
def handle\_text(message):  
 bot.send\_message(message.chat.id, getwiki(message.text))  
# Запускаем бота  
bot.polling(none\_stop=True, interval=0)

Создание телеграм бота «Две кнопки».

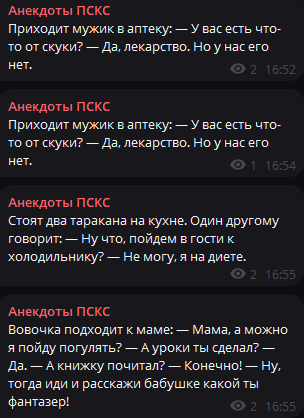
Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Листинг бота «Две конопки»

import telebot  
import random  
from telebot import types  
# Загружаем список интересных фактов  
f = open('data/facts.txt', 'r', encoding='UTF-8')  
facts = f.read().split('\n')  
f.close()  
# Загружаем список поговорок  
f = open('data/thinks.txt', 'r', encoding='UTF-8')  
thinks = f.read().split('\n')  
f.close()  
# Создаем бота  
bot = telebot.TeleBot('token')  
# Команда start  
@bot.message\_handler(commands=["start"])  
def start(m, res=False):  
 # Добавляем две кнопки  
 markup=types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1=types.KeyboardButton("Факт")  
 item2=types.KeyboardButton("Поговорка")  
 markup.add(item1)  
 markup.add(item2)  
 bot.send\_message(m.chat.id, 'Нажмите: \nФакт для получения интересного факта\nПоговорка — для получения мудрой цитаты ', reply\_markup=markup)  
# Получение сообщений от юзера  
@bot.message\_handler(content\_types=["text"])  
def handle\_text(message):  
 # Если юзер прислал 1, выдаем ему случайный факт  
 if message.text.strip() == 'Факт' :  
 answer = random.choice(facts)  
 # Если юзер прислал 2, выдаем умную мысль  
 elif message.text.strip() == 'Поговорка':  
 answer = random.choice(thinks)  
 # Отсылаем юзеру сообщение в его чат  
 bot.send\_message(message.chat.id, answer)  
# Запускаем бота  
bot.polling(none\_stop=True, interval=0)

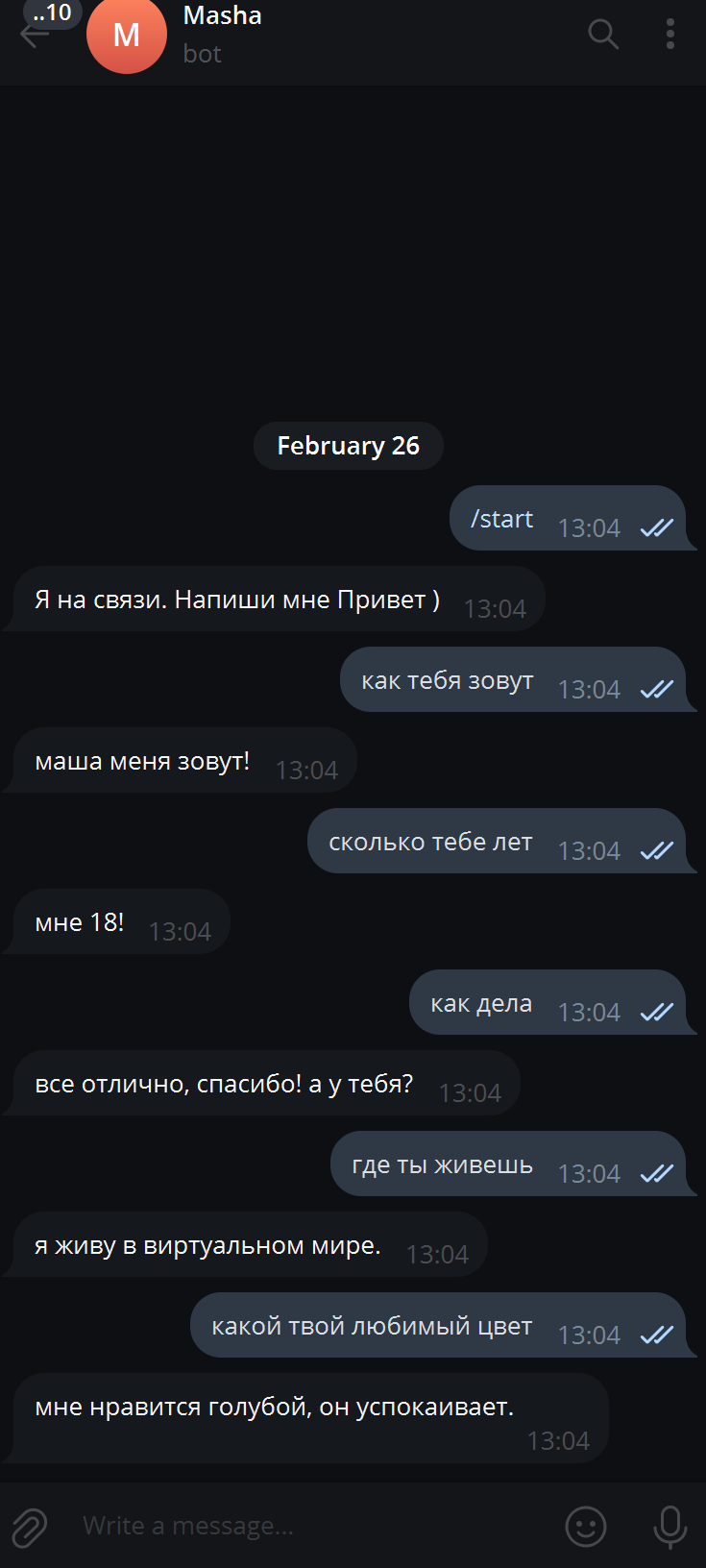
Создание телеграм бота «Анекдот».

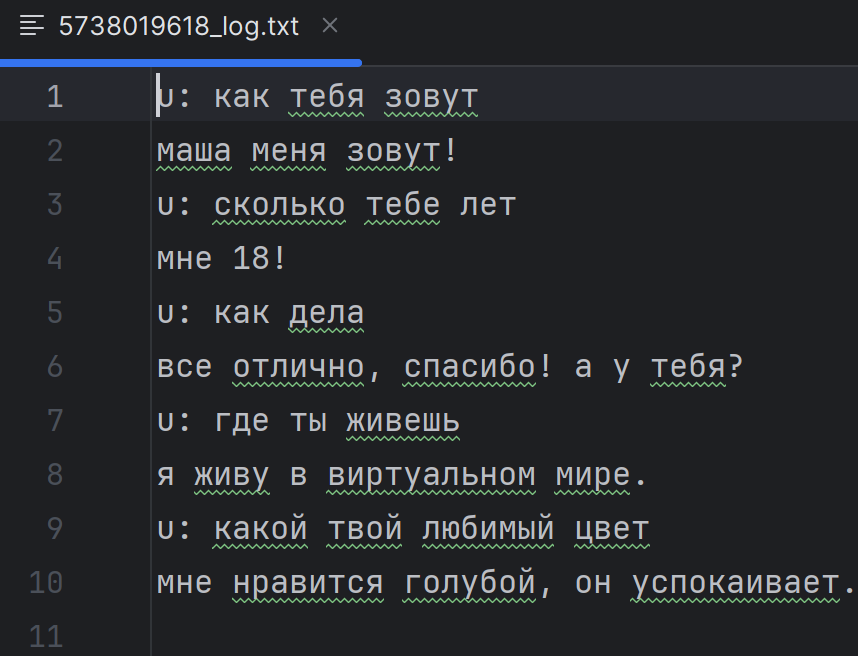


Листинг «Анекдота»

import telebot  
import time  
  
# Токен, который выдает @botfather  
bot = telebot.TeleBot(‘’)  
# Адрес телеграм-канала, начинается с @  
CHANNEL\_NAME = '@psks\_anekdot'  
# Загружаем список шуток  
f = open('data/fun.txt', 'r', encoding='UTF-8')  
jokes = f.read().split('\n')  
f.close()  
# Пока не закончатся шутки, посылаем их в канал  
for joke in jokes:  
 try:  
 bot.send\_message(CHANNEL\_NAME, joke)  
 # Делаем паузу в один час  
 time.sleep(10)  
 except telebot.apihelper.ApiTelegramException as e:  
 print(f"Не удалось отправить сообщение в канал {CHANNEL\_NAME}: {e}")  
bot.send\_message(CHANNEL\_NAME, "Анекдоты закончились :-(")

Создание телеграм бота «Маша».

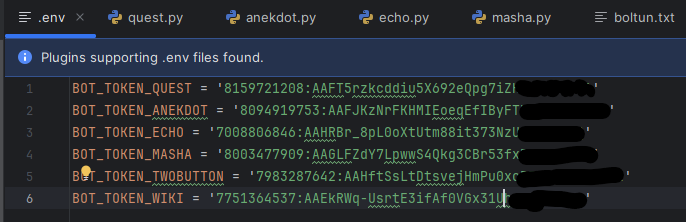




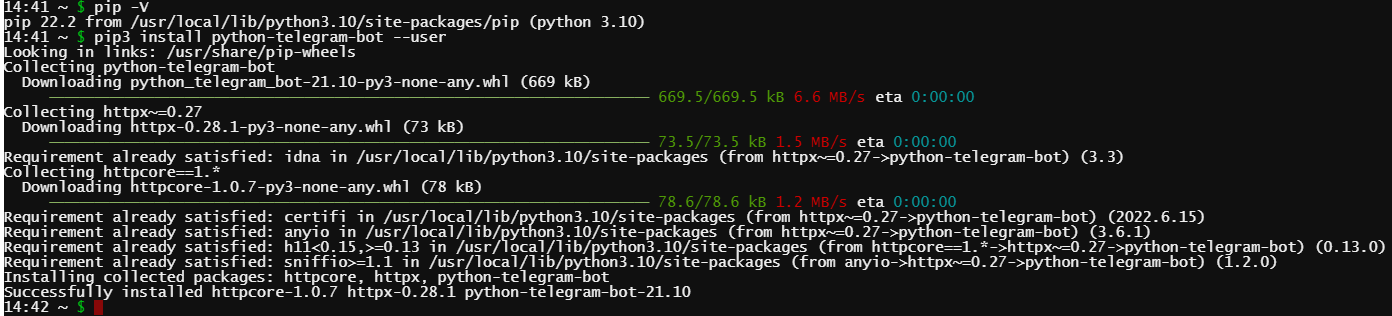
Листинг «Маша»

import telebot  
import os  
from fuzzywuzzy import fuzz  
# Создаем бота, пишем свой токен  
bot = telebot.TeleBot(‘token’)  
# Загружаем список фраз и ответов в массив  
mas=[]  
if os.path.exists('data/boltun.txt'):  
 f=open('data/boltun.txt', 'r', encoding='UTF-8')  
 for x in f:  
 if(len(x.strip()) > 2):  
 mas.append(x.strip().lower())  
 f.close()  
# С помощью fuzzywuzzy вычисляем наиболее похожую фразу и выдаем в качестве ответа следующий элемент списка  
def answer(text):  
 try:  
 text=text.lower().strip()  
 if os.path.exists('data/boltun.txt'):  
 a = 0  
 n = 0  
 nn = 0  
 for q in mas:  
 if('u: ' in q):  
 # С помощью fuzzywuzzy получаем, насколько похожи две строки  
 aa=(fuzz.token\_sort\_ratio(q.replace('u: ',''), text))  
 if(aa > a and aa!= a):  
 a = aa  
 nn = n  
 n = n + 1  
 s = mas[nn + 1]  
 return s  
 else:  
 return 'Ошибка'  
 except:  
 return 'Ошибка'  
# Команда «Старт»  
@bot.message\_handler(commands=["start"])  
def start(m, res=False):  
 bot.send\_message(m.chat.id, 'Я на связи. Напиши мне Привет )')  
# Получение сообщений от юзера  
@bot.message\_handler(content\_types=["text"])  
def handle\_text(message):  
 # Запись логов  
 f=open('data/' + str(message.chat.id) + '\_log.txt', 'a', encoding='UTF-8')  
 s=answer(message.text)  
 f.write('u: ' + message.text + '\n' + s +'\n')  
 f.close()  
 # Отправка ответа  
 bot.send\_message(message.chat.id, s)  
# Запускаем бота  
bot.polling(none\_stop=True, interval=0)

Создали виртуальное окружение .env для хранения токенов.



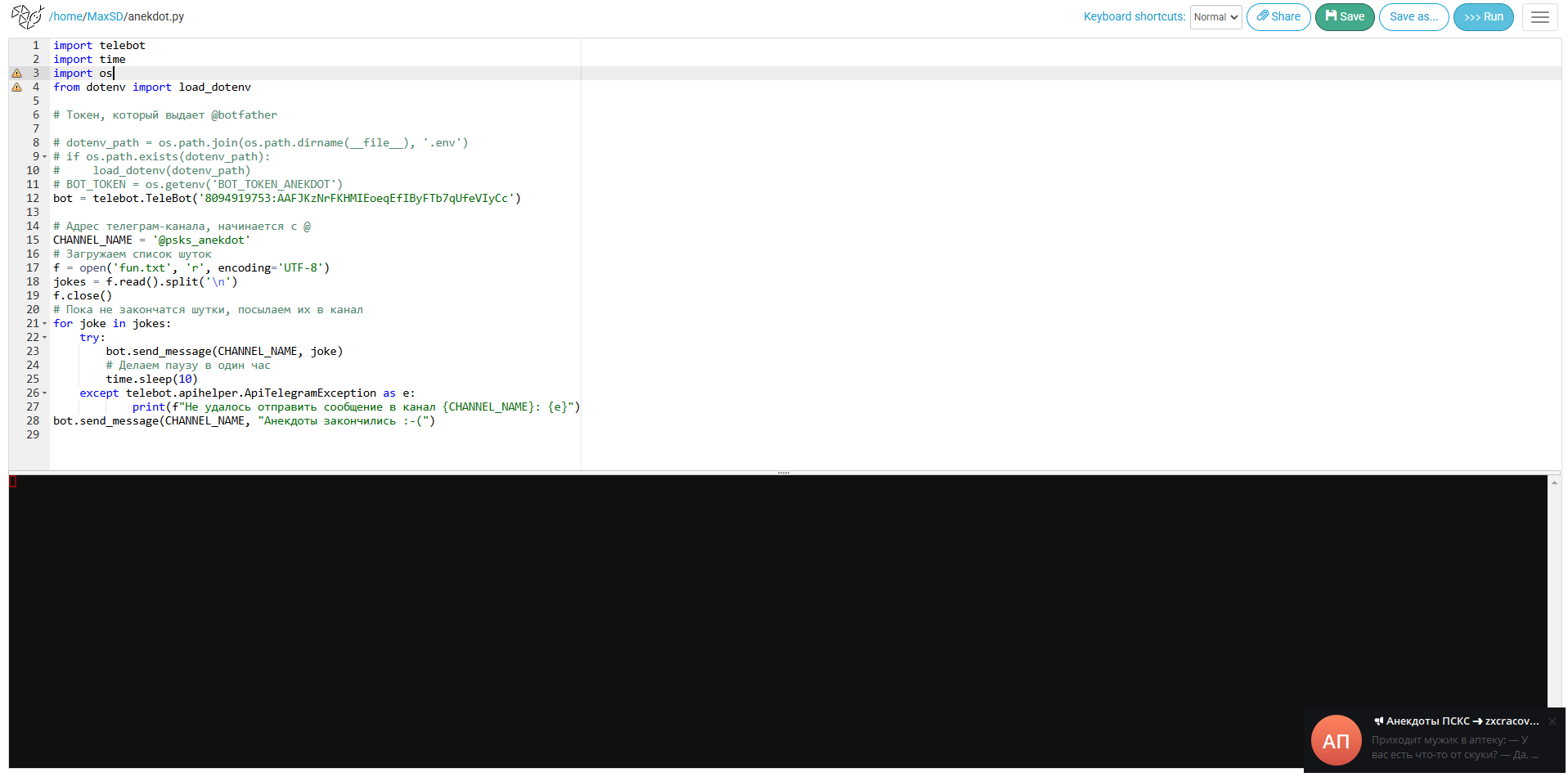
Настраиваем бесплатный сервер PythonAnywhere.



Загрузили файл для проверки



Проверка работоспособности, анекдоты пишутся.



**Вывод**

Мы научились создавать чат-ботов в телеграм, поняли принцип работы Telegram API. Изучили бесплатный сервис для размещения кода.