

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по ЛР№6

Выполнил:

студент группы ИУ5-34Б

Ступницкий Михаил

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023 г.

Задание:

Реализовать бота-администратора для Telegram

Текст программы:

```
import telebot
from telebot import types
import random
import time

bot = telebot.TeleBot('6485200058:AAG3pSh3YAEiafb5BGp-k8VbdOkXkdn8wps')

name = ''
surname = ''
age = 0
stats = {}

@bot.message_handler(commands=['start'])
def start(message):
    bot.reply_to(message, "Привет! Я бот для управления чатом. Напиши /menu, чтобы увидеть все функции.")

@bot.message_handler(commands=['menu'])
def menu(message):
    bot.reply_to(message, 'Вот что я умею:\n/reg - Знакомство\n/menu - вызов меню\n/clowntest - тест на клоуна\n\n/kick - кикнуть пользователя\n/mute - замутить пользователя на определенное время\n/unmute - размутить пользователя')

@bot.message_handler(commands=['Привет'])
def hi(message):
    bot.reply_to(message, "Привет, чем я могу тебе помочь?")

@bot.message_handler(commands=['reg'])
def reg(message):
    bot.reply_to(message, "Как тебя зовут?")
    bot.register_next_step_handler(message, get_name)

@bot.message_handler(commands=['clowntest'])
def clowntest(message):
    bot.reply_to(message, "Как тебя зовут?")
    bot.register_next_step_handler(message, get_clown)

@bot.message_handler(commands=['result'])
def result(message):
    a = random.randint(0, 100)
    answer = 'Ты на ' + str(a) + '% клоун'
    bot.send_message(message.from_user.id, answer,
reply_markup=types.ReplyKeyboardRemove())

def get_name(message):
    global name
    name = message.text
    bot.send_message(message.from_user.id, 'Какая у тебя фамилия?')
    bot.register_next_step_handler(message, get_surname)
```

```

def get_surname(message):
    global surname
    surname = message.text
    bot.send_message(message.from_user.id, 'Сколько тебе лет?')
    bot.register_next_step_handler(message, get_age)

def get_age(message):
    global age;
    while age == 0: # проверяем что возраст изменился
        try:
            age = int(message.text) # проверяем, что возраст введен
корректно
        except Exception:
            bot.send_message(message.from_user.id, 'Цифрами, пожалуйста');
            keyboard = types.InlineKeyboardMarkup(); # наша клавиатура
            key_yes = types.InlineKeyboardButton(text='Да', callback_data='yes'); #
кнопка «Да»
            keyboard.add(key_yes); # добавляем кнопку в клавиатуру
            key_no = types.InlineKeyboardButton(text='Нет', callback_data='no');
            keyboard.add(key_no);
            question = 'Тебе ' + str(age) + ' лет, тебя зовут ' + name + ' ' +
surname + '?';
            bot.send_message(message.from_user.id, text=question,
reply_markup=keyboard)
            age = 0

def get_clown(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard = True)
    btn1 = types.KeyboardButton("Узнать результат")
    markup.add(btn1)
    bot.send_message(message.from_user.id, "Результат готов",
reply_markup=markup)

@bot.message_handler(commands=['kick'])
def kick_user(message):
    if message.reply_to_message:
        chat_id = message.chat.id
        user_id = message.reply_to_message.from_user.id
        user_status = bot.get_chat_member(chat_id, user_id).status
        if user_status == 'administrator' or user_status == 'creator':
            bot.reply_to(message, "Невозможно кикнуть администратора.")
        else:
            bot.kick_chat_member(chat_id, user_id)
            bot.reply_to(message, f"Пользователь
{message.reply_to_message.from_user.username} был кикнут.")
    else:
        bot.reply_to(message, "Эта команда должна быть использована в ответ
на сообщение пользователя, которого вы хотите кикнуть.")

@bot.message_handler(commands=['mute'])
def mute_user(message):
    if message.reply_to_message:
        chat_id = message.chat.id
        user_id = message.reply_to_message.from_user.id
        user_status = bot.get_chat_member(chat_id, user_id).status
        if user_status == 'administrator' or user_status == 'creator':
            bot.reply_to(message, "Невозможно замутить администратора.")
        else:
            duration = 60 # Значение по умолчанию - 1 минута
            args = message.text.split()[1:]
            if args:

```

```

        try:
            duration = int(args[0])
        except ValueError:
            bot.reply_to(message, "Неправильный формат времени.")
            return
        if duration < 1:
            bot.reply_to(message, "Время должно быть положительным
числом.")
            return
        if duration > 1440:
            bot.reply_to(message, "Максимальное время - 1 день.")
            return
        bot.restrict_chat_member(chat_id, user_id,
until_date=time.time()+duration*60)
        bot.reply_to(message, f"Пользователь
{message.reply_to_message.from_user.username} замучен на {duration} минут.")
    else:
        bot.reply_to(message, "Эта команда должна быть использована в ответ
на сообщение пользователя, которого вы хотите замутить.")

@bot.message_handler(commands=['unmute'])
def unmute_user(message):
    if message.reply_to_message:
        chat_id = message.chat.id
        user_id = message.reply_to_message.from_user.id
        bot.restrict_chat_member(chat_id, user_id, can_send_messages=True,
can_send_media_messages=True, can_send_other_messages=True,
can_add_web_page_previews=True)
        bot.reply_to(message, f"Пользователь
{message.reply_to_message.from_user.username} размучен.")
    else:
        bot.reply_to(message, "Эта команда должна быть использована в ответ
на сообщение пользователя, которого вы хотите размутить.")

bad_words = ['клоун', 'негатив', 'физика']

def check_message(message):
    for word in bad_words:
        if word in message.text.lower():
            return True
    return False

@bot.message_handler(func=lambda message: True)
def handle_message(message):
    if check_message(message):
        chat_id = message.chat.id
        user_id = message.from_user.id
        user_status = bot.get_chat_member(chat_id, user_id).status
        if user_status == 'administrator' or user_status == 'creator':
            bot.reply_to(message, "Невозможно кикнуть администратора.
Предупреждение за использование запрещенных слов")
        else:
            bot.kick_chat_member(message.chat.id, message.from_user.id)
            bot.send_message(message.chat.id, f"Пользователь
{message.from_user.username} был удален из чата за использование запрещенных
слов")
    else:
        print(message.text)

@bot.callback_query_handler(func=lambda call: True)
def callback_worker(call):
    global foods

```

```

if call.data == "yes":
    bot.send_message(call.message.chat.id, 'Запомню : )');
elif call.data == "no":
    bot.send_message(call.message.chat.id, 'Чтобы пройти регистрацию
заново, напиши /reg')
    bot.edit_message_reply_markup(chat_id=call.message.chat.id,
message_id=call.message.message_id)

```

bot.polling(none_stop=True, interval=0)

Результат выполнения программы:

