

Отчет по домашнему заданию

Telegram бот переводчик

Текст программы

```
import telebot

from deep_translator import GoogleTranslator, exceptions

BOT_TOKEN = '7610180894:AAGnFqUmMRF2lktYcvg6pCt6SEI6_SkXRCY'
bot = telebot.TeleBot(BOT_TOKEN)

# Доступные языки
languages = {
    'ru': 'Русский',
    'en': 'Английский',
    'es': 'Испанский',
    'de': 'Немецкий',
    'it': 'Итальянский',
    'fr': 'Французский',
}

# Языки по умолчанию
DEFAULT_SOURCE_LANGUAGE = 'ru'
DEFAULT_TARGET_LANGUAGE = 'en'

# Состояние пользователя
user_states = {}

def create_translator_keyboard():
```

```
keyboard = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
keyboard.add("Выбрать исходный язык")
keyboard.add("Выбрать язык перевода")
keyboard.add("Помощь")
keyboard.add("Стоп")
keyboard.add("Старт")
return keyboard
```

```
def create_language_keyboard():
    markup = telebot.types.ReplyKeyboardMarkup(resize_keyboard=True)
    for code, name in languages.items():
        markup.add(name)
    markup.add("Назад")
    return markup
```

```
@bot.message_handler(commands=['start'])
def start_command(message):
    chat_id = message.chat.id
    user_states[chat_id] = {
        'source_language': DEFAULT_SOURCE_LANGUAGE,
        'target_language': DEFAULT_TARGET_LANGUAGE,
        'state': 'main_menu'
    }
    bot.send_message(chat_id,
        f"Привет! Используются языки по умолчанию:
        {languages[DEFAULT_SOURCE_LANGUAGE]} ->
        {languages[DEFAULT_TARGET_LANGUAGE]}. Выберите действие:",
        reply_markup=create_translator_keyboard())
```

```

@bot.message_handler(func=lambda message: True)
def handle_message(message):
    chat_id = message.chat.id

    if chat_id not in user_states:
        start_command(message)
        return

    state = user_states[chat_id]['state']

    if state == 'main_menu':
        if message.text == "Выбрать исходный язык":
            bot.send_message(chat_id, "Выберите исходный язык:",
reply_markup=create_language_keyboard())

            user_states[chat_id]['state'] = 'choosing_source'

        elif message.text == "Выбрать язык перевода":
            bot.send_message(chat_id, "Выберите язык перевода:",
reply_markup=create_language_keyboard())

            user_states[chat_id]['state'] = 'choosing_target'

        elif message.text == "Помощь":
            bot.send_message(chat_id, "Привет, я бот-переводчик, выбери
исходный язык и язык перевода. Затем напиши слово/текст, перевод
которого тебе нужен", reply_markup=create_translator_keyboard())

        elif message.text == "Стоп":
            bot.send_message(chat_id, "Работа бота остановлена",
reply_markup=None)

            del user_states[chat_id]

    else:
        # Обработка текста, если не выбрано действие

```

```

        translate_and_send(message)

    elif state in ['choosing_source', 'choosing_target']:
        if message.text == "Назад":
            user_states[chat_id]['state'] = 'main_menu'
            bot.send_message(chat_id, "Возвращаемся в главное меню",
reply_markup=create_translator_keyboard())
        elif message.text in languages.values():
            lang_code =
list(languages.keys())[list(languages.values()).index(message.text)]
            if state == 'choosing_source':
                user_states[chat_id]['source_language'] = lang_code
            else:
                user_states[chat_id]['target_language'] = lang_code
            user_states[chat_id]['state'] = 'main_menu'
            bot.send_message(chat_id, f"Язык выбран! Теперь введите текст для
перевода.", reply_markup=create_translator_keyboard())
        else:
            bot.reply_to(message, "Пожалуйста, выберите язык из предложенных.")
    else:
        translate_and_send(message)

```

```

def translate_and_send(message):
    chat_id = message.chat.id
    source_language = user_states[chat_id]['source_language']
    target_language = user_states[chat_id]['target_language']
    try:
        translator = GoogleTranslator(source=source_language,
target=target_language)

```

```
translated_text = translator.translate(message.text)
bot.reply_to(message, translated_text)
except Exception as e:
    bot.reply_to(message, f"Ошибка перевода: {e}")
```

```
bot.polling()
```

Примеры выполнения программы



Привет! Используются языки по умолчанию: Русский -> Английский. Выберите действие:

00:25

Выбрать исходный язык 00:25 ✓✓

Выберите исходный язык: 00:25

Русский 00:25 ✓✓

Язык выбран! Теперь введите текст для перевода. 00:25

Выбрать язык перевода 00:25 ✓✓

Выберите язык перевода: 00:25

Немецкий 00:25 ✓✓

Язык выбран! Теперь введите текст для перевода. 00:25

Кошка села на асфальт 00:25 ✓✓

Карина

Кошка села на асфальт

Die Katze setzte sich auf den Asphalt 00:25

Помощь 00:25 ✓✓



Написать сообщение...



Выбрать исходный язык

Выбрать язык перевода

Помощь

Стоп

