**УДК 004.43**

**РАЗРАБОТКА БОТА-ПЕРЕВОДЧИКА С ПОМОЩЬЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON**

**Мокин Н. А.1,**

 *Нижнетагильский государственный социально-педагогический институт, филиал Российского государственного профессионально-педагогического университета, РФ, г. Нижний Тагил Тагил, e-mail: snoypeee@mail.ru*

**Аннотация**

**В статье рассматривается процесс разработки бота на языке программирования python, для изучения иностранных слов. Обоснован выбор технологий и программных средств, используемых при разработке. Цель разрабатываемого чат-бота повышение навыков в новых словах.**

Ключевые слова: Python, Чатбот, Разработка бота, Переводчик.

**DEVELOPMENT A TRANSLATE-BOT FOR TELEGRAM IN PYTHON PROGRAMMING LANGUAGE**

**Mokin N. A.1**

 *Nizhny Tagil State Socio-Pedagogical Institute, branch of the Russian State Vocational Pedagogical University, Russia, Nizhny Tagil*

**The article discusses the process of developing a bot for learning foreign words. The choice of technologies and software tools used in the development is justified. The purpose of the chatbot being developed is to improve skills in new words.**

Keywords: Python, Chatbot, Translator, Development.

**Введение**

В цифровую эпоху все сидят в социальных сетях и общаются в мессенджерах, поэтому компании тоже переходят туда — для удобства клиентов. Боты нужны для выполнения рутинных, монотонных операций и упрощения некоторых действиях в интернете.

Чат-боты открывают людям новые возможности для улучшения себя и своих умений. В сети сложно найти хорошего чат-бота для изучения иностранных слов.

Благодаря данному боту «MemorizingWords» вы сможете изучить новые слова и прослушать, а также произносятся, а в будущем будут появляться больше дополнений для еще более удобной работы.

**Цель исследования** — изучить основы создания чат-ботов в мессенджере «Telegram».

Задачи работы:

* Провести сравнительный анализ языков программирования для разработки чат-бота;
* описать функции бота;
* разработать учебного бота «MemorizingWords» в среде Visual Studio с использованием языка программирования Python.

**Основы создания чат-бота в Telegram**

Телеграм стал основой в создании чат-ботов для мессенджеров. В нем работает большое количество ботов. Среди них виртуальные сервисы Почты России, Сбербанк-бот Телеграм, бот журнала «Медуза» и так далее. Отзывы о Телеграм-ботах подтверждают, что площадка очень удобна. Популярный мессенджер объединяет миллионы пользователей, и с каждым днем их становится все больше. Преимущество Телеграма в том, что у него есть собственный интерфейс для создания чат-ботов.

Телеграм имеет очевидные достоинства:

* повышенная безопасность и защита данных шифрованием;
* высокий уровень конфиденциальности за счет политики мессенджеров;
* открытое API, наличие подробной документации и живого сообщества разработчиков;
* отсутствие оплаты.

Разработка чат-бота для Telegram условно делится на два этапа:

* создание оболочки;
* программирование функционала.

Первым шагом разработки приложения является регистрация у специального чат-бота «BotFather» (см. рис. 7). Данный бот помогает создать оболочку чат-бота. Нет необходимости создавать чат-бота с нуля.

Регистрация начинается c команды «/newbot», после чего предлагается ввести название чат-бота с обязательным условием: конце названия указывается «Bot» или «\_bot». Имя для бота выбирается простое, для удобства и простоты поиска пользователями.

При выполнении всех условий, а именно введении уникального имени для бота, «BotFather» выдаст токен (специальный набор символов для доступа к НТТР АРI Telegram Bot) и URL-адрес для доступа к чат-боту. Далее заполняется профиль и настраивается меню.

С помощью BotFather можно добавить описание созданного чат-‘бота, добавить изображение профиля, изменить имя, присвоить текст, который ‘отображается при первом открытии чат-бота и др.

После настроек Telegram и получения токена можно приступить к разработке программной части бота. Для разработки использовались необходимые библиотеки на языке программирования Python.

В первую очередь библиотеки необходимы для возможности подключения чат-бота к библиотеке. Во время разработки в код указывается токен созданного чат-бота (см.Рис. 7).

‘

Рис. 7. Создание и получение API бота

 В этом боте есть еще несколько функций для редактирования и настройки бота они представлены в таблице 1.

 Таблица 1

Команды для изменения чат-ботов

|  |  |
| --- | --- |
| **Команда** | **Описание** |
| /setname | Изменяет существующее имя |
| /setdescription | Присваивает текст, который отображается при первом открытии чат-бота |
| /setabottext | Присваивает текст в поле «О чат-боте» |
| /setuserpic | Присваивает выбранную картинку |
| /setcommand | Позволяет создать список доступных команд |
| /deletebot | Удаляет выбранного чат-бота |

Помимо команд для изменения основных параметров чат-ботов существует ряд команд, которые позволяют выводить неизменяемые параметры (токен). Для изменения параметров токена используются различные команды, представленные в таблице 2.

Таблица 2.

Команды для дополнительной настройки чат-бота

|  |  |
| --- | --- |
| **Команда** | **Описание** |
| /token | Возвращает полученный ранее токен у выбранного бота |
| /revoke | Аннулирует токен доступа к боту |
| /setinline | Включает или выключает возможность вызывать, бота из других чатов |
| /setpprivacy | Включает режим конфиденциальности. |

Пользователи Telegram могут взаимодействовать с чат-ботами несколькими способами: используя команды с параметрами или встроенные клавиатуры.

Для разработки данного бот были подключены библиотеки Python такие как:

* google\_trans\_new Бесплатный и неограниченный API Python для Google Translate;
* gtts (Google Text-to-Speech) библиотека Python и инструмент командной строки для взаимодействия с API преобразования текста в речь Google Translate. Записать озвученные данные mp3 в файл;
* pyTelegramBotApi Простая, но расширяемая реализация Python для Telegram Bot API.

Для начала установим аватарку для нашего бота. Заходим в Bot\_Father пишем команду /setuserpic, выбираем нужного бота и вставляем выбранную картинку (см. Рис. 8).



Рис. 8. Установка главного изображения для бота

Далее мы сможем установить описание для бота, чтобы пользователи понимали, что делает наш бот. Для этого понадобится команда /setdescription (см.Рис. 9).

Рис. 9. Установка описания бота

Начало работы с чат-ботом начинается с команды /start (см. Рис. 10). Данная команда активирует команду и бот начинает диалог с пользователем. Код начала работы представлен на листинге 1.

Листинг 1

From google\_trans\_new import google\_translator
import gtts
import os
import telebot

bot = telebot.TeleBot('1822191029:AAHXDC6YojMAz5IpkK1-OPUh-MBubHiiC6U')

@bot.message\_handler(commands=['start'])def send\_start(message):
 bot.reply\_to(message, 'Привет, я бот для запоминания английских слов, надеюсь мы подружимся. Напиши мне любое слово, и я переведу и расскажу, как его правильно прочитать.')



Рис. 10. Приветственное сообщение

Далее следует реализация получения текста, который прислал пользователь, обращение к google переводчику и определение языка сообщения, код реализации представлен на листинге 2.

Листинг 2

@bot.message\_handler(content\_types='text'def do\_translate(message):
 translator = google\_translator()
 lang = translator.detect(message.text)

После реализации следует реализация определения языка для перевода и отправления произношения в mp3 формате (см. Рис. 11), код реализации представлен на листинге 3.

 Листинг 3

if lang[0] == 'ru':
 transed = translator.translate(message.text, lang\_tgt='en')
 voice = gtts.gTTS(transed)
 voice.save(transed + ".mp3")
 bot.reply\_to(message, transed)
 bot.send\_audio(message.chat.id, audio=open(transed + ".mp3", "rb"))
 os.remove(transed + ".mp3")
elif lang[0] == 'en'
 transed = translator.translate(message.text, lang\_tgt='ru')
 voice = gtts.gTTS(transed)
 voice.save(transed + ".mp3")
 bot.reply\_to(message, transed)
 bot.send\_audio(message.chat.id, audio=open(transed + ".mp3", "rb"))
 os.remove(transed + ".mp3")
else:
 bot.reply\_to(message, "Я вас не понял")



**Рис. 11.**

В конце программного кода реализован метод в случае какой-либо ошибки в чат-боте, код реализации представлен на листинге 4.

Листинг 4

if \_\_name\_\_ == '\_\_main\_\_':
 while True:
 try:
 bot.polling*(*none\_stop=True*)*
 except Exception as e:
 print*(*"error: " + str*(*e*))*

Весь код с комментариями представлен на листинге 5.

Листинг 5

from google\_trans\_new import google\_translator
import gtts
import os
import telebot # импортируем библиотеки

bot = telebot.TeleBot*(*'1822191029:AAHXDC6YojMAz5IpkK1-OPUh-MBubHiiC6U'*)* # токен от botfather

@bot.message\_handler*(*commands=*[*'start'*])*def send\_start*(*message*)*:
 bot.reply\_to*(*message, 'Привет, я бот для запоминания английских слов, надеюсь мы подружимся. '
 'Напиши мне любое слово, и я переведу и расскажу как его правильно прочитать.'*)* # сообщение при прописывание комманды start

@bot.message\_handler*(*content\_types='text'*)* # с помощью этого мы получаем текст который прислал поль3ователь
def do\_translate*(*message*)*:
 translator = google\_translator*()* #обращяемся к переводчику
 lang = translator.detect*(*message.text*)* #определяем язык сообщения пользователя

 if lang*[*0*]* == 'ru': # если язык русский
 transed = translator.translate*(*message.text, lang\_tgt='en'*)* # переводим
 voice = gtts.gTTS*(*transed*)* # делаем гс
 voice.save*(*transed + ".mp3"*)* # сохраняем гс
 bot.reply\_to*(*message, transed*)* #отправляем переведённый текст
 bot.send\_audio*(*message.chat.id, audio=open*(*transed + ".mp3", "rb"*))* # отправляем озвучку
 os.remove*(*transed + ".mp3"*)* # удаляем файл гс (озвучки)
 elif lang*[*0*]* == 'en':# если язык английский (далее код аналогичен вышеприведённому)
 transed = translator.translate*(*message.text, lang\_tgt='ru'*)* # только тут меняем lang\_tgt='ru' на ру (переводит с английского на русский)
 voice = gtts.gTTS*(*transed*)* voice.save*(*transed + ".mp3"*)* bot.reply\_to*(*message, transed*)* bot.send\_audio*(*message.chat.id, audio=open*(*transed + ".mp3", "rb"*))* os.remove*(*transed + ".mp3"*)* else:
 bot.reply\_to*(*message, "Я вас не понял"*)* # если нас чего-то не устраивает (например сообщение всё из символов)

if \_\_name\_\_ == '\_\_main\_\_':
 while True:
 try:
 bot.polling*(*none\_stop=True*)* #запускаем бота
 except Exception as e: # если ошибка
 print*(*"error: " + str*(*e*))* # от выведем её в терминал

**Заключение**

В результате работы были решены следующие задачи:

* проведен сравнительный анализ технических средств для разработки чат-бота, а именно рассмотрены общие понятия чат-бота и его преимущества, а также где применяются чат-боты, была представлена статистика и анализ языков программирования и выбран язык Python для разработки чат-бота.
* изучены новые библиотеки Python для разработки чат-бота. Разница между двумя популярными библиотеками для чат-ботов;
* разработан учебный чат-бот «MemorizingWords» в Telegram с использованием языка Python. Функционал чат-бота заключается в изучении и запоминания новых слов, а также в будущем тестах для запоминания все больших слов. Чат-бот был протестирован на компьютере и мобильном устройстве.

**Список литературы**

1. Библиотека AIOGram [Электронный ресурс]. – Режим доступа: https://github.com/aiogram/aiogram –Загл. с экрана.
2. Библиотека pyTelegramBotAPI [Электронный ресурс]. – Режим доступа: https://github.com/eternnoir/pyTelegramBotAPI#methods –Загл. с экрана.
3. Виды языков программирования [Электронный ресурс]. – URL: http://csaa.ru/vidy-jazykov-programmirovanija/ – Загл. с экрана.
4. Задачи, которые решают чат-боты [Электронный ресурс]. – Режим доступа: https://www.cossa.ru/trends/190984/ –Загл. с экрана.
5. Интернет-технологии [Электронный ресурс]. – Режим доступа: https://www.internet-technologies.ru/articles/php-ruby-python – Загл. с экрана.
6. Как создать телеграм-бота на облачных функциях [Электронный ресурс]. – Режим доступа: https://kb.selectel.ru/docs/selectel-cloud –Загл. с экрана.
7. Как создать своего бота в BotFather? [Электронный ресурс]. – Режим доступа: https://botcreators.ru/blog/kak-sozdat-svoego-bota-v-botfather/ –Загл. с экрана.
8. Какая библиотека для создания telegram бота на Python лучше и удобнее? [Электронный ресурс]. – Режим доступа: https://qna.habr.com/q/815357 –Загл. с экрана
9. Мы сделали из чатботов помощников, теперь пора сделать из них собеседников [Электронный ресурс]. – Режим доступа: https://tjournal.ru/tech/58633-my-sdelali-iz-chatbotov-pomoshchnikov-teper-pora-sdelat-iz-nih-sobesednikov –Загл. с экрана.
10. Основы программирования на примере языка Python [Электронный ресурс]. – Режим доступа: https://dfedorov.spb.ru/python3/book.pdf – Загл. с экрана.