

Московский государственный технический университет им. Н.Э. Баумана

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

Отчет по проделанной лабораторной работе номер 5-6
по курсу

Базовые компоненты интернет технологий

Вариант 14

Подготовил:

Студент группы РТ5-31Б

Мамаев Т.Э.

Проверил:

Доцент кафедры ИУ5

Гапанюк Ю.Е.

23 Декабря 2021г

Описание задания

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

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

Текст программы

```
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 = ["KIA", "BMW", "Honda"]
vid_size = ["седан", "внедорожник", "купе"]
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 марку, которую 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, данной марки 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 кузова 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 автосалон, 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]
token = 5052833662:AAE4w-5HRbpQSb48H6AhxvJaP2V0XLUbN3A


Результат выполнения программы



testbot
бот




Сегодня




Тима
/start

✓✓ 13:25




testbot
Это автосалон, нажмите кнопку **/go** чтобы перейти к товарам

13:25




Тима
/go

✓✓ 13:25



testbot
Выберите марку, которую хотите приобрести:

13:25



Тима
KIA

✓✓ 13:25