""" Работает с этими модулями:

python-telegram-bot==11.1.0 redis==3.2.1 """

```
import os
import logging
import redis

from telegram.ext import Filters, Updater
from telegram.ext import CallbackQueryHandler, CommandHandler, MessageHandler

_database = None

def start(bot, update):
    """
    Хэндлер для состояния START.

    Бот отвечает пользователю фразой "Привет!" и переводит его в состояние ECHO.
    Теперь в ответ на его команды будет запускаться хэндлер echo.
    """
    update.message.reply_text(text='Привет!')
    return "ECHO"


def echo(bot, update):
    """
    Хэндлер для состояния ECHO.

    Бот отвечает пользователю тем же, что пользователь ему написал.
    Оставляет пользователя в состоянии ECHO.
    """
    users_reply = update.message.text
    update.message.reply_text(users_reply)
    return "ECHO"


def handle_users_reply(bot, update):
    """
    Функция, которая запускается при любом сообщении от пользователя и решает как его
обработать.

    Эта функция запускается в ответ на эти действия пользователя:
        * Нажатие на inline-кнопку в боте
        * Отправка сообщения боту
        * Отправка команды боту
    Она получает стейт пользователя из базы данных и запускает соответствующую
функцию-обработчик (хэндлер).
    Функция-обработчик возвращает следующее состояние, которое записывается в базу
данных.
    Если пользователь только начал пользоваться ботом, Telegram форсит его написать
"/start",
    поэтому по этой фразе выставляется стартовое состояние.
    Если пользователь захочет начать общение с ботом заново, он также может
```

```
воспользоваться этой командой.
    """
    db = get_database_connection()
    if update.message:
        user_reply = update.message.text
        chat_id = update.message.chat_id
    elif update.callback_query:
        user_reply = update.callback_query.data
        chat_id = update.callback_query.message.chat_id
    else:
        return
    if user_reply == '/start':
        user_state = 'START'
    else:
        user_state = db.get(chat_id).decode("utf-8")

    states_functions = {
        'START': start,
        'ECHO': echo
    }
    state_handler = states_functions[user_state]
    # Если вы вдруг не заметите, что python-telegram-bot перехватывает ошибки.
    # Оставляю этот try...except, чтобы код не падал молча.
    # Этот фрагмент можно переписать.
    try:
        next_state = state_handler(bot, update)
        db.set(chat_id, next_state)
    except Exception as err:
        print(err)

def get_database_connection():
    """
    Возвращает конекшн с базой данных Redis, либо создаёт новый, если он ещё не
создан.
    """
    global _database
    if _database is None:
        database_password = os.getenv("DATABASE_PASSWORD")
        database_host = os.getenv("DATABASE_HOST")
        database_port = os.getenv("DATABASE_PORT")
        _database = redis.Redis(host=database_host, port=database_port,
password=database_password)
    return _database


if __name__ == '__main__':
    token = os.getenv("TELEGRAM_TOKEN")
    updater = Updater(token)
    dispatcher = updater.dispatcher
    dispatcher.add_handler(CallbackQueryHandler(handle_users_reply))
    dispatcher.add_handler(MessageHandler(Filters.text, handle_users_reply))
```

```
    dispatcher.add_handler(CommandHandler('start', handle_users_reply))
    updater.start_polling()
```