**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Разработка простого бота для Telegram с использованием языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Маркин Денис |  | Нардид Анатолий Николаевич |
|  |  |  |
|  |  |  |

Москва, 2024 г.

Постановка задачи

1. Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Текст программы

.env

TOKEN=secret

ADMIN=secret

Токены не указаны в целях приватности

Bot.py

import asyncio

import os

from dotenv import find\_dotenv, load\_dotenv

from aiogram import Bot, Dispatcher, types

from aiogram.filters import CommandStart

load\_dotenv(find\_dotenv())

from handlers.user\_private import user\_private\_router

from common.bot\_cmds\_list import private

ALLOWED\_UPDATES = ['message', 'edited\_message']

bot = Bot(token = os.getenv('TOKEN'))

dp = Dispatcher() #класс диспетчера

dp.include\_router(user\_private\_router)

async def main() -> None:

    await bot.delete\_webhook(drop\_pending\_updates=True)

    try:

        await bot.set\_my\_commands(commands=private, scope=types.BotCommandScopeAllPrivateChats())

    except Exception as e:

        print(f"An error occurred: {e}")

    await dp.start\_polling(bot, allowed\_updates= dp.resolve\_used\_update\_types() ) #запуск бота

asyncio.run(main())

Папка Klava

Inline.py

from aiogram.types import InlineKeyboardButton

from aiogram.utils.keyboard import InlineKeyboardBuilder

def get\_callback\_btns(

    \*,

    btns: dict[str, str],

    sizes: tuple[int] = (2,)):

    keyboard = InlineKeyboardBuilder()

    for text, data in btns.items():

        keyboard.add(InlineKeyboardButton(text=text, callback\_data=data))

    return keyboard.adjust(\*sizes).as\_markup()

def get\_url\_btns(

    \*,

    btns: dict[str, str],

    sizes: tuple[int] = (2,)):

    keyboard = InlineKeyboardBuilder()

    for text, url in btns.items():

        keyboard.add(InlineKeyboardButton(text=text, url=url))

    return keyboard.adjust(\*sizes).as\_markup()

def get\_inlineMix\_btns(

    \*,

    btns: dict[str, str],

    sizes: tuple[int] = (2,)):

    keyboard = InlineKeyboardBuilder()

    for text, value in btns.items():

        if '://' in value:

            keyboard.add(InlineKeyboardButton(text=text, url=value))

        else:

            keyboard.add(InlineKeyboardButton(text=text, callback\_data=value))

    return keyboard.adjust(\*sizes).as\_markup()

reply.py

from aiogram.types import KeyboardButton

from aiogram.utils.keyboard import ReplyKeyboardBuilder

def get\_keyboard(

    \*btns: str,

    placeholder: str = None,

    request\_contact: int = None,

    request\_location: int = None,

    sizes: tuple[int] = (2,),

):

    keyboard = ReplyKeyboardBuilder()

    for index, text in enumerate(btns, start=0):

        if request\_contact and request\_contact == index:

            keyboard.add(KeyboardButton(text=text, request\_contact=True))

        elif request\_location and request\_location == index:

            keyboard.add(KeyboardButton(text=text, request\_location=True))

        else:

            keyboard.add(KeyboardButton(text=text))

    return keyboard.adjust(\*sizes).as\_markup(

            resize\_keyboard=True, input\_field\_placeholder=placeholder)

user\_pirvate.py

import asyncio

import os

import random

import sqlite3

from aiogram import F, types, Router, Bot

from aiogram.filters import CommandStart, Command, or\_f

from klava import reply, inline

from klava.reply import get\_keyboard

user\_private\_router = Router()

path\_mas = ['C:\images\DIO.jpg', "C:\images\Megaded.png", "C:\images\Lemmi.jpg",   "C:\images\Phil.jpg",  "C:\images\Bruce.jpg",]

citata\_mas= ['Музыканты моего поколения — живой пример того, чего можно достичь при наличии веры в свое правое дело, и мы пришли сюда, чтобы сказать другим: «Не сдавайтесь!»',

             '– У доктора не было права так со мной разговаривать. Я не собираюсь позволять кому-то вроде него посылать во вселенную слова, которые не соответствуют действительности обо мне. В моей реальности, когда он говорил эти вещи, они были неправдой для меня, и я не собирался это принимать. Дэйв Мастейн о диагнозе, не позволяющему ему играть ',

             'Быть быстрым и злым недостаточно - необходимо быть быстрым и злым ублюдком и тогда люди полюбят тебя'

             ]

ADMIN\_ID = os.getenv('ADMIN')

IS\_ADMIN = False

IGNORE\_CASE = False

bot = Bot(token = os.getenv('TOKEN'))

def delete\_ach():

    conn = sqlite3.connect('achs.db')

    cursor = conn.cursor()

    cursor.execute('''DELETE FROM achs''')

    conn.commit()

    conn.close()

def create\_db():

    conn = sqlite3.connect('achs.db')

    cursor = conn.cursor()

    cursor.execute('''

        CREATE TABLE IF NOT EXISTS achs (

            id INTEGER PRIMARY KEY,

            user\_id INTEGER NOT NULL,

            ach TEXT NOT NULL

        )

    ''')

    conn.commit()

    conn.close()

def save\_ach(user\_id, message):

    conn = sqlite3.connect('achs.db')

    cursor = conn.cursor()

    cursor.execute('''

        INSERT INTO achs (user\_id, ach) VALUES (?, ?)

    ''', (user\_id, message))

    conn.commit()

    conn.close()

def get\_ach(user\_id):

    conn = sqlite3.connect('achs.db')

    cursor = conn.cursor()

    cursor.execute('SELECT ach FROM achs WHERE user\_id = ?', (user\_id,))

    messages = cursor.fetchall()

    conn.close()

    return messages

create\_db()

#------------------------Общее-----------------

@user\_private\_router.message(or\_f(CommandStart(), (F.text.lower() == "назад")))

async def start\_command(message: types.Message):

    await message.answer("Привет! Я бот 🤖, который поможет тебе с учебой!🧠 Я могу посоветовать музыку 🎵 для занятий, подсказать действенные техники для успешного обучения 🖥️, придать мотивацию, а так же помогу тебе вести список дел, которые ты завершил 📂 Давай начнем!",

        reply\_markup = reply.get\_keyboard(

            "Мои достижения",

            "Техники",

            "Поддержать дух",

            "Музыка",

            sizes = (2, 2)

        ),

    )

#------------Про достижения SQL и прочие страшилки-----------------------

@user\_private\_router.message(or\_f(Command('ach'), (F.text.lower() == "мои достижения")))

async def info\_panel(message: types.Message):

    await message.answer("Выбери, что ты хочешь сделать. Если хочешь внести дело, напиши его в следующем формате 'Д:Спас кота' 📚",

                         reply\_markup= reply.get\_keyboard(

                             "Посмотреть список 😎",

                             "Очистить список ❌ ",

                             "Назад ↓",

                             sizes = (2, 2)

                         ),

    )

@user\_private\_router.message(F.text.lower().contains("д:"))

async def info\_panel(message: types.Message):

    str = message.text[2:]

    save\_ach(message.from\_user.id, str)

    await message.answer("Ваше дело занесено!✅")

@user\_private\_router.message(F.text.lower() == 'очистить список')

async def info\_panel(message: types.Message):

    delete\_ach()

    await message.answer("Список достижений очищен!❌")

@user\_private\_router.message(F.text.lower() == 'посмотреть список')

async def info\_panel(message: types.Message):

    user\_id = message.from\_user.id

    messages = get\_ach(user\_id)

    if messages:

        response = "Ваши дела: \n"

        for msg in messages:

            response += f"{msg[0]} \n"

    else:

        response = "Вы пока ничего не сделали🖕"

    await message.answer(response)

#----------------------Про мотивацию--------------------------------

@user\_private\_router.message(or\_f(Command("quote"), (F.text.lower() == "поддержать дух")))

async def info\_panel(message: types.Message):

    a = random.randint(0, 4)

    try:

        file\_path = path\_mas[a]

        photo = types.FSInputFile(path = file\_path)

        await bot.send\_photo(chat\_id=message.chat.id, photo=photo)

        await bot.send\_message(message.chat.id, text = citata\_mas[a])

    except FileNotFoundError:

        await bot.send\_message(message.chat.id, "Изображение не найдено.")

    except Exception as e:

        await bot.send\_message(message.chat.id, f"Произошла ошибка: {e}")

#-----------------------Про техники------------------------

@user\_private\_router.message(or\_f(Command('techniques'), (F.text.lower() == "техники")))

async def info\_panel(message: types.Message):

    await message.answer("Существует несколько техник для эффективного обучения 🌶️ про какую вы хотите узнать подробнее? 🤔",

                         reply\_markup= reply.get\_keyboard(

                             "Помодоро",

                             "Метод '90 на 30'",

                             "Матрица Эйзенхауэра",

                             "Метод GTD",

                             "Назад ",

                             sizes = (2, 2)

                         ),

    )

@user\_private\_router.message((F.text.lower() == 'помодоро'))

async def menu\_command(message: types.Message):

    await message.answer("Поставьте таймер на 25 минут и в течение этого времени выполняйте работу, ни на что не отвлекаясь. После "

                          "сделайте перерыв 5 минут и повторите цикл заново — продолжайте работу. Через каждые четыре цикла делайте большой "

                           "перерыв в 30 минут. Смысл метода в том, что монотонная работа разбивается на небольшие части, и после каждой вы получаете заслуженный отдых "

                            "Так легче взяться за такие задачи, и можно выполнить их лучше",

                            reply\_markup= inline.get\_inlineMix\_btns(

                                btns = {

                                    'Засечь': 'time1',

                                    'Подробнее': 'https://skillbox.ru/media/management/pomodoro/'

                                }

                            )

    )

@user\_private\_router.message((F.text.lower() == "метод '90 на 30'"))

async def menu\_command(message: types.Message):

    await message.answer("По этому методу на работу отводится полтора часа (90 минут), а после — полчаса на отдых. После этого цикл повторяется."

                         " Каждый полный цикл занимает два часа."

                        "Важно: первые циклы отводите на самые важные и сложные дела, а следующие – на менее важные",

                            reply\_markup= inline.get\_inlineMix\_btns(

                                btns = {

                                    'Засечь': 'time2',

                                    'Подробнее': 'https://habr.com/ru/companies/click/articles/656109/'

                                }

                            )

    )

@user\_private\_router.message((F.text.lower() == "метод gtd"))

async def menu\_command(message: types.Message):

    await message.answer("Запишите все ваши дела в блокнот или тетрадь, распределите ее по разным категориям, например 'Учеба', 'Друзья', 'Дом'. Затем, следуйте следующему"

                         " плану и постепенно выполняйте дела!",

                            reply\_markup= inline.get\_inlineMix\_btns(

                                btns = {

                                    'Я составил список': 'Я сос',

                                    'Подробнее': 'https://habr.com/ru/articles/599391/'

                                }

                            )

    )

@user\_private\_router.message((F.text.lower() == "матрица эйзенхауэра"))

async def menu\_command(message: types.Message):

    await message.answer("Вам нужно записать все ваши дела, распределив их на четыре группы по важности и срочности:"

                        "срочные и важные; срочные, но не важные; несрочные, но важные; несрочные и не важные",

                            reply\_markup= inline.get\_inlineMix\_btns(

                                btns = {

                                    'Предоставь таблицу': 'табло',

                                    'Подробнее': 'https://skillbox.ru/media/management/matritsa\_eyzenkhauera/'

                                }

                            )

    )

@user\_private\_router.callback\_query(F.data.startswith("табло"))

async def process\_callback\_button(callback\_query: types.CallbackQuery):

    try:

        file\_path = 'C:\images\matrix.jpg'

        photo = types.FSInputFile(path = file\_path)

        await bot.send\_photo(chat\_id=callback\_query.message.chat.id, photo=photo)

        await bot.send\_message(callback\_query.message.chat.id, "Хорошо потрудиться!")

    except FileNotFoundError:

        await bot.send\_message(callback\_query.message.chat.id, "Изображение не найдено.")

    except Exception as e:

        await bot.send\_message(callback\_query.message.chat.id, f"Произошла ошибка: {e}")

@user\_private\_router.callback\_query(F.data.startswith("Я сос"))

async def process\_callback\_button(callback\_query: types.CallbackQuery):

    try:

        file\_path = 'C:\images\GTD.jpg'

        photo = types.FSInputFile(path = file\_path)

        await bot.send\_photo(chat\_id=callback\_query.message.chat.id, photo=photo)

        await bot.send\_message(callback\_query.message.chat.id, "Успешной работы!")

    except FileNotFoundError:

        await bot.send\_message(callback\_query.message.chat.id, "Изображение не найдено.")

    except Exception as e:

        await bot.send\_message(callback\_query.message.chat.id, f"Произошла ошибка: {e}")

@user\_private\_router.callback\_query(F.data.startswith("t"))

async def process\_callback\_button(callback\_query: types.CallbackQuery):

    await bot.answer\_callback\_query(callback\_query.id)

    if callback\_query.data == 'time1':

        await bot.edit\_message\_text("Время пошло! Работайте 25 минут 💻",

                                     chat\_id=str(callback\_query.from\_user.id),

                                     message\_id=callback\_query.message.message\_id)

        await asyncio.sleep(25\*60)

        await bot.edit\_message\_text("Время вышло! Сделайте большой или маленький перерыв и возвращайтесь 🥂 ",

                                     chat\_id=str(callback\_query.from\_user.id),

                                     message\_id=callback\_query.message.message\_id)

    elif callback\_query.data == 'time2':

        await bot.edit\_message\_text("Время пошло! Работайте 90 минут 💻",

                                     chat\_id=str(callback\_query.from\_user.id),

                                     message\_id=callback\_query.message.message\_id)

        await asyncio.sleep(90\*60)

        await bot.edit\_message\_text("Время вышло! Сделайте перерыв 30 минут и возвращайтесь 🥂",

                                     chat\_id=str(callback\_query.from\_user.id),

                                     message\_id=callback\_query.message.message\_id)

#----------------------------------------------Музычка-------------------------------------------

@user\_private\_router.message(or\_f(Command('music'), (F.text.lower() == "музыка")))

async def process\_callback\_button(message: types.Message):

    await message.answer("Многие любят выполнять свои обязанности под тихую музыку на фоне 🎧 она помогает работать дольше и делать это в более хорошем настроении. "

                         "Чтож, какое настроение сегодня?🤗",

                         reply\_markup= inline.get\_inlineMix\_btns(

                             btns = {

                                "Мрачноеִֶ 🦇" : 'мрачное',

                                "Сосредоточенное 🎯" : 'серьезное',

                                "Спокойное 😌" : 'спокойное',

                                "Боевое 🎸" : 'боевое',

                             }

                         ),

    )

@user\_private\_router.callback\_query(F.data.startswith('бо'))

async def info\_panel(callback\_query: types.CallbackQuery):

    await bot.send\_message(

        chat\_id=callback\_query.from\_user.id,

        text=("Отлично, я вижу, что вы готовы драться не на жизнь, а насмерть 🪓 Только пообещайте не отвлекаться! Хорошей работы! "),

        reply\_markup=inline.get\_inlineMix\_btns(

            btns={

                "Зов севера": 'https://youtu.be/9WXsdApQIY4?si=DkLEiYA\_Kl90wsOU',

                "Вигвам шамана": 'https://youtu.be/\_Nqlg\_hZdVc?si=l24THmcdVUb58M7o',

                "Мудрость дракона": 'https://youtu.be/2rcckLnrAbg?si=XNG7\_pKmVYj8oCcA',

                "Тяжелый метал": 'https://my.mail.ru/mail/alex-enm/video/46908/714343.html?from=videoplayer',

            }

        )

    )

    await callback\_query.answer()

@user\_private\_router.callback\_query(F.data.startswith('спок'))

async def info\_panel(callback\_query: types.CallbackQuery):

    await bot.send\_message(

        chat\_id=callback\_query.from\_user.id,

        text=("Рад, что вы сегодня спокойны 🌊 с вашими задачами вам поможет справиться приятный джаз и Lo-Fi, приятного прослушивания! "),

        reply\_markup=inline.get\_inlineMix\_btns(

            btns={

                "Lo-fi": 'https://www.youtube.com/live/d2VdpHxmbPE?si=rLOMXqrqgFabNSZd',

                "Chill Lo-fi": 'https://youtu.be/CLeZyIID9Bo?si=3yI1pOfC22Zlsk2w',

                "Спокойный джаз": 'https://www.youtube.com/live/2oWN7waCXIo?si=M1q5WbXAivpAs-NH',

                "Босса нова": 'https://youtu.be/7GkHh3qLwgU?si=7HhQRYjk0CskJA0U',

            }

        )

    )

    await callback\_query.answer()

@user\_private\_router.callback\_query(F.data.startswith('серь'))

async def info\_panel(callback\_query: types.CallbackQuery):

    await bot.send\_message(

        chat\_id=callback\_query.from\_user.id,

        text=("Вижу вы настроены на продуктивную работу! 🖥️ Готов поспорить, с этой музыкой ваше задание "

              "выполнится проще"),

        reply\_markup=inline.get\_inlineMix\_btns(

            btns={

                "Глубокая концентрация": 'https://youtu.be/j8L6IvuYGOQ?si=hGsxDzCnWRp1jjWM',

                "Ночной город": 'https://youtu.be/n9Y2Eb4BaSg?si=IXw-eY6Wrlx41JPg',

                "Осенний вечер": 'https://youtu.be/m6SOJlkN1zU?si=zPwF\_mCPxXOeuMK2',

                "Научная музыка": 'https://youtu.be/5i0Z0E5yaYI?si=Ys6Bm3yzDPCercDp',

            }

        )

    )

    await callback\_query.answer()

@user\_private\_router.callback\_query(F.data.startswith('мрач'))

async def info\_panel(callback\_query: types.CallbackQuery):

    await bot.send\_message(

        chat\_id=callback\_query.from\_user.id,

        text=("С таким настроением только строить великие планы и совершать великие деяния!⚔️ "

              "Вот несколько вариантов для продуктивной работы, но не забывайте, "

              "что даже великие не забывали отдыхать и совершали ошибки 🛡️"),

        reply\_markup=inline.get\_inlineMix\_btns(

            btns={

                "Гром и классика": 'https://youtu.be/86WzoZt-ezs?si=fBGxVPG1ARWeptER',

                "Величие Рима": 'https://youtu.be/hC0Th-HZx5M?si=g\_NkzZA58RRQmy2X',

                "Стойкость и терпение": 'https://youtu.be/Mbb\_un0rkH4?si=ooE2r5hLnQhVIuyd',

                "Пугающее средневековье": 'https://youtu.be/-ZWwmVDQ20Q?si=4cpfWbpLq44HNY3N',

            }

        )

    )

    await callback\_query.answer()

#---------Для админа, если оно вообще будет. SHATTEREDDD

ADMIN\_KB = get\_keyboard(

    "Добавить товар",

    "Изменить товар",

    "Удалить товар",

    "Я так, просто посмотреть зашел",

    placeholder="Выберите действие",

    sizes=(2, 1, 1),

)

@user\_private\_router.message(Command("admin"))

async def admin\_panel(message: types.Message):

    if message.from\_user.id == ADMIN\_ID:

        await message.answer("Вы администратор! Выберите опцию:", reply\_markup= ADMIN\_KB)

        IS\_ADMIN = True

    else:

        await message.answer("У вас нет прав администратора.")

#--------------------Пасхалски для самых маленьких-------------------

@user\_private\_router.message(F.text == "1602")

async def menu\_command(message: types.Message):

    await message.answer("Great trial is coming :)")

chat\_types.py

from aiogram.filters import Filter

from aiogram import Bot, types

class ChatTypeFilter(Filter):

    def \_\_init\_\_(self, chat\_types: list[str]) -> None:

        self.chat\_types = chat\_types

    async def \_\_call\_\_(self, message: types.Message) -> bool:

        return message.chat.type in self.chat\_types

IsAdmin = False

Bot\_cmds\_list.py

from aiogram.types import BotCommand

private = [

        BotCommand(command="ach", description="Посмотреть меню"),

        BotCommand(command="techniques", description="Узнать о техниках"),

        BotCommand(command="music", description="Для продуктивной работы"),

        BotCommand(command="quote", description="Поддержать дух"),

        BotCommand(command="start", description = "Запустить бота")

]

Результаты работы

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание