

Московский государственный технический университет им.
Н.Э. Баумана

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

Отчет по лаб 5-6
“Разработка Телеграм бота”
Базовые компоненты интернет технологий

Вариант 8

Подготовил:
Студент группы РТ5-31Б
Коваленко В.И.

Проверил:
Доцент кафедры ИУ5
Гапанюк Ю.Е.

12 Декабря 2021г.

Цель лабораторной работы: изучение разработки ботов в Telegram.

Задание:

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

Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

Листинг

```
main.py
import logging
from aiogram import *
from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup,
KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton
from aiogram.dispatcher.filters import Text
from aiogram.dispatcher import *
from aiogram.dispatcher.filters.state import *
from aiogram.types.bot_command import *
from logging import *
import asyncio
from aiogram.contrib.fsm_storage.memory import *
import configparser

goods_name = ["видеокарта", "оперативная память", "флешка"]
vid_size = ["2Гб", "4Гб", "8Гб"]
dost = ["Москва", "Санкт-Петербург", "Казань"]

class Order(StatesGroup):
    waiting_goods = State()
    waiting_size = State()
    waiting_dost = State()

async def Start(message: types.Message):
    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    for name in goods_name:
        keyboard.add(name)
    await message.answer("Выберите товар, который хотите приобрести:",
reply_markup=keyboard)
    await Order.waiting_goods.set()

async def Chose(message: types.Message, state: FSMContext):
```

```

if message.text.lower() not in goods_name:
    await message.answer(f"К сожалению, данного товара нет в наличии. \n"
                        f"Выберете товар из списка ниже")
    return
await state.update_data(chosen_good=message.text.lower())
keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
for size in vid_size:
    keyboard.add(size)
await Order.next()
await message.answer("Выберите размер памяти:", reply_markup=keyboard)
await Order.waiting_size.set()

async def Size_chose(message: types.Message, state: FSMContext):
    if message.text.lower() not in vid_size:
        await message.answer(f"К сожалению, данного объема нет в наличии. \n"
                            f"Выберите объем из списка ниже")
        return
    await state.update_data(chosen_size=message.text.lower())

    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    for d in dost:
        keyboard.add(d)
    await Order.next()
    await message.answer("Выберите город доставки:", reply_markup=keyboard)

async def Dost_chose(message: types.Message, state: FSMContext):

    user_data = await state.get_data()

    await message.answer(f"Вы заказали {user_data['chosen_good']} на
    {user_data['chosen_size']} в город {message.text.lower()}! \n"
                        f"Спасибо за заказ!")
    await state.finish()

def register_handlers_food(dp: Dispatcher):
    dp.register_message_handler(Start, commands="go", state="*")
    dp.register_message_handler(Chose, state=Order.waiting_goods)
    dp.register_message_handler(Size_chose, state=Order.waiting_size)
    dp.register_message_handler(Dost_chose, state=Order.waiting_dost)

async def bot_start(message: types.Message, state: FSMContext):

```

```

    await state.finish()
    await message.answer(
        'Это магазин комплектующих, нажмите кнопку "/go" чтобы перейти к товарам',
        reply_markup=types.ReplyKeyboardRemove()
    )

```

```

def register_handlers_common(dp: Dispatcher):
    dp.register_message_handler(bot_start, commands="start", state="*")

```

```

logger = logging.getLogger(__name__)

```

```

async def main():
    logging.basicConfig(
        level=logging.INFO,
        format="%(asctime)s - %(levelname)s - %(name)s - %(message)s",
    )
    logger.error("Starting bot")
    config = configparser.ConfigParser()
    config.read('bot.ini')
    bot = Bot(token=config["BOT"]["token"])
    dp = Dispatcher(bot, storage=MemoryStorage())

    register_handlers_common(dp)
    register_handlers_food(dp)
    await dp.start_polling()

if __name__ == '__main__':
    asyncio.run(main())

```

bot.ini

[BOT]

token = 5028996897:AAG1ks-NKXIFdtOxY6996bD38jnstm-OoWI

