|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  Отчет по выполнению лабораторной работы №6 по дисциплине “Парадигмы и конструкции языков программирования”  38.03.05 Бизнес-Информатика (уровень бакалавриата)  Студент ИБМ3-34 Шевченко Г.А.  2024 г. | |

Код программы:  
import telebot

from telebot.types import Message, ReplyKeyboardMarkup, KeyboardButton

# Здесь нужно поместить ваш токен от BotFather

TOKEN = '7513357305:AAFc8IYifmI-UJGh\_csV937rpjJRnD5lb5U'

bot = telebot.TeleBot(TOKEN)

# Определяем состояния

STATE\_LIGHT\_OFF = 'Свет выключен'

STATE\_LIGHT\_ON = 'Свет включен'

STATE\_ALARM\_OFF = 'Сигнализация выключена'

STATE\_ALARM\_ON = 'Сигнализация включена'

STATE\_TV\_OFF = 'ТВ-приставка выключена'

STATE\_TV\_ON = 'ТВ-приставка включена'

# Храним состояние каждого пользователя

user\_states = {}

# Функция для инициализации состояния

def initialize\_user(chat\_id):

    if chat\_id not in user\_states:

        user\_states[chat\_id] = {

            'light': STATE\_LIGHT\_OFF,

            'alarm': STATE\_ALARM\_OFF,

            'tv': STATE\_TV\_OFF

        }

# Функция для создания клавиатуры

def create\_keyboard():

    markup = ReplyKeyboardMarkup(resize\_keyboard=True)

    markup.row(KeyboardButton('Включить свет'), KeyboardButton('Выключить свет'))

    markup.row(KeyboardButton('Включить сигнализацию'), KeyboardButton('Выключить сигнализацию'))

    markup.row(KeyboardButton('Включить ТВ-приставку'), KeyboardButton('Выключить ТВ-приставку'))

    return markup

# Функция-обработчик команд /start

@bot.message\_handler(commands=['start'])

def start\_command(message: Message):

    chat\_id = message.chat.id

    initialize\_user(chat\_id)

    bot.send\_message(

        chat\_id,

        'Добро пожаловать! Управляйте светом, сигнализацией и ТВ-приставкой с помощью кнопок ниже.',

        reply\_markup=create\_keyboard()

    )

# Функция-обработчик текстовых сообщений

@bot.message\_handler(func=lambda m: True)

def handle\_message(message: Message):

    chat\_id = message.chat.id

    initialize\_user(chat\_id)

    user\_state = user\_states[chat\_id]

    if message.text.lower() == 'включить свет':

        if user\_state['light'] == STATE\_LIGHT\_ON:

            bot.send\_message(chat\_id, 'Ошибка: свет уже включен.', reply\_markup=create\_keyboard())

        else:

            user\_state['light'] = STATE\_LIGHT\_ON

            bot.send\_message(chat\_id, 'Свет включен.', reply\_markup=create\_keyboard())

    elif message.text.lower() == 'выключить свет':

        if user\_state['light'] == STATE\_LIGHT\_OFF:

            bot.send\_message(chat\_id, 'Ошибка: свет уже выключен.', reply\_markup=create\_keyboard())

        else:

            user\_state['light'] = STATE\_LIGHT\_OFF

            bot.send\_message(chat\_id, 'Свет выключен.', reply\_markup=create\_keyboard())

    elif message.text.lower() == 'включить сигнализацию':

        if user\_state['alarm'] == STATE\_ALARM\_ON:

            bot.send\_message(chat\_id, 'Ошибка: сигнализация уже включена.', reply\_markup=create\_keyboard())

        else:

            user\_state['alarm'] = STATE\_ALARM\_ON

            bot.send\_message(chat\_id, 'Сигнализация включена.', reply\_markup=create\_keyboard())

    elif message.text.lower() == 'выключить сигнализацию':

        if user\_state['alarm'] == STATE\_ALARM\_OFF:

            bot.send\_message(chat\_id, 'Ошибка: сигнализация уже выключена.', reply\_markup=create\_keyboard())

        else:

            user\_state['alarm'] = STATE\_ALARM\_OFF

            bot.send\_message(chat\_id, 'Сигнализация выключена.', reply\_markup=create\_keyboard())

    elif message.text.lower() == 'включить тв-приставку':

        if user\_state['tv'] == STATE\_TV\_ON:

            bot.send\_message(chat\_id, 'Ошибка: ТВ-приставка уже включена.', reply\_markup=create\_keyboard())

        else:

            user\_state['tv'] = STATE\_TV\_ON

            bot.send\_message(chat\_id, 'ТВ-приставка включена.', reply\_markup=create\_keyboard())

    elif message.text.lower() == 'выключить тв-приставку':

        if user\_state['tv'] == STATE\_TV\_OFF:

            bot.send\_message(chat\_id, 'Ошибка: ТВ-приставка уже выключена.', reply\_markup=create\_keyboard())

        else:

            user\_state['tv'] = STATE\_TV\_OFF

            bot.send\_message(chat\_id, 'ТВ-приставка выключена.', reply\_markup=create\_keyboard())

    else:

        bot.send\_message(chat\_id, 'Я вас не понял. Используйте доступные команды.', reply\_markup=create\_keyboard())

# Запуск бота

if \_\_name\_\_ == '\_\_main\_\_':

    print('Бот запущен...')

    bot.polling(none\_stop=True)

Результат выполнения программы:



