

**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:  
студент группы ИУ5-31Б  
Маркин Денис

Проверил:  
преподаватель каф. ИУ5  
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())

```

Панка 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()

```

```

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\_private.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 любят 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 coc"))
```

```
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)
```

```
#-----Музыка-----
```

```
@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 = {
```

```
"Мрачное 🌧" : '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_KI90wsOU',
                "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=3yl1pOfC22ZlSk2w',
            "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=("Видю 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/j8L6lvuYGOQ?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('мрпч'))

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=ooE2r5hLnQhVluyd',

"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.")
```

```
#-----Пасхалки 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

28 ноября

/start 19:18 ✓

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!

19:18

Мои достижения 19:18 ✓

Выбери, что ты хочешь сделать. Если хочешь внести дело, напиши его в следующем формате 'Д:Спас кота' 🐱

19:18

Д:Защитил физику 19:18 ✓

Ваше дело занесено! ✓ 19:18

Техники 19:18 ✓

Существует несколько техник для эффективного обучения 🌶️ про какую вы хотите узнать подробнее? 🤔

19:18

Метод '90 на 30' 19:18 ✓

Время пошло! Работайте 90 минут 🖥️ 19:18

Музыка 19:19 ✓

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? 😊

19:19

Мрачное 🌧️

Сосредоточенное 🎯

Спокойное 😌

Боевое 🎸

Вижу вы настроены на продуктивную работу! 🖥️ Готов поспорить, с этой музыкой ваше задание выполнится проще

19:19

Глубокая концентрация ↗

Ночной город ↗

Осенний вечер ↗

Научная музыка ↗

