

Московский государственный технический
университет имени Н. Э. Баумана.

Факультет “Радиотехнический”
Кафедра ИУ5 “Системы обработки информации и управления”

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

Выполнил:

Студент группы РТ5-31Б

Цыгичко А.Н.

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2024 г.

Код программы:

Main.py

```
import logging
from telegram.ext import Updater
from telegram.ext import CommandHandler
from telegram import ReplyKeyboardMarkup

import sqlite3

logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.DEBUG
)
start_keyboard = [['/words', '/pictures'],
                  ['/top', '/start']]
start_markup = ReplyKeyboardMarkup(start_keyboard, one_time_keyboard=True)
logger = logging.getLogger(__name__)

TOKEN = '5370722364:AAHiDt3uZTlQXYwK56NrGMYjbFlqf8Wgtuc'

connect_users = sqlite3.connect("words.db", check_same_thread=False)
cursor_users = connect_users.cursor()
users = cursor_users.execute("SELECT * FROM users").fetchall()
id_of_newloh = 0
try:
    id_of_newloh = users[-1][0]
except:
    id_of_newloh = 0
connect_users.commit()
import words_learning
import pictures_learning

def start(update, context):
    global users, id_of_newloh
    print("XXXXXXXXXXXXXXXXXXXXXXXXXXXX", any(update.message.from_user.username
in i for i in users))
    update.message.reply_text(
        "Привет, " + update.message.from_user[
            'first_name'] + ", Я Ботяра.\nЯ обучу тебя английскому.\nДля
начала выбери режим.",
        reply_markup=start_markup)
    if not any(update.message.from_user.username in i for i in users):
        users.append((id_of_newloh + 1, update.message.from_user.username))
        cursor_users.execute(
            f"INSERT INTO users VALUES({id_of_newloh + 1},
'{update.message.from_user.username}',0 ) ").fetchall()
        connect_users.commit()
        id_of_newloh += 1

def help(update, context):
    update.message.reply_text(
        "Вы можете выбрать 2 режима:\nОтгадывание слова по картинке или
перевод слова с английского на русский.\n"
        "Также вы можете взять подсказку, чтобы получить 3 варианта
ответа.\n"
        "В боте есть свой рейтинг. За каждое правильноотгаданное слово вы
получаете баллы.\nПри использовании подсказки баллы уменьшаются вдвое\n"
        "Чтобы узнать топ пользователей введите команду /top")
```

```

def top(update, context):
    ret = ""
    top_scores = cursor_users.execute("SELECT name, score FROM users ORDER
BY score DESC LIMIT 10").fetchall()
    print(top_scores)
    for i, aboba in enumerate(top_scores):
        ret += str(i+1) + " @" + str(aboba[0]) + " " + str(aboba[1]) + "\n"
    update.message.reply_text(ret)

def main():
    updater = Updater(TOKEN)
    dp = updater.dispatcher
    dp.add_handler(CommandHandler("start", start))
    dp.add_handler(CommandHandler("help", help))
    dp.add_handler(CommandHandler("top", top))
    dp.add_handler(words_learning.conv_handler)
    dp.add_handler(pictures_learning.conv_handler)
    updater.start_polling()
    updater.idle()

if __name__ == '__main__':
    main()

```

parserDB.py

```

import sqlite3

c = sqlite3.connect("words.db")
cur = c.cursor()
with open("slova.txt", encoding='utf-8') as a:
    for i in a.readlines():
        v = i[:i.index("[") - 1] + "::::::::" + i[i.index("]") + 2:-
1].lower()
        f, g = v.split("::::::::")
        cur.execute(f"INSERT INTO words VALUES(NULL, '{f}', '{g}')"
).fetchall()
    c.commit()
    print(g)

```

picturesLearning.py

```

import logging
from telegram.ext import Updater, MessageHandler, Filters
from telegram.ext import CommandHandler, CallbackQueryHandler
from telegram import ReplyKeyboardMarkup
from telegram.ext import ConversationHandler
from random import randint, sample, choice, shuffle
from main import start_markup
import sqlite3
from gtts import gTTS

import os

all_pics = os.listdir("C:\\Users\\Alex4\\PycharmProjects\\bot\\pictures")

```

```

z = [i[:-4] for i in all_pics]
id = 0
user_id = -1
connect_points = sqlite3.connect("words.db", check_same_thread=False)
cursor_points = connect_points.cursor()
stop_keyboard = [['/stop']]
stop_markup = ReplyKeyboardMarkup(stop_keyboard, one_time_keyboard=True)
da_net = [['Да', 'Нет']]
da_net_markup = ReplyKeyboardMarkup(da_net, one_time_keyboard=True)
hint_list = []
win_stickers = ["CAACAgIAAxkBAAEEroBieAOdhgs-63vqO-
d5o7I_P5Sz7gACbRQAavh48Ev_35tLbqKxRyQE",

"CAACAgIAAxkBAAEEroRieAVyHEGctGEtQkpc4NRMqsbHgAC6yMAAj_lAUl35ZIodwxRkyQE",
"CAACAgIAAxkBAAEEropieAWENVIGRWl9FxSZ60kJGWh3yAACUBIAAmiK-
UusK6lyxrWHkyQE",

"CAACAgIAAxkBAAEEroxieAWGPiIt0d8HJ2thZL8um0yrfsgACShcAAAnq68EsNzlcoSPmYyyQE",

"CAACAgIAAxkBAAEErpBieAYI3GRqblYOW79n540ltYe7nQACDhsAApwiuEkaScQf14vkKyQE",

"CAACAgIAAxkBAAEErpJieAYOLlTMD8eEek4pN3mALtHfEQAC4RUAArRM2EnZ3DPB3rAICCCQE"]
lose_stickers =
["CAACAgIAAxkBAAEEroZieAV7k0CAKbzo3feMb5P5mmtxhQACgxAAAmuA8Uvd1I_YEKj09yQE"
,

"CAACAgIAAxkBAAEErohieAWAW8NFVMDoPFYravCidPgN2wACcBIAAt6p8Et8ICHIsOd3qyQE",

"CAACAgIAAxkBAAEEro5ieAWPM_NAbkQxQXzXKMQCJ2vBQQACmRUAAitleEhlyI6rouJTisQE"]

def pics(update, context):
    global z, id, user_id
    user_id = cursor_points.execute(
        "SELECT id FROM users WHERE name = '" +
update.message.from_user.username + "'" ).fetchall()[0][0]
    connect_points.commit()
    update.message.reply_text("Чтобы остановить игру напиши команду /stop",
reply_markup=stop_markup)
    id = randint(0, len(z) - 1)
    update.message.reply_text("Напишите слово изображенное на картинке на
английском языке")

update.message.reply_photo(open("C:\\Users\\Alex4\\PycharmProjects\\bot\\pi
ctures\\" + z[id] + '.jpg', 'rb'))
    return 1

def stop(update, context):
    update.message.reply_text("Хапоooooooooооо!", reply_markup=start_markup)
    return ConversationHandler.END

def question(update, context):
    global z, id
    v = update.message.text
    print(z[id])
    if v.lower() == z[id]:
        update.message.reply_text(
            "Абсолютно верно 👍, ваши баллы были начислены вам на счет. ",
            reply_markup=da_net_markup)

```

```

        update.message.reply_sticker(choice(win_stickers))
        update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
        print(user_id)
        print(cursor_points.execute("UPDATE users SET score = score + ?
WHERE id = ?", (10, user_id)).fetchall())
        connect_points.commit()
        return 6
    else:
        update.message.reply_text("Ответ не верный☹️((((",
                                reply_markup=da_net_markup)
        update.message.reply_sticker(choice(lose_stickers))
        update.message.reply_text("Хотите воспользоваться подсказкой?
(Да,Нет)")
        return 5

def hint(update, context):
    global z, id
    v = update.message.text
    print(z[id])
    if v.lower() == z[id]:
        update.message.reply_text(
            "Абсолютно верно 🏆, ваши баллы были начислены вам на счет. ",
            reply_markup=da_net_markup)
        update.message.reply_sticker(choice(win_stickers))
        update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
        print(cursor_points.execute("UPDATE users SET score = score + ?
WHERE id = ?", (5, user_id)).fetchall())
        connect_points.commit()
        return 6
    else:
        update.message.reply_text("Ответ не верный☹️((((",
                                reply_markup=da_net_markup)
        update.message.reply_text("Правильный ответ: " + z[id])
        update.message.reply_sticker(choice(lose_stickers))
        update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
        return 6

def hintornot(update, context):
    global z, id, hint_list
    v = update.message.text
    if v.lower() in "нет":
        update.message.reply_text("Хотите продолжить игру?",
                                reply_markup=da_net_markup)
        return 4
    elif v.lower() in "да":
        hint_list = [z[id]] + sample(z, 2)
        while len(set(hint_list)) != 3:
            hint_list = list(set(hint_list))
            hint_list.append(choice(z))
        shuffle(hint_list)
        update.message.reply_text("Выберите 1 слово",
                                reply_markup=ReplyKeyboardMarkup([hint_list], one_time_keyboard=True))
        return 2

```



```

from main import start_markup
from gtts import gTTS
stop_keyboard = [['/stop']]
stop_markup = ReplyKeyboardMarkup(stop_keyboard, one_time_keyboard=True)
da_net = [['Да', 'Нет']]
da_net_markup = ReplyKeyboardMarkup(da_net, one_time_keyboard=True)

win_stickers = ["CAACAgIAAxkBAAEEroBieAOdhgs-63vqO-
d5o7I_P5Sz7gACbRQAAvh48Ev_35tLbqKxRyQE",

"CAACAgIAAxkBAAEEroRieAVyHEGctGEtQkpc4NRMqsbgHgAC6yMAAj_1AUl35ZIodwxRkyQE",
"CAACAgIAAxkBAAEEropieAWENVIGRWl9FxSZ60kJGWh3yAACUBIAAmiK-
UusK6lyxrWHkyQE",

"CAACAgIAAxkBAAEEroxieAWGPit0d8HJ2thZL8um0yrfsgACShcAAAnq68EsNzlcoSPmYyyQE",

"CAACAgIAAxkBAAEErpBieAYI3GRqblYOW79n54OltYe7nQACDhsAApwiuEkaScQf14vkKyQE",

"CAACAgIAAxkBAAEErpJieAYOLlTMD8eEek4pN3mALtHfEQAC4RUAArRM2EnZ3DPB3rAICCQE"]
lose_stickers =
["CAACAgIAAxkBAAEEroZieAV7k0CAKbzo3feMb5P5mmtxhQACGxAAAmuA8Uvd1I_YEKj09yQE"
,

"CAACAgIAAxkBAAEErohieAWAW8NFVMDoPFYravCidPgN2wACcBIAAt6p8Et8ICHIsOd3qyQE",

"CAACAgIAAxkBAAEEro5ieAWPM_NAbkQxQXzXKMQCJ2vBQQACmRUAAitleEhlyI6rouJTisQE"]

connect_words = sqlite3.connect("words.db", check_same_thread=False)
cursor_words = connect_words.cursor()

id = 0
z = []
hint_list = []
user_id = -1

def words(update, context):
    global z, id, user_id
    user_id = cursor_words.execute(
        "SELECT id FROM users WHERE name = '" +
update.message.from_user.username + "'"").fetchall()[0][0]
    connect_words.commit()
    update.message.reply_text("Чтобы остановить игру напиши команду /stop",
reply_markup=stop_markup)
    z = cursor_words.execute("SELECT * FROM words").fetchall()
    connect_words.commit()
    id = randint(0, len(z) - 1)
    update.message.reply_text("Переведите слово(перевод должен быть в
несовершенном виде инфинитива)\n" + z[id][1])
    return 1

def stop(update, context):
    update.message.reply_text("Хапоooooooooш!", reply_markup=start_markup)
    return ConversationHandler.END

def question(update, context):
    global z, id
    v = update.message.text
    print(z[id])

```

```

        if v.lower() == z[id][2]:
            update.message.reply_text(
                "Абсолютно верно 🎉, ваши баллы были начислены вам на счет. ",
                reply_markup=da_net_markup)
            update.message.reply_sticker(choice(win_stickers))
            update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
            print(user_id)
            print(cursor_words.execute("UPDATE users SET score = score + ?
WHERE id = ?", (10, user_id)).fetchall())
            connect_words.commit()
            return 6
        else:
            update.message.reply_text("Ответ не верный😞(("",
                reply_markup=da_net_markup)
            update.message.reply_sticker(choice(lose_stickers))
            update.message.reply_text("Хотите воспользоваться подсказкой?
(Да,Нет)")
            return 5

def hint(update, context):
    global z, id
    v = update.message.text
    print(z[id])
    if v.lower() == z[id][2]:
        update.message.reply_text(
            "Абсолютно верно 🎉, ваши баллы были начислены вам на счет. ",
            reply_markup=da_net_markup)
        update.message.reply_sticker(choice(win_stickers))
        update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
        print(cursor_words.execute("UPDATE users SET score = score + ?
WHERE id = ?", (5, user_id)).fetchall())
        connect_words.commit()
        return 6
    else:
        update.message.reply_text("Ответ не верный😞(("",
reply_markup=da_net_markup)
        update.message.reply_text("Правильный ответ: " + z[id][2])
        update.message.reply_sticker(choice(lose_stickers))
        update.message.reply_text("Хотите прослушать как читается это
слово? (Да,Нет)")
        return 6

def hintornot(update, context):
    global z, id, hint_list
    v = update.message.text
    if v.lower() in "нет":
        update.message.reply_text("Хотите продолжить игру?",
reply_markup=da_net_markup)
        return 4
    elif v.lower() in "да":
        hint_list = [z[id][2]] + sample(z, 2)
        hint_list[1] = hint_list[1][2]
        hint_list[2] = hint_list[2][2]
        while len(set(hint_list)) != 3:
            hint_list = list(set(hint_list))
            hint_list.append(choice(z)[2])

```



```

shuffle(hint_list)
update.message.reply_text("Выберите 1 слово",

reply_markup=ReplyKeyboardMarkup([hint_list], one_time_keyboard=True))
    return 2

def continueornot(update, context):
    global z, id
    v = update.message.text
    if v.lower() in "нет":
        update.message.reply_text("GG WP", reply_markup=start_markup)

update.message.reply_sticker("CAACAgIAAxkBAAEErpRieAcjAAEaF8getDqoPMV7nNMdf
Z4AAhwRAAJcxFFJV96Mu-GSkgUkBA")
        return ConversationHandler.END
    elif v.lower() in "да":
        update.message.reply_text("OKAAAAAY, Let's GO",
reply_markup=stop_markup)
        id = randint(0, len(z) - 1)
        update.message.reply_text("Переведите слово (перевод должен быть в
несовершенном виде инфинитива)\n" + z[id][1])
        return 1

def wanttolisten(update, context):
    global z, id
    v = update.message.text
    if v.lower() in "да":
        word = gTTS(text=z[id][1], lang="en", slow=False)
        word.save("currentword.mp3")
        update.message.reply_audio(open(r'currentword.mp3', 'rb'))
        update.message.reply_text("Хотите продолжить игру?",
reply_markup=da_net_markup)
    elif v.lower() in "нет":
        update.message.reply_text("Хотите продолжить игру?",
reply_markup=da_net_markup)
        return 4

conv_handler = ConversationHandler(
    entry_points=[CommandHandler('words', words)],
    states={
        1: [MessageHandler(Filters.text & ~Filters.command, question)],
        2: [MessageHandler(Filters.text & ~Filters.command, hint)],
        3: [MessageHandler(Filters.text & ~Filters.command, words)],
        4: [MessageHandler(Filters.text & ~Filters.command,
continueornot)],
        5: [MessageHandler(Filters.text & ~Filters.command, hintornot)],
        6: [MessageHandler(Filters.text & ~Filters.command, wanttolisten)]
    },

    fallbacks=[CommandHandler('stop', stop)]
)

```

Вывод:

