

ISSN 2409-546X

ЮНЫЙ УЧЁНЫЙ

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



6+

8
2023

Юный ученый

Международный научный журнал

№ 8 (71) / 2023

Издается с февраля 2015 г.

Главный редактор: Ахметов Ильдар Геннадьевич, кандидат технических наук

Редакционная коллегия:

Жураев Хусниддин Олтинбоевич, доктор педагогических наук (Узбекистан)

Иванова Юлия Валентиновна, доктор философских наук

Каленский Александр Васильевич, доктор физико-математических наук

Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)

Куташов Вячеслав Анатольевич, доктор медицинских наук

Лактионов Константин Станиславович, доктор биологических наук

Сараева Надежда Михайловна, доктор психологических наук

Абдрасилов Турганбай Курманбаевич, доктор философии (PhD) по философским наукам (Казахстан)

Авдеюк Оксана Алексеевна, кандидат технических наук

Айдаров Оразхан Турсункожаевич, кандидат географических наук (Казахстан)

Алиева Тарана Ибрагим кызы, кандидат химических наук (Азербайджан)

Ахметова Валерия Валерьевна, кандидат медицинских наук

Бердиев Эргаш Абдуллаевич, кандидат медицинских наук (Узбекистан)

Брезгин Вячеслав Сергеевич, кандидат экономических наук

Данилов Олег Евгеньевич, кандидат педагогических наук

Дёмин Александр Викторович, кандидат биологических наук

Дядюн Кристина Владимировна, кандидат юридических наук

Желнова Кристина Владимировна, кандидат экономических наук

Жуйкова Тамара Павловна, кандидат педагогических наук

Игнатова Мария Александровна, кандидат искусствоведения

Искаков Руслан Маратбекович, кандидат технических наук (Казахстан)

Калдыбай Кайнар Калдыбайулы, доктор философии (PhD) по философским наукам (Казахстан)

Кенесов Асхат Алмасович, кандидат политических наук

Коварда Владимир Васильевич, кандидат физико-математических наук

Комогорцев Максим Геннадьевич, кандидат технических наук

Котляров Алексей Васильевич, кандидат геолого-минералогических наук

Кузьмина Виолетта Михайловна, кандидат исторических наук, кандидат психологических наук

Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)

Кучерявенко Светлана Алексеевна, кандидат экономических наук

Лескова Екатерина Викторовна, кандидат физико-математических наук

Макеева Ирина Александровна, кандидат педагогических наук

Матвиенко Евгений Владимирович, кандидат биологических наук

Матроскина Татьяна Викторовна, кандидат экономических наук

Матусевич Марина Степановна, кандидат педагогических наук

Мусаева Ума Алиевна, кандидат технических наук

Насимов Мурат Орленбаевич, кандидат политических наук (Казахстан)

Паридинова Ботагоз Жаппаровна, магистр философии (Казахстан)

Прончев Геннадий Борисович, кандидат физико-математических наук

Рахмонов Азизхон Боситхонович, доктор педагогических наук (Узбекистан)

Семахин Андрей Михайлович, кандидат технических наук

Сенцов Аркадий Эдуардович, кандидат политических наук

Сенюшкин Николай Сергеевич, кандидат технических наук

Султанова Дилшода Намозовна, доктор архитектуры (Узбекистан)

Титова Елена Ивановна, кандидат педагогических наук

Ткаченко Ирина Георгиевна, кандидат филологических наук

Федорова Мария Сергеевна, кандидат архитектуры

Фозилов Садриддин Файзуллаевич, кандидат химических наук (Узбекистан)

Яхина Асия Сергеевна, кандидат технических наук

Ячинова Светлана Николаевна, кандидат педагогических наук

Международный редакционный совет:

Айрян Заруи Геворковна, кандидат филологических наук, доцент (Армения)
Арошидзе Паата Леонидович, доктор экономических наук, ассоциированный профессор (Грузия)
Атаев Загир Вагитович, кандидат географических наук, профессор (Россия)
Ахмеденов Кажмурат Максutowич, кандидат географических наук, ассоциированный профессор (Казахстан)
Бидова Бэла Бертовна, доктор юридических наук, доцент (Россия)
Борисов Вячеслав Викторович, доктор педагогических наук, профессор (Украина)
Буриев Хасан Чутбаевич, доктор биологических наук, профессор (Узбекистан)
Велковска Гена Цветкова, доктор экономических наук, доцент (Болгария)
Гайич Тамара, доктор экономических наук (Сербия)
Данатаров Агахан, кандидат технических наук (Туркменистан)
Данилов Александр Максимович, доктор технических наук, профессор (Россия)
Демидов Алексей Александрович, доктор медицинских наук, профессор (Россия)
Досманбетов Динар Бакбергенович, доктор философии (PhD), проректор по развитию и экономическим вопросам (Казахстан)
Ешиев Абдыракман Молдоалиевич, доктор медицинских наук, доцент, зав. отделением (Кыргызстан)
Жолдошев Сапарбай Тезекбаевич, доктор медицинских наук, профессор (Кыргызстан)
Игисинов Нурбек Сагинбекович, доктор медицинских наук, профессор (Казахстан)
Кадыров Кутлуг-Бек Бекмурадович, доктор педагогических наук, и. о. профессора, декан (Узбекистан)
Каленский Александр Васильевич, доктор физико-математических наук, профессор (Россия)
Козырева Ольга Анатольевна, кандидат педагогических наук, доцент (Россия)
Колпак Евгений Петрович, доктор физико-математических наук, профессор (Россия)
Кочербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Куташов Вячеслав Анатольевич, доктор медицинских наук, профессор (Россия)
Кыят Эмине Лейла, доктор экономических наук (Турция)
Лю Цзюань, доктор филологических наук, профессор (Китай)
Малес Людмила Владимировна, доктор социологических наук, доцент (Украина)
Нагервадзе Марина Алиевна, доктор биологических наук, профессор (Грузия)
Нурмамедли Фазиль Алигусейн оглы, кандидат геолого-минералогических наук (Азербайджан)
Прокопьев Николай Яковлевич, доктор медицинских наук, профессор (Россия)
Прокофьева Марина Анатольевна, кандидат педагогических наук, доцент (Казахстан)
Рахматуллин Рафаэль Юсупович, доктор философских наук, профессор (Россия)
Ребезов Максим Борисович, доктор сельскохозяйственных наук, профессор (Россия)
Сорока Юлия Георгиевна, доктор социологических наук, доцент (Украина)
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)
Узаков Гулом Норбоевич, доктор технических наук, доцент (Узбекистан)
Федорова Мария Сергеевна, кандидат архитектуры (Россия)
Хоналиев Назарали Хоналиевич, доктор экономических наук, старший научный сотрудник (Таджикистан)
Хоссейни Амир, доктор филологических наук (Иран)
Шарипов Аскар Калиевич, доктор экономических наук, доцент (Казахстан)
Шуклина Зинаида Николаевна, доктор экономических наук (Россия)

СОДЕРЖАНИЕ

РУССКИЙ ЯЗЫК

Фирстова М. В.

Русские слова, заимствованные из английского языка 1

ЛИТЕРАТУРА

Гамалей С. А.

Проблема социального сиротства в произведении Ганса Христиана Андерсена «Девочка со спичками» 4

Завальнова А. А.

Калужане в «Записках охотника» Тургенева и их реальный исторический портрет 6

Королев И. Е.

Фольклорные и литературные источники повести Н. В. Гоголя «Вий» 8

ИСТОРИЯ

Валиуллина З. Р., Хакимова С. И.

Исследование судебных процессов в СССР и Российской Федерации, посвященных раскрытию обстоятельств геноцида мирных советских граждан в годы Великой Отечественной войны 1941–1945 гг. (на базе «Точки роста» МБОУ СОШ № 1 с. Аскино Республики Башкортостан) 11

Драгун Е. А.

Вклад семьи Чуйко в историю страны 21

Леонов Д. Р.

Александр Мюнхенский: святой мученик движения Сопротивления 25

ОБЩЕСТВОЗНАНИЕ

Барков А. М.

Влияние компьютера на развитие ребенка 28

Хмель В. В.

Выбор моего будущего местожительства 36

ЭКОНОМИКА

Eleubayuly N.

The awareness of Kazakhstan school students in financial literacy and their interest 40

ИНФОРМАТИКА

Ким А. С.

Роль хэширования в работе «белого хакера» 45

Урбан И. Б.

Telegram-бот «Фонетический разбор слова» на Python 50

ФИЗИКА

Закиров Д. И.

Разработка портативной метеостанции на базе микроконтроллера ESP8266 56

Полицина П. А.

Сверхмассивные темные звезды: обзор теории 60

БИОЛОГИЯ

Батуро П. Р., Лигачева В. С., Селифонова Д. Р., Тихонова В. Н.

Оценка адаптационного потенциала растений культуры *in vitro* в условиях Ботанического сада имени Б. В. Гроздова 63

<i>Железнов С. И.</i> Разработка метода количественной оценки риска инвазии людей церкариозом	67
<i>Ильматова А. В., Путинцева Е. М.</i> Влияние зубных паст и зубных щеток на микрофлору ротовой полости	75
<i>Милькова Д. В.</i> Насекомые-опылители Ленинградской области	80
<i>Садьков А. И.</i> Изучение оптимальных условий выращивания томатов на аэропонике	84
<i>Филипченко А. А.</i> Черенкование как способ вегетативного размножения. Виды черенкования	89
ЕСТЕСТВОЗНАНИЕ	
<i>Крученкова И. В.</i> Растения на страже здоровья	91
<i>Тарасов Ю. И.</i> Модель музыкальной радиостанции в жизни ребёнка	93
ФИЗИЧЕСКАЯ КУЛЬТУРА	
<i>Костина А. И.</i> Спортивное молодежное средство массовой информации	95
ЭКОЛОГИЯ	
<i>Белоножкина А. А.</i> Бумага в домашних условиях — это возможно!	97
<i>Рябухина А. Н.</i> Вода как важная составляющая всего живого на Земле	100
ПЕДАГОГИКА И ПСИХОЛОГИЯ	
<i>Доронин Р. С., Сороколетов Е. С.</i> К вопросу о популяризации профессий аграрной сферы среди выпускников школ	103
<i>Трепакова Т. В.</i> Управление чувством гнева	106

Telegram-бот «Фонетический разбор слова» на Python

Урбан Иван Борисович, учащийся 8-го класса

Научный руководитель: Симаков Егор Евгеньевич, учитель математики, информатики и ИКТ
МАОУ Лицей № 1 г. Южно-Сахалинска

На сегодняшний день различные мессенджеры, такие как Telegram, используются не только для переписки с другом или поздравления с праздником. Для многих они стали одним из основных источников информации, средством делового общения, неотъемлемой частью ведения бизнеса. При этом количество пользователей постоянно растет. А для того, чтобы помочь владельцам крупных каналов в выполнении рутинных задач, были разработаны боты — небольшие приложения, которые самостоятельно выполняют заранее созданные задачи.

Ключевые слова: telegram, бот, python, программирование

О telegram-ботах

Сегодня telegram-боты — популярный способ получения, обработки и распространения информации. Боты используются многими сообществами, сервисами и отдельными пользователями сети Интернет. Telegram-бот работает непосредственно через интерфейс приложения Telegram, он имитирует действия «живого» пользователя. Многие компании, развивающие бизнес через глобальную сеть Интернет, используют возможности ботов для коммуникации с целевой аудиторией и выполнения однообразной рутинной работы (без использования наемных сотрудников). Однако их и в личных целях обычных пользователей. Идея данного проекта возникла следующим образом. Младший брат автора исследования учится во втором классе начальной школы. В настоящее время, в рамках школьной программы он изучает тему «Фонетический разбор слова», и иногда у него возникают некоторые трудности с корректным разбором слова. Улучшить навыки по данной теме может помочь telegram-бот. Его можно использовать для тренировки и при проверке домашних заданий, самоконтроле. При этом написанный алгоритм имеет возможность масштабирования и адаптации для решения не только данной конкретной задачи, но и серии схожих задач.

Telegram-бот состоит из двух частей: «backend» — внутренняя часть программы, отвечающая за обработку информации; «frontend» — визуализация данных и взаимодействие с пользователем. Новые боты создаются с помощью специальной утилиты «@BotFather» от создателя платформы Telegram, Павла Дурова. В Telegram используется один общий вид ботов, которых от обычных пользователей отличает только наличие приставки «bot» в имени. Сами же боты делятся на несколько направлений:

- Чат-боты, имитирующий общение на заданную тематику.
- Боты-информаторы о тех или иных событиях (новости, мероприятия).
- Игровые боты — в основном, это текстовые версии разных игр.
- Боты-асистенты, разработанные онлайн-сервисами как дополнение к основной веб-версии.

О разработке telegram-ботов

Для написания telegram-бота использовался язык программирования Python. Язык Python имеет несколько стандартных примеров использования при разработке приложений, в числе которых Веб-разработка на стороне сервера (backend). Она включает в себя сложные серверные функции, с помощью которых веб-сервисы отображают информацию для пользователя. Например, веб-сайты должны взаимодействовать с базами данных и другими веб-сайтами, а также защищать данные при их отправке по сети. Одним из преимуществ языка Python является наличие большого количества свободно распространяемых библиотек — наборов часто используемых функций, которые разработчики могут включать в свои программы Python, чтобы не писать код с нуля. По умолчанию в Python доступна стандартная библиотека, которая содержит большое количество многократно используемых функций. Кроме того, доступно более 137 000 библиотек Python для различных задач. Чтобы использовать библиотеки в программе, нужно их сначала установить в консоли командой `pip install__`. Далее в коде программы, перед использованием библиотеки нужно импортировать её командой `import__` или `from __ import__`. В данном проекте использовалась библиотека «aiogram», которая предоставляет наиболее полный объем функций для создания telegram-ботов.

Общий алгоритм работы любого telegram-бота достаточно прост:

- бот имеет токен (уникальный набор цифр и букв), по которому устанавливается соединение программного обеспечения с ботом;
- сообщения, команды и запросы, отправленные пользователями, передаются на программное обеспечение, запущенное на серверах разработчиков;
- получая сообщение или команду, бот действует по заданному сценарию.

С помощью библиотеки «aiogram» данный общий алгоритм реализуется следующим образом. Библиотека использует стандартный токен telegram-бота и устанавливает соединение с платформой Telegram. Далее, с помощью блока кода handler telegram-бот «ловит» сообщения пользователя либо по определенному тексту, либо по команде. Под каждый handler (блок кода, отвечающий за получение сообщений от пользователей и формирование ответов) необходимо создать функцию, описывающую действия бота после срабатывания хэндлера. По данному алгоритму бот будет обрабатывать информацию и отправлять ответ человеку.

В примере, приведенном далее, хэндлер получает сообщение «/start» и отправляет человеку в ответ сообщение «Привет».

```
@dp.message_handler(commands='start')
async def test(msg: types.Message):
    await msg.answer('Привет')
```

Разработка telegram-бота «Фонетический разбор слова»

Далее приведены основные этапы создания telegram-бота для фонетического разбора слова. Сначала создадим основной функционал программы — то есть разбор слова. Затем напишем работу бота в телеграмме.

Для проверки слова и его фонетического разбора, нужна справочная информация. Создадим в отдельном файле «мини словарь».

```
vowels = ['a', 'y', 'o', 'ы', 'э', 'я', 'ю', 'ё', 'и', 'е']
consonants = ['б', 'в', 'г', 'д', 'ж', 'з', 'й', 'к', 'л', 'м', 'н',
              'п', 'р', 'с', 'т', 'ф', 'х', 'ц', 'ш', 'щ']
nn_let = ['ь', 'ъ']
special_vowels = ['я', 'ю', 'ё', 'е', 'я', 'ю', 'ё', 'е']
spec_vow_key = {'я': ['й', 'а'], 'ю': ['й', 'у'], 'ё': ['й', 'о'],
               'е': ['й', 'э']}
voice_cons = ['б', 'в', 'г', 'д', 'ж', 'з', 'л', 'м', 'н', 'р', 'й']
vo_co_double = ['б', 'в', 'г', 'д', 'ж', 'з']
vo_co_undouble = ['л', 'м', 'н', 'р', 'й']
dull_cons = ['к', 'п', 'с', 'т', 'ф', 'х', 'ц', 'ч', 'ш']
du_co_double = ['к', 'п', 'с', 'т', 'ф']
du_co_undouble = ['х', 'ц', 'ч', 'ш']
vo_co_doub = {'б': 'п', 'в': 'ф', 'д': 'т', 'з': 'с', 'ж': 'ш', 'г': 'х'}
du_co_doub = {'п': 'б', 'ф': 'в', 'т': 'д', 'с': 'з', 'ш': 'ж', 'к': 'г'}
```

Чтобы этим словариком можно было пользоваться, импортируем эти справочные списки в файл с основными функциями

```
from Fonetic.Modules.info_back_end import vowels, \
    consonants, nn_let, special_vowels, spec_vow_key, \
    voice_cons, vo_co_double, vo_co_undouble, dull_cons, \
    du_co_double, du_co_undouble, vo_co_doub, du_co_doub
```

Первая функция будет преобразовывать заданное пользователем слово в список из букв это слова, для дальнейшей работы с ним. В ней работает цикл, который «пробегают» по слову занося каждую букву в список. В конце функция соответственно вернёт список из букв заданного слова.

```
def list(word): # Первая функция
    word_list = [] # Создание списка
    for i in word: # Цикл
        word_list.append(i)
    return word_list # Окончание, возвращение списка
```

Вторая функция «sound_count», в свою очередь, берёт за исходные данные список, который составит предыдущая функция. В ней также за основу берётся цикл, который проверяет каждую букву с помощью созданного нами «мини словарика» и определяет, сколько звуков она даёт. Выводит эта функция уже строку типа «n букв и x звуков».

```
def sound_count(word_list): # Вторая функция
    sound_quant = 0
    letter_quant = len(word_list) # Подсчет количества букв в слова
    index = 0
    for i in word_list: # Цикл, считающий количество звуков в слове
        if i in consonants:
            sound_quant += 1
        elif i in vowels:
            if i in special_vowels:
                if index == 0:
                    sound_quant += 2
                elif index > 0:
                    index_let = word_list.index(i, index)
                    if word_list[index_let - 1] in nn_let or \
                       word_list[index_let - 1] in vowels:
                        sound_quant += 2
                    elif word_list[index_let - 1] in consonants:
                        sound_quant += 1
            else:
                sound_quant += 1
        elif i in nn_let:
            pass
        index += 1
    # Формирование конечного вывода
    result = f'{letter_quant} букв и {sound_quant} звуков'
    # Возвращение итогового результата
    return result
```

Третья функция «every_let» будет составлять характеристику звука каждой буквы. Код данной функции занимает 215 строк, поэтому опишем его алгоритм без листинга. За входные данные функция принимает список букв, созданный первой функцией, но к нему добавляется ещё один параметр, указываемый пользователем — номер ударного слога, в слове. С помощью цикла «пробегают» по списку букв и подробно описывают звуки, даваемые каждой буквой. В начале функции создаётся главный список «sounds» для характеристик звуков. А цикл записывает каждую характеристику как ещё один временный список «snd1» и его добавляет в главный список. В конце функция вернёт объёмный основной список «sounds».

В начале основного файла импортируем библиотеку aiogram, токен и справочные списки из «словарика»

```
import config
import asyncio
from aiogram import Dispatcher, types
from Fonetic.Modules.func_back_end import sound_count, every_let, list
from Fonetic.Modules.info_back_end import special_vowels, vowels
```


Создадим первый Handler. Реагировать он будет на команду «start», в ответ поприветствует и отправит инструкцию пользования.

```
@dp.message_handler(commands='start')
async def cmd_start(msg: types.Message):
    await msg.answer("Привет, я могу сделать тебе\n"
                    "фонетический разбор слова\n"
                    "Используй команду '/f', /n"\n"
                    "напиши своё слово через пробел и\n"
                    "еще через пробел n-ый ударный слог\n"
                    "Пример: '/f слово n'")
```

Создадим второй хандлер. Реагировать он будет на команду «f»: «вырезать» слово и номер ударного слога из сообщения пользователя и передавать функциям для расчета. В нём будет функция, преобразовывающая большой список характеристик звуков из третьей функции «every_let» в читабельное сообщение. И это сообщение он отправит пользователю.

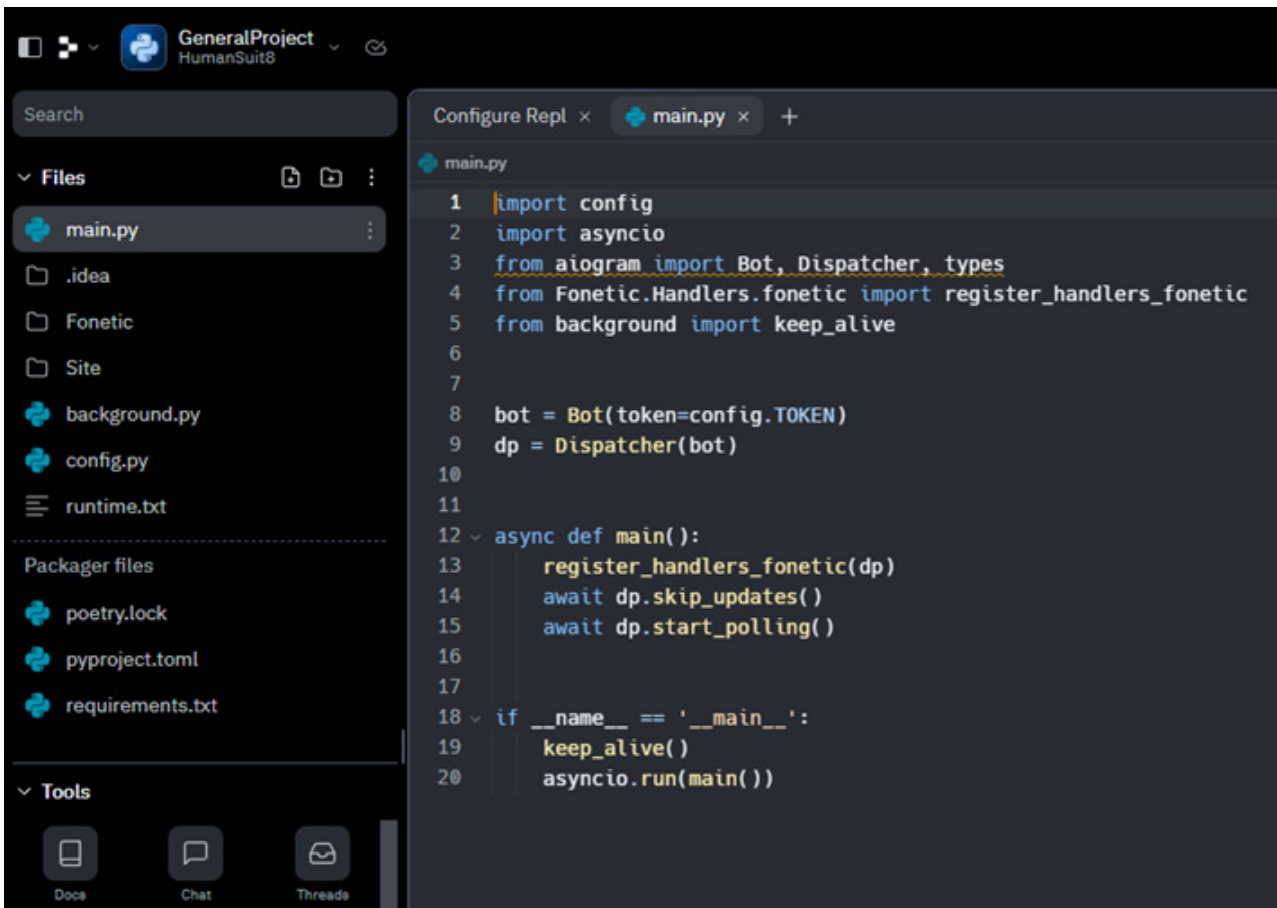
```
@dp.message_handler(commands='f')
async def cmd_fonetic(msg: types.Message):
    hit = int(msg.text[-1:])
    word = msg.text[3:-2]
    word_list = list(word)
    await msg.answer(f'{word} - {sound_count(word_list)}')
    ev_let = every_let(word_list, hit)
    output = ''
    try:
        index = 0
        for p in ev_let:
            if p[0] in special_vowels:
                if p[0] in special_vowels and p[1] in vowels:
                    output += (f'{p[0]} - [{p[1]}] - {p[2]}{p[3]}\n')
                elif p[0] in special_vowels:
                    x = f'{p[0]} - '
                    for d in p[1]:
                        x += d
                    output += (f'{x}\n')
                output += (f' - {p[2]}{p[3]}{p[4]}\n')
            else:
                y = ''
                for a in p:
                    y += a
                output += (f'{y}\n')
            index += 1
    except:
        pass
    await msg.answer(output)
```

Запуск telegram-бота «Фонетический разбор слова»

После завершения написания полного кода готовой программы, её нужно запустить на исполнение, а также разместить на одном из публичных сайтов в глобальной сети Интернет для постоянной работы. Это можно сделать несколькими способами:

1. Запустить программу на своём компьютере с постоянным доступом в сеть Интернет, но тогда ПК должен быть постоянно включен, что сопровождается затратами электроэнергии и трафика доступа в сеть.
2. Запустить на сторонних облачных сервисах: Heroku, AWS, Replit.
3. Запустить на домашнем сервере, при этом необходимо знать особенности настройки сетевого оборудования.

Конечно, способ необходимо выбирать, исходя из целей проекта и возможностей разработчика. В рамках данного исследования оптимальным является второй способ. А именно, бот был размещен на бесплатном сервисе Replit, от создателей портала GitHub. По факту это редактор программного кода, доступный онлайн, с авторизацией и сохранением проектов в облаке.

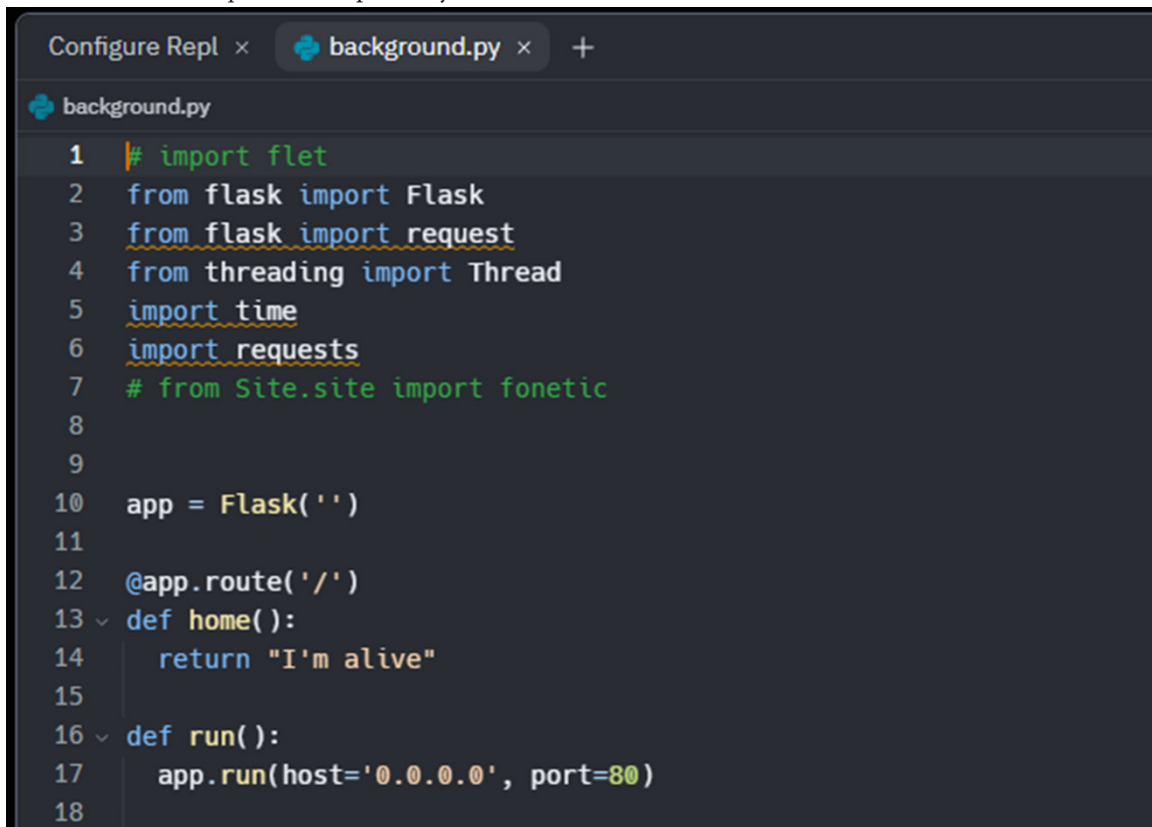


```

1 import config
2 import asyncio
3 from aiogram import Bot, Dispatcher, types
4 from Fonetic.Handlers.fonetic import register_handlers_fonetic
5 from background import keep_alive
6
7
8 bot = Bot(token=config.TOKEN)
9 dp = Dispatcher(bot)
10
11
12 async def main():
13     register_handlers_fonetic(dp)
14     await dp.skip_updates()
15     await dp.start_polling()
16
17
18 if __name__ == '__main__':
19     keep_alive()
20     asyncio.run(main())

```

На публичном ресурсе Replit можно запускать свой собственный код, но в бесплатной версии тарифа он выключится через 30 минут. Однако, есть способ обойти данное ограничение. Этот способ изложен на популярном информационном ресурсе habr.com. Суть метода заключается в том, что к основному файлу добавляется функция `keep_alive()` из дополнительного файла, которая запускает небольшой сайт на библиотеке flask.



```

1 # import flet
2 from flask import Flask
3 from flask import request
4 from threading import Thread
5 import time
6 import requests
7 # from Site.site import fonetic
8
9
10 app = Flask('')
11
12 @app.route('/')
13 def home():
14     return "I'm alive"
15
16 def run():
17     app.run(host='0.0.0.0', port=80)
18

```

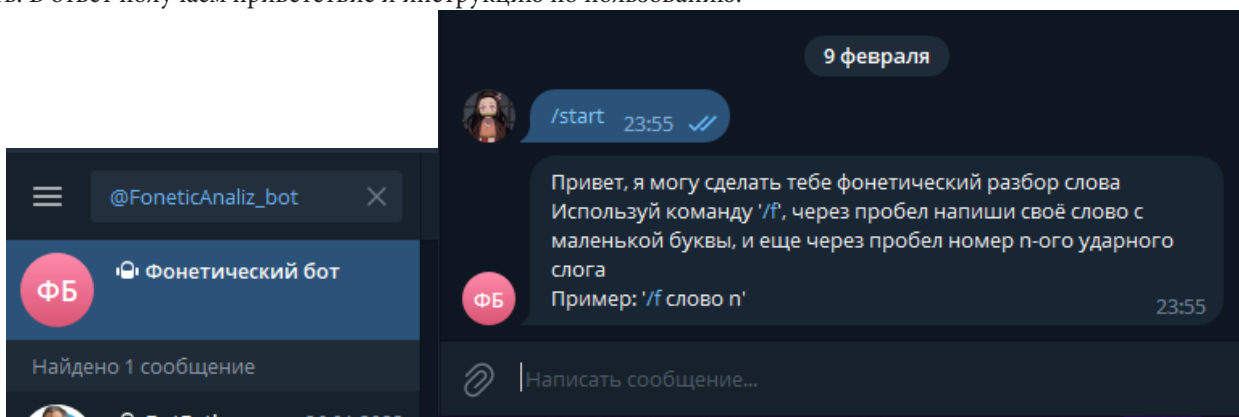
```

19 # def run2():
20 #     flet.app(target=fonetic, view=flet.WEB_BROWSER)
21
22 def keep_alive():
23     t = Thread(target=run)
24     # t2 = Thread(target=run2)
25     t.start()
26     # t2.start()

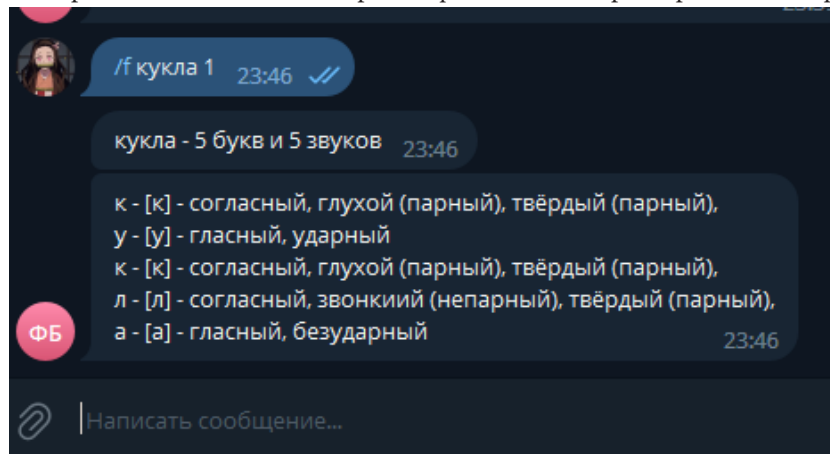
```

Также потребуется использование сторонней службы мониторинга «UpTimerRobot», которая по настройке будет заходить на сайт каждые 5 минут. Тем самым будет создаваться иллюзия «живого» пользователя из-за чего Replit будет думать, что на сайт постоянно кто-то заходит, и продлевать время работы.

Заключительным этапом является проверка работоспособности telegram-бота. Для чего в поисковой строке Telegram вписываем имя пользователя бота «@FoneticAnaliz_bot». Далее пишем «/start» или нажимаем на кнопку Запустить. В ответ получаем приветствие и инструкцию по пользованию.



Следуя инструкции, отправляем сообщение с запросом фонетического разбора слова, например, «кукла»:



Убеждаемся, что разработанный telegram-бот работает. Кроме того, любой желающий может проверить работоспособность telegram-бота, найдя его по имени пользователя, указанному выше.

ЛИТЕРАТУРА:

1. Гаско, Р. Простой Python просто с нуля / Р. Гаско — М.: Солон-Пресс, 2019. — 256 с. — ISBN 978-5-91359-334-4.
2. Мэтиз, Э. Рик Гаско Простой Python просто с нуля / Э. Мэтиз — С-Пб.: Питер, 2020. — 512 с. — ISBN 978-5-4461-1528-0.
3. Портал «Habr» — URL: <https://habr.com/ru/all/> (дата обращения: 24.01.2023).
4. Портал «tlgrm». Статья «Боты: информация для разработчиков» — URL: <https://tlgrm.ru/docs/bots> (дата обращения: 03.12.2022).