

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

**Отчет по лабораторной работе №6
«Основные конструкции языка Python»**

Выполнил:
студент группы ИУ5-
32Б
Зайцев А.Д.

Москва, 2021 г.

Описание задания:

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

Текст программы:

Файл asyncbot2:

```
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.dispatcher.filters.state import StatesGroup, State
from aiogram.dispatcher.filters import Command
from aiogram.dispatcher.storage import FSMContext
from aiogram.utils import executor
from config import TOKEN
from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup,
KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton, Message

from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.contrib.middlewares.logging import LoggingMiddleware

bot = Bot(token=TOKEN)
dp = Dispatcher(bot, storage=MemoryStorage())
dp.middleware.setup(LoggingMiddleware())

class Test(StatesGroup):
    Q0 = State()
    Q1 = State()
    Q2 = State()
    Q3 = State()

first_course={
    1 : "Уха",
    2 : "Борщ",
    3 : "Солянка"
}
main_course={
    1 : "Спагетти карбонара",
    2 : "Котлета от Шефа",
    3 : "Наггетсы в сычуанском соусе"
}
dessert={
    1 : "Шарик сливочного мороженого",
    2 : "Чизкейк",
    3 : "Макфлури"
}

@dp.message_handler(state="*", commands=['start'])
```

```

async def starting_process(message: types.Message):
    await bot.send_message(message.from_user.id, "Привет\nты можешь сделать заказ
из 3 блюд\n1)Первое блюдо\n2)Второе блюдо\n3)Десерт\nЧтобы начать формировать
заказ напиши /form\nЕсли хочешь начать формировать заказ заново, напиши /clear")
    await Test.Q0.set()

@dp.message_handler(state="*", commands=['clear'])
async def starting_process(message: types.Message, state: FSMContext):
    await state.finish()
    await Test.Q0.set()

#вызов команды /form в процессе заказа
@dp.message_handler(state=Test.Q1, commands=['form'])
async def refuse_command(message: types.Message ):
    await bot.send_message(message.from_user.id, "прежде чем формировать новый
заказ,завершите предыдущий либо напишите /clear")
@dp.message_handler(state=Test.Q2, commands=['form'])
async def refuse_command(message: types.Message ):
    await bot.send_message(message.from_user.id, "прежде чем формировать новый
заказ,завершите предыдущий либо напишите /clear")
@dp.message_handler(state=Test.Q3, commands=['form'])
async def refuse_command(message: types.Message ):
    await bot.send_message(message.from_user.id, "прежде чем формировать новый
заказ,завершите предыдущий либо напишите /clear")

@dp.message_handler(state=Test.Q0, commands=['form'])
async def starting_process(message: types.Message, state: FSMContext):
    await bot.send_message(message.from_user.id, "Выбери первое
блюдо\n1)Уха\n2)Борщ\n3)Солянка")
    await Test.Q1.set()

@dp.message_handler(state=Test.Q1)
async def first_choosing(message: types.Message, state: FSMContext):

    answer = int(message.text)
    if (answer != 1 and answer !=2 and answer !=3):
        return await bot.send_message(message.from_user.id, "такого варианта
ответа нет выбери еще раз!!!!!!!!!!")
    await state.update_data(q1 = answer)
    await bot.send_message(message.from_user.id, "Выбери второе блюдо\n1)спагетти
карбонара\n2)котлета от шефа\n3)наггетсы в сычуанском соусе")
    await Test.Q2.set()

@dp.message_handler(state=Test.Q2)
async def second_choosing(message: types.Message, state: FSMContext):
    answer = int(message.text)
    if (answer != 1 and answer !=2 and answer !=3):
        return await bot.send_message(message.from_user.id, "такого варианта
ответа нет выбери еще раз!!!!!!!!!!")
    await state.update_data(q2 = answer)

```

```

        await bot.send_message(message.from_user.id, "Выбери десерт\n1)шарик
сливочного мороженого\n2)чизкейк\n3)макфлури")
        await Test.Q3.set()

@dp.message_handler(state=Test.Q3)
async def second_choosing(message: types.Message, state: FSMContext):
    answer = int(message.text)
    if (answer != 1 and answer !=2 and answer !=3):
        return await bot.send_message(message.from_user.id, "такого варианта
ответа нет выбери еще раз!!!!!!")
    await state.update_data(q3 = answer)
    data = await state.get_data()

    await bot.send_message(message.from_user.id, "Ваш заказ:\nПервое
блюдо:\n{}\nОсновное
блюдо:\n{}\nДесерт:\n{}".format(first_course[data.get("q1")], main_course[data.get(
"q2")], dessert[data.get("q3")]))
    await Test.Q0.set()

async def shutdown(dispatcher: Dispatcher):
    await dispatcher.storage.close()
    await dispatcher.storage.wait_closed()

if __name__ == '__main__':
    executor.start_polling(dp, on_shutdown=shutdown)

```

Файл config.py:

```
TOKEN="5086492621:AAEhnZviAv4rJ_gQ64dnPluMfrgWh1l13FU"
```

Пример выполнения программы:

