

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

Отчёт по Домашнему заданию
по курсу «Парадигмы и конструкции языков программирования»

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Выполнил:
Студент группы РТ5-31Б
Иванов А. А.

Москва, 2024 г.

В качестве ДЗ был реализован телеграмм бот, способный переводить текст с русского языка на иностранный (в боте присутствует выбор 5 иностранных языков).

Код программы:

```
"ДЗ по курсу ПиКЯП 3 сем"
```

```
import telebot #библиотека для работы с Telegram Bot API
```

```
from telebot.types import ReplyKeyboardMarkup, KeyboardButton,  
InlineKeyboardMarkup, InlineKeyboardButton
```

```
from googletrans import Translator # Библиотека перевода
```

```
import logging # Библиотека для логов
```

```
# Создаем объект переводчика
```

```
translator = Translator()
```

```
TOKEN = "8099116151:AAGpGoBf2BSo_FrChimBnnb2laD2-Vt7Rjs" # токен  
tg-бота
```

```
bot = telebot.TeleBot(TOKEN) # Инициализация объекта бота
```

```
# Настройка логов: выводит: время - [код ошибки] название ошибки
```

```
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(message)s')
```

```
logger = logging.getLogger() # Создаем объект для записи логов
```

```
# Языки для перевода
```

```
LANGUAGES = {
```

```
    "en": "English",
```

```
    "es": "Spanish",
```

```
"fr": "French",  
"de": "German",  
"it": "Italian"  
}
```

Словари для статистики пользователей:

```
user_language = {} # Хранит выбранный язык перевода для каждого пользователя
```

```
user_stats = {} # Хранит статистику переводов по языкам для каждого пользователя
```

```
def lang_keyboard():
```

```
    """Создает клавиатуру с кнопками для выбора языка"""
```

```
    keyboard = InlineKeyboardMarkup() # клавиатура с кнопками
```

```
    for lang_code, lang_name in LANGUAGES.items():
```

```
        # Добавляем кнопку с названием языка
```

```
        keyboard.add(InlineKeyboardButton(lang_name,  
            callback_data=f"lang_{lang_code}"))
```

```
    return keyboard
```


```
def main_menu():
```

```
    """Создает главное меню с основными кнопками"""
```

```
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True,  
        one_time_keyboard=True)
```

```
    # Добавляем кнопки
```

```
    keyboard.add(KeyboardButton("Статистика 
```

```
keyboard.add(KeyboardButton("Помощь 
```

```
def trans_menu():

    """Создает меню с кнопкой завершения перевода"""

    keyboard = ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)

    # Добавляем кнопку для выхода из режима перевода

    keyboard.add(KeyboardButton("Завершить перевод"))

    return keyboard
```

```
def update_stats(user_id, lang_code):

    """Обновляет статистику переводов для конкретного пользователя"""

    if user_id not in user_stats:

        # если пользователь отсутствует в статистике, создаем запись с
        нулями

        user_stats[user_id] = {lang: 0 for lang in LANGUAGES.keys()}

        # Увеличиваем количество переводов для выбранного языка

        user_stats[user_id][lang_code] += 1
```

```
def get_stats(user_id):

    """Возвращает статистику переводов для пользователя"""

    if user_id not in user_stats:

        # если статистики нет, возвращаем сообщение
```

```

    return "Нет данных о вашей активности"

    # форматируем статистику

    stats = user_stats[user_id]

    return "\n".join([f'{LANGUAGES[lang]}: {count}' for lang, count in
stats.items() if count > 0])

@bot.message_handler(commands=["start"])

def start(message):

    """Обрабатывает команду /start"""

    bot.send_message(

        message.chat.id,

        "Привет! Я могу переводить текст с русского на другие языки. Выберите
язык для перевода:",

        reply_markup=lang_keyboard() # клавиатура выбора языка

    )

    bot.send_message(

        message.chat.id,

        "Также доступно меню:",

        reply_markup=main_menu() # клавиатура

    )


@bot.callback_query_handler(func=lambda call: call.data.startswith("lang_"))

def select_lang(call):

    """Обрабатывает выбор языка перевода"""

```

```
lang_code = call.data.split("_")[1] # извлекает код языка из данных кнопки  
user_language[call.message.chat.id] = lang_code #Сохраняет выбор языка  
для пользователя
```

```
# Редактируем текст сообщения, чтобы отобразить выбранный язык
```

```
bot.edit_message_text(  
    chat_id=call.message.chat.id,  
    message_id=call.message.message_id,  
    text=f"Вы выбрали язык: {LANGUAGES[lang_code]}",  
    reply_markup=None # Убираем клавиатуру  
)
```

```
# ввести текст для перевода
```

```
bot.send_message(  
    call.message.chat.id,  
    "Введите текст для перевода:",  
    reply_markup=trans_menu() #кнопка завершения перевода  
)
```

```
@bot.message_handler(func=lambda message: message.text == "Завершить  
перевод")
```

```
def end_trans(message):
```

```
    """Завершает режим перевода"""
```

```
    user_language.pop(message.chat.id, None) #удаляем текущий язык из  
словаря
```

```
bot.send_message(  
    message.chat.id,
```

```
"Перевод завершён. Выберите язык для перевода:",  
    reply_markup=lang_keyboard()  
)  
bot.send_message(  
    message.chat.id,  
    "Также доступно меню:",  
    reply_markup=main_menu() # клавиатура  
)
```

```
@bot.message_handler(func=lambda message: message.text == "Статистика )
```

```
def stats_button(message):
```

```
    """Отображает статистику переводов"""
```

```
    if message.chat.id in user_language:
```

```
        # Если пользователь находится в режиме перевода, просим завершить  
его переводж
```

```
        bot.send_message(  
            message.chat.id,
```

```
            "Вы сейчас находитесь в режиме перевода. Завершите перевод, чтобы  
увидеть статистику"
```

```
        )
```

```
    else:
```

```
        # Отображаем статистику
```

```
        stats_message = get_stats(message.chat.id)
```


```
        bot.send_message(  
            message.chat.id,
```

```

        f"Ваша статистика переводов:\n{stats_message}",

        reply_markup=main_menu()

    )

@bot.message_handler(func=lambda message: message.text == "Помощь ")
def help_button(message):

    """Отображает справочную информацию."""

    if message.chat.id in user_language:

        # Сообщаем что помощь недоступна в режиме перевода

        bot.send_message(

            message.chat.id,

            "Вы сейчас находитесь в режиме перевода. Завершите перевод, чтобы получить помощь."

        )

    else:

        # Показываем инструкции


        bot.send_message(

            message.chat.id,

            "Доступные функции:\n"

            "- Выберите язык для перевода.\n"

            "- Введите текст на русском, и я переведу его на выбранный язык.\n"

            "- Нажмите \"Статистика \", чтобы увидеть количество переводов по языкам.\n"

            "- Используйте \"Завершить перевод\", чтобы выбрать другой язык.",

            reply_markup=main_menu()

```


)

```
@bot.message_handler(func=lambda message: message.text == "Информация i")
```

```
def info_button(message):
```

```
    """Отображает информацию об авторе бота."""
```

```
    if message.chat.id in user_language:
```

```
        bot.send_message(
```

```
            message.chat.id,
```

```
            "Вы сейчас находитесь в режиме перевода. Завершите перевод, чтобы  
            получить информацию."
```

```
        )
```

```
    else:
```

```
        # отображаем информацию
```

```
        bot.send_message(
```

```
            message.chat.id,
```

```
            "Выполнил:\nИванов А. А. РТ5-31Б",
```

```
            reply_markup=main_menu()
```

```
        )
```

```
@bot.message_handler(func=lambda message: message.chat.id in user_language)
```

```
def trans_message(message):
```

```
    """Переводит введенный пользователем текст на выбранный язык."""
```

```
    lang_code = user_language[message.chat.id] # Получаем код выбранного  
    языка
```

```
    text_to_translate = message.text # Текст который нужно перевести
```

```

try:

    #перевод текста

    translated = translator.translate(text_to_translate, src="ru", dest=lang_code)

    bot.send_message(

        message.chat.id,

        f'Перевод ({LANGUAGES[lang_code]}):\n{translated.text}',

        reply_markup=trans_menu()

    )

    # обновляем статистику

    update_stats(message.chat.id, lang_code)

except Exception as e:

    # Обрабатываем ошибки и логируем их

    bot.send_message(message.chat.id, "Ошибка перевода.")

    logger.error(f"Ошибка перевода: {e}")

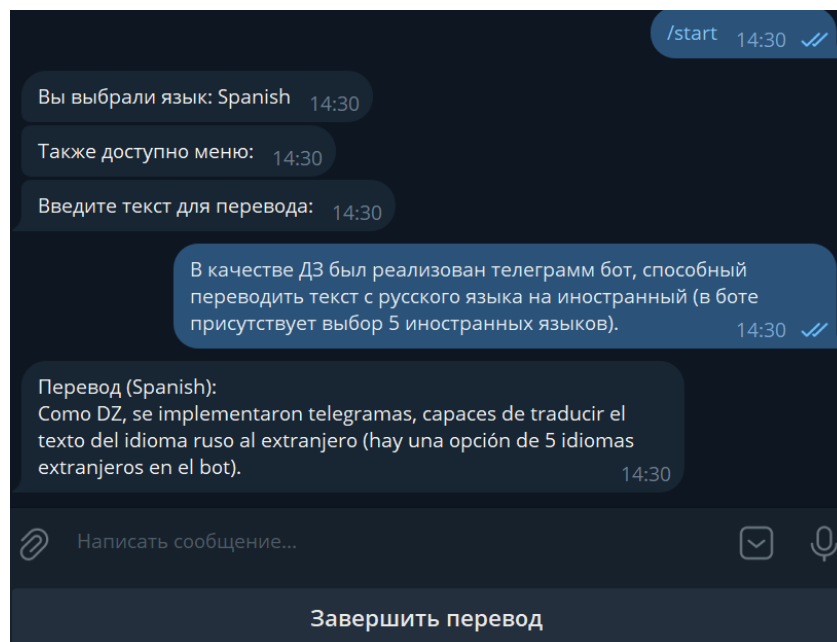
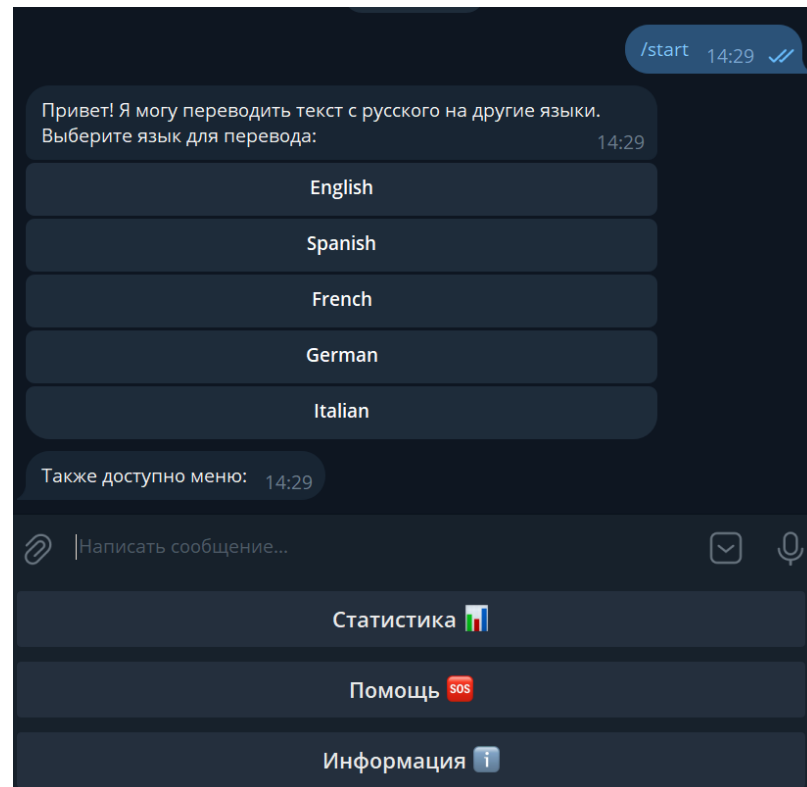

if __name__ == "__main__":

    print("Бот запущен...")

    bot.infinity_polling() # Запускаем бота

```

Вывод:



Также доступно меню: 14:30

Статистика 📊 14:30 ✓

Ваша статистика переводов:
Spanish: 1 14:30

Помощь 🆘 14:30 ✓

Доступные функции:
- Выберите язык для перевода.
- Введите текст на русском, и я переведу его на выбранный язык.
- Нажмите "Статистика 📊", чтобы увидеть количество переводов по языкам.
- Используйте "Завершить перевод", чтобы выбрать другой язык.

14:30

Информация ⓘ 14:30 ✓

Выполнил:
Иванов А. А. РТ5-31Б 14:30



Написать сообщение...



Статистика 📊

Помощь 🆘

Информация ⓘ